Chipless RFID Systems Using Advanced Artificial Intelligence

Larry M. Arjomandi

Nemai Chandra Karmakar

Chipless RFID Systems Using Advanced Artificial Intelligence

For a listing of recent titles in the
Artech House Microwave Library,
turn to the back of this book.

Chipless RFID Systems Using Advanced Artificial Intelligence

Larry M. Arjomandi

Nemai Chandra Karmakar

ARTECH HOUSE
BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data
A catalog record for this book is available from the British Library.

ISBN-13: 978-1-63081-948-4

Cover design by Andy Meaden Creative

10 9 8 7 6 5 4 3 2 1

Contents

Preface

When I was a student at Ferdowsi University of Mashhad I was passionate to learn about neural networks and microwave technology. I visited several research facilities across the nation in search of a trustworthy microwave simulation program. In one of these research facilities, I saw the CNL2 simulation software from Artech House books. I put together a letter and sent it by snail mail to the publisher, requesting an evaluation copy. To my amazement, Artech House sent me a floppy disk with the CNL2 demo on it. It greatly aided me in finishing my master's thesis.

The years passed, and I chose to pursue a PhD after becoming a senior engineer. I approached an Artech House editor at an international conference to see if they might publish my study, and they said yes. This is where the inspiration for this book's narrative began.

I wanted to make the reader's task easier when I began to compose this book. Instead of using tedious pure engineering mathematical formulas, I placed more focus on the application method in this work.

The book's major objective is to give readers—both practicing engineers and engineering students—modular design thinking and problem-solving abilities to help them with cutting-edge applications in the chipless RFID field. A wide range of applications in signal processing, digital electronics design, wireless sensor design, and the incorporation of machine learning into routine engineering processes are covered by the methodology presented in the book.

The reader's opinions, inquiries, and recommendations will be greatly welcomed. Please don't hesitate to get in touch by email at m.arjomandi@gmail.com.

Larry Arjomandi
January 2023

1

Introduction

The Internet of Things (IoT) is a real phenomenon around us, with more and more intelligent devices interconnected and managed by the internet. These "things" or "objects" are identified by various means and their data is processed, normally without human interaction. IoT can make the world more visible for everyone: a world where every product can be uniquely traced and analyzed. A shoe manufacturer would know where their products are all around the globe, how long they are used by consumers, and when and how they are recycled. Consumers would benefit from automated counterfeiting checks to ensure purchased goods are not knockoffs [1, 2]. A piece of clothing would interact with the washing machine and ask for a particular wash cycle setting [3]. And in general, theft would be a problem of the past because every product could be traced and located [4]. As well, the final consumer of a particular medicine could be determined from the location where the medicine container is opened, and the effect of the medicine could be measured on the patient as the bathroom analyzes and reports on the wastewater contents.

For the IoT to operate, there are many technologies to read and process the data, including visible bar codes, radio frequency identification (RFID), and biometric information readers such as fingerprint, face, and voice recognition. In all of these techniques, a transducer converts the analog data (such as the tag's frequency response data in the RFID or the face captured in the cell phone) to the digital form. This digitized data is then processed by a local processor (or remotely by a cloud processor) or stored in a database for the later use.

RFID technology is used as a part of wider IoT for tracking livestock, tracking products in factory assembly lines, and for many other applications. The RFID's means of capturing data is by using electromagnetic waves. This technology does not require a line of sight (LoS), which allows the RFID tags to be perfectly embedded inside of objects. This makes RFID tags better in managing shoplifting problems compared to optical barcodes as the tags are not necessarily visible and therefore have less chance of being tampered with. RFID tags attached to products during the manufacturing phase are not only stopping shoplifters and employees attempting to leave a store with stolen items, but they also provide meaningful data for lost items from factory to sale. As well, compared to visible bar codes or quick response (QR) codes, RFID tags also have a longer reading range, and an automatic readout can be done for real-time tracking without human intervention.

RFID applications are spread throughout many different markets and industries, including livestock identification and automated vehicle identification (AVI) systems, because of their capability to trace objects on the move. In manufacturing lines, where other identification methods fail during the rough and constant handling environment, RFID is able to track the items to the end of the process.

In this introduction chapter a brief description of the different parts of an RFID system and their characteristics is presented and problems identified in current practices.

1.1 Overview Model of RFID

The overall view of an RFID system is shown in Figure 1.1. It follows the general idea of the IoT: a reader (transducer) sends an interrogation signal to the tag, and the back scattered signal is captured by the reader. This received signal then is converted to the digital form using a high-speed analog-to-digital converter (ADC). This digitized data is then sent to a local processor (or a remote

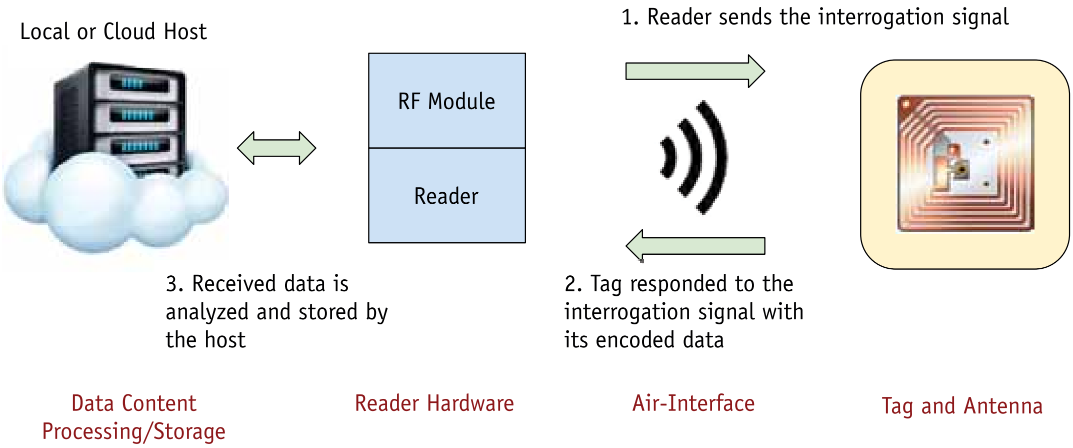

Local or Cloud Host
RF Module
Reader
1. Reader sends the interrogation signal
3. Received data is analyzed and stored by the host
2. Tag responded to the interrogation signal with its encoded data
Data Content Processing/Storage
Reader Hardware
Air-Interface
Tag and Antenna

Figure 1.1 The RFID system working overview. The system has three main components: tags, a reader, and detection algorithms. The local/cloud host in this picture is normally integrated into the reader itself.

cloud processor) for processing or is stored in a database for any further analysis.

1.2 Different Types of RFID

As previously said, any RFID system has three components: tags, a reader, and some decoding algorithms in the host. The evolution of RFID is interesting: the first RFID system was introduced in the 1940s during the last years of World War II [5]. The Germans knew that the radar reflection of an airplane is different once the airplane is rolling, so they asked their pilots to roll the planes before landing as a signal of being friendly. This was the first passive RFID tag system. British scientists expanded that exciting idea in a secret project under Watson-Watt's supervision. A transmitter was placed on each British plane, which was quiet until it received the ground station radar signals. The transmitter then began broadcasting a signal back to show the identification of the airplane as friendly. This was the first active RFID tag system [6].

Figure 1.2 illustrates the different types of RFID tags. RFID tags can be chipped or chipless. Chipped tags can be active/semi-active or passive. The chipped RFID tags contain application specific integrated circuits (ASIC) chipsets and they are powered up

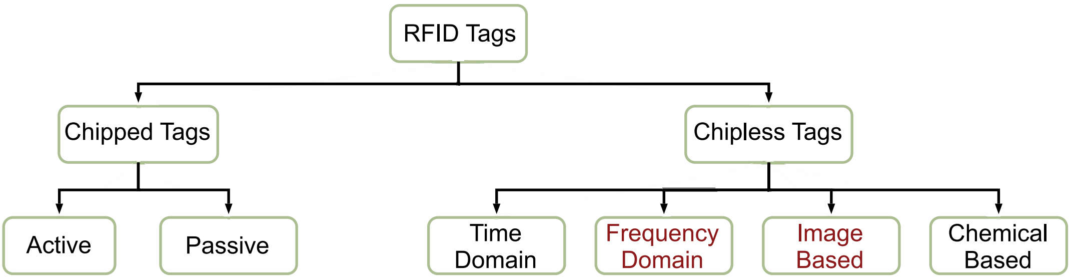


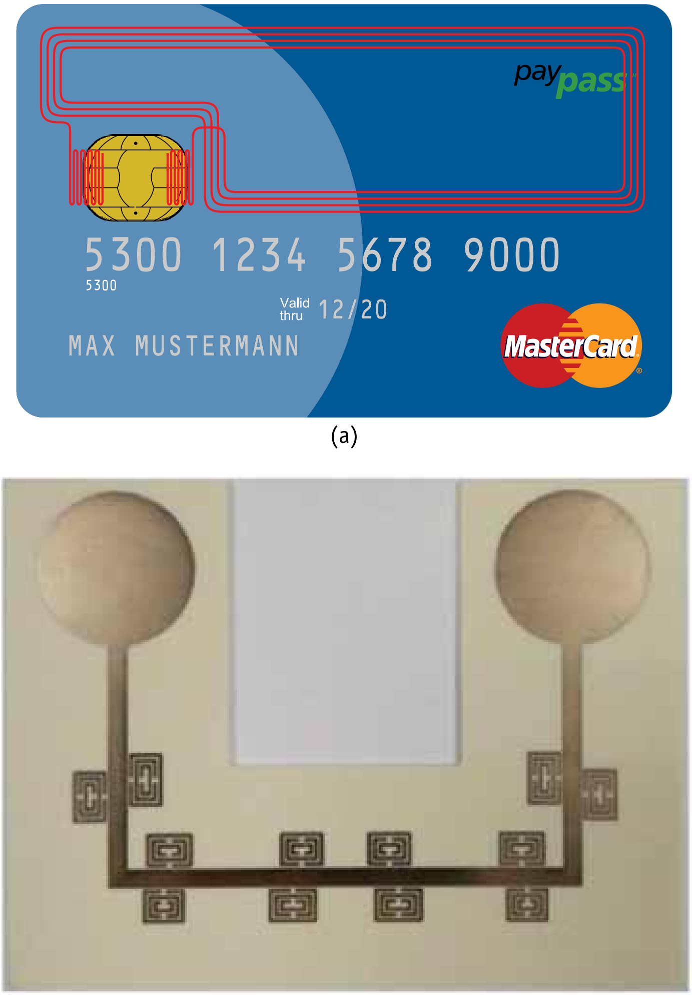
Figure 1.2 Different types of RFID tags at a glance. Our focus in this book will be frequency-domain and image-based tags, indicated in red.

by either an onboard battery in the active/semiactive tags or by the interrogation signal energy in the passive tags. An example of an active tag is your contactless credit card, where the chipset is powered by the electromagnetic energy generated by the card reader. The chipped tags respond to the readers by their communication protocols and reveal their encoded data. The other type are chipless tags. Chipless RFID tags are simply printed metallic patterns and contain no chipset. Chipless tags are all passive [7, 8]. An illustration of chipped and chipless tags is shown in Figure 1.3.

The absence of any smart elements in the chipless tags makes the design of their readers much more challenging compared to their chipped tag reader counterparts. The chipless tag detection on the reader side is based on the fact that tags act as backscatters and data encoders simultaneously. Based on the detection method, this backscattered signal is being processed and the tag's data is extracted. Chipless tag design should consider three factors: maximum backscatter reflection, maximum data encoding, and symmetrical design. These areas are covered in detail in Chapter 2, where the design aspects of the tags are discussed.

1.3 Different Types of Chipless RFID

As per Figure 1.2, based on the detection method of the backscattered signal, chipless RFID is classified into the four main categories: time-based, image-based, frequency-based, and other mixed (or hybrid) domains, such as polarization-phase or space-frequency [8, 11, 12].



(a)

(b)

Figure 1.3 An example of (a) chipped [9] and (b) chipless tags [10]. The coil in (a) amplifies the received signal and powers up the chip. The printed circles in (b) act like an antenna in the specific range of interest, and the small coils represent the encoded bits by making cavities in the frequency response. (Printed with permission.)

In the time-domain method, a sharp pulse is sent to the tag. The tag substrate is typically made of piezoelectric material, and an interdigital transducer (IDT) is used to transform the electromagnetic pulse into a surface acoustic wave (SAW). This SAW goes through some discontinuities in the tag's surface, which creates some different delays in the response. The return SAW is then converted back to an electromagnetic wave using IDT. This signal has therefore some train of delayed pulses that correspond to the position of those discontinuities in the tag's surface. Figure 1.4 illustrates a time-domain tag based on SAW substrate [8, 13].

Because of the time delay nature, the time domain method has more robustness in harsh environments, such as in railroad cars, compared to the other methods [14, 15]. Harsh environments for the chipless RFID here are places with lots of metal surroundings or moving objects. The time domain tags however need expensive SAW substrates and the reader is required to make and detect sharp narrow pulses with very fast ADCs.

In the frequency (or phase) domain method, the reader sends a broadband signal to the tag and "listens" to the echoes. The attenuation or cavities in the response signals indicate the presence of the encoded data in the tag. Frequency domain methods are by far the most commonly used ones and are the easiest to implement in the reader side. There are a few problems with frequency domain methods, however. If the substrate is not appropriate for the microwave or mm-wave frequency spectrum (like using generic papers or plastics as the tag's substrate), the Q-factor in the

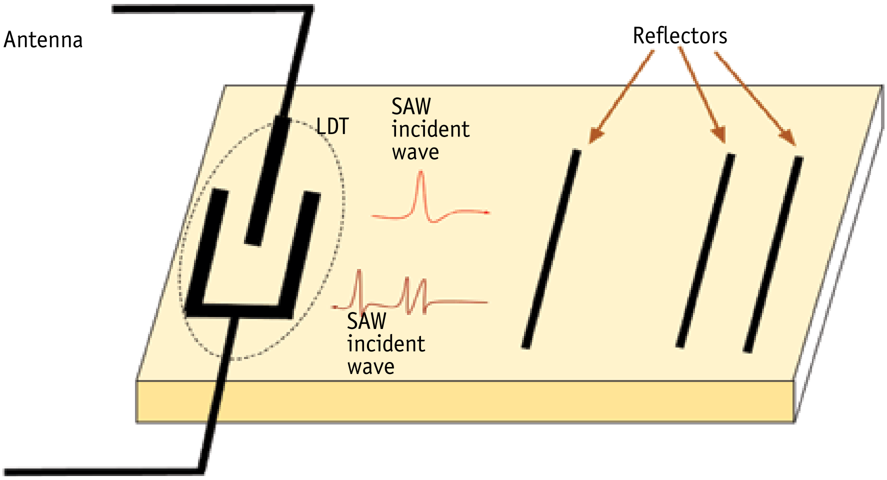


Figure 1.4 A time-domain tag based on SAW substrate. The tag consists of IDTs plus an antenna, and a few reflectors that act as encoders. The tag's surface is normally piezoelectric or diamond. Each reflector creates a delay in the nanosecond order, as shown by the reflected signal.

frequency response is not high, which results in reading difficulties and restricts the detectable bits and reading range. Changing the angle and distance of the reader to the tag results in a change in the frequency response. Putting a few tags together also shifts the frequency response of the individual tags. Figure 1.5 shows a frequency-based tag, which is widely used in this book.

In the image-based method, which is mainly based on synthetic aperture radar (SAR) technology [16], successive scanning is normally used to illuminate the tag in a two-dimensional (2-D) image. The tag is placed in a linear rail in front of the reader and its position changes as the linear rail moves. For each tag/reader position, the reader sends a pulse or a broadband signal and the resultant backscatter echoes are captured. Based on the relative position/angle of the reader to the tag, these backscattered signals are processed to make a high-resolution 2-D image of the tag. Instead of linear rail SAR, sometimes the tag is rotated in a fixed distance from the reader. This is similar to inverse synthetic aperture radar (iSAR) in radar applications. An alternative of physical movements for the tags and reader is using multiple-input-multiple-output (MIMO) technique, in which many real (or virtual) transmitters and receivers are implemented.

As the image-based methods produce a 2-D image of the tag, their decoding capacity is potentially higher than the other methods, and many well-developed image processing techniques are available for recognition and interpretation of the tag's data.

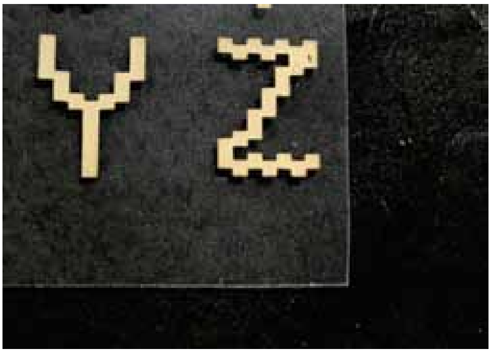


Figure 1.5 Sample of an alphanumerical chipless tag printed on a plastic substrate. Depending on the detection methods, these tags can be considered as frequency- or image-based tags.

However, image-based methods need high-precision rails in the SAR mode, many frequency band captures in different angles in the iSAR mode [17, 18], or a number of transmitters and receivers in the MIMO mode [14, 19, 20]. Their implementation is therefore more expensive and computation costs are higher once compared to the time or frequency methods.

Chemical tags are sensitive to the condition changes on their surroundings. The condition can be something like temperature or smell in perishable food. These tags may be used for storage of a few data encoded bits in the presence or absence of certain chemicals. Chemical tags are normally referred to as sensors. We will go into more details of frequency and image-based tags in Chapter 2.

1.4 Market Aspects for Chipless RFID

In the RFID domain, chipless tags have the advantage of being very cheap (less than a cent) and printable on different substrates. This includes printing on hard substrates (like printed circuit board (PCB), Taconic, and FR-4), plastic substrates (such as polyester), and, recently, on plain paper [21] or on paper with a thin layer of plastic cover (flexography) [22, 23]. Therefore, much like bar codes, chipless RFID tags cost very little and they can be placed and printed onto any item [12].

Chipped tags cost less than 10 cents for quantity orders of one million, and as stated above, chipless are less than a cent for 100,000 or more ordering quantities [24]. If data handling is the primary focus, chipped tags are usually best, and if certain physical properties (like sensing aspects) or the tag price is important, chipless tags are often prioritized. However, other factors, such as security, environment noise, and reading distance are also important in deciding which type of tag to use.

Figure 1.6 shows the forecast trend for both the whole RFID and chipless RFID markets, with an increasing trend of about 33% per year for the chipless RFID. While the total current (2018) market for RFID is around 20 billion U.S. dollars, chipless RFID has a share of 6.4 billion dollars, and growth estimation expects it to reach 11 billion dollars (red line) by 2020. This market value includes chipless tags, readers, and their accessories [24, 25].

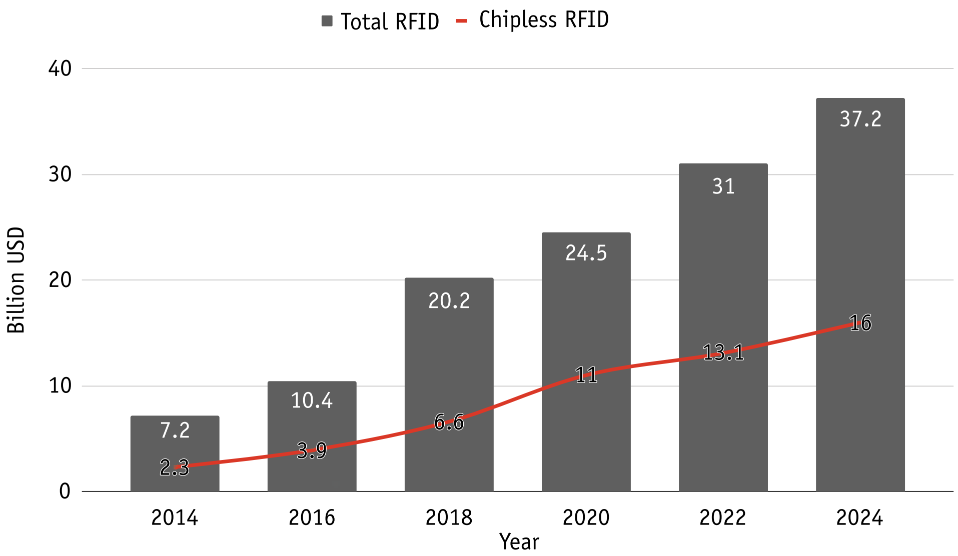


Figure 1.6 Projected global RFID market growth. Note that COVID-19 pandemic effects have not been considered.

1.5 RFID Frequency Spectrum

For RFID, the industrial, scientific, and medical (ISM) band can be used. The ISM radio bands are portions of the frequency spectrum reserved internationally for industrial, scientific, and medical purposes. The original intention of the ISM band allocation was to exclude applications in telecommunications, but as those bands do not require licenses, more and more communication-based applications are using this spectrum. These include near-field communications (NFC), Wireless Fidelity (Wi-Fi), and Bluetooth applications. As a general rule, the ISM spectrum is becoming smaller to open up space for the data-hungry application of the fifth generation of cellular networks (5G) in many countries [26].

At the higher side of the ISM band, there will be less interference expected as communications become more point-to-point and the behavior of the wave gets closer to the infrared spectrum. A wavelength between 1 and 10 mm (30–300 GHz) is called a mm-wave. In this spectrum, the frequency bands of 57–64 GHz and 244–246 GHz comprise the ISM band, so no licensing is needed for radio-location applications, including RFID. The availability of the 57–64 GHz spectrum is subject to local acceptance. In the United

States, only 61–61.5 GHz is available for the ISM application. In Australia, the whole spectrum is currently free for point-to-point communications [27].

There are a variety of uses for mm-wave spectrum, especially at the 60-GHz band. For RFID, the other applications that use this spectrum are considered to be noise. The main possible noise sources for RFID in this band are those from radio broadcasting (the IEEE 802.11ad protocol is licensed for Wireless Gigabit (WiGig) [28]) and cellular phone backbone transmissions for mobile telecommunication links. The main attenuators of this band are atmospheric absorption caused by water vapor and oxygen, as seen in Figure 1.7 [29–31]. Path loss air attenuation is a serious limitation factor in the mm-wave band compared to lower frequencies commonly used for chipless RFID. As will be explained in Chapter

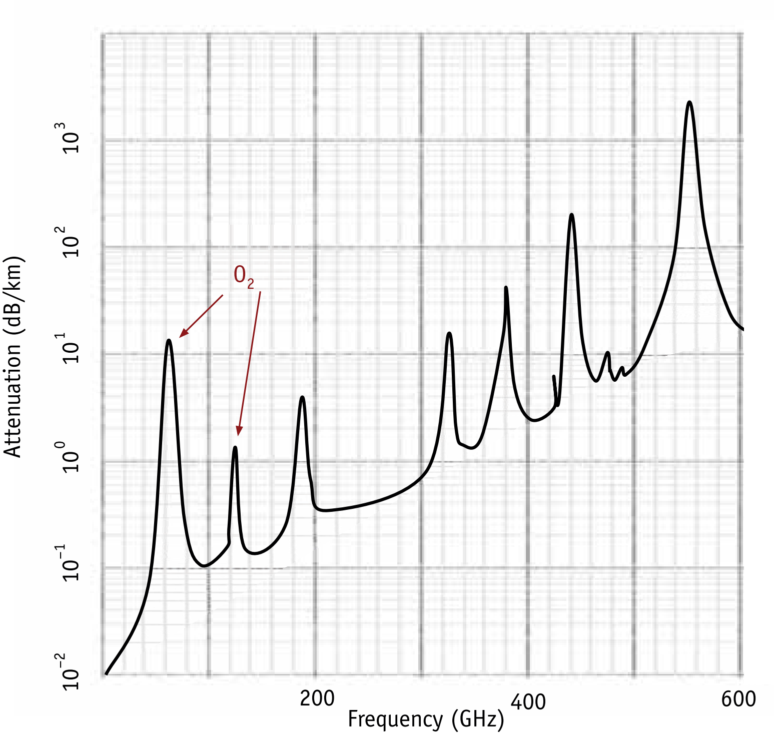


Figure 1.7 The main attenuators in mm-waves are water vapor and oxygen, with up to 10 dB/km for 60 GHz [29]. The attenuation graph is for moist air (101 kPa). The attenuation peaks around 60 and 120 GHz are caused by oxygen, and the rest of the peaks are due to water vapor.

3, the air attenuation in the 60-GHz spectrum is more than 10 times higher compared to the 25-GHz spectrum and 30 dB higher than the 6-GHz ultrawideband (UWB) spectrum for the same reader-to-tag distances.

The penetration depth and surface effects of the mm-wave band is also important. Because the wavelengths of mm-waves are large compared to natural and synthetic fibers, mm-waves tend to pass through most materials, such as clothing, plastic, and papers, and have deeper dielectric penetration. This makes mm-wave spectrum an ideal candidate for scanning technologies as they can achieve higher resolutions [32].

Implementing chipless RFID in the 60-GHz ISM band has its own advantages and disadvantages. The advantages are higher bits per tag's footprint, or equivalently, lower tag size for any specific data, miniature reader components, and antennas, ability to use more directive reader antenna arrays with narrower beamwidths, and higher gain, and hence less tag collision problems. The disadvantages are more attenuation in air and substances as it gets closer to the infrared spectrum (700 nm to 1 mm), and more complicated circuits.

This book addresses the practical challenges in implementing and detecting printed tags in mm-wave. These challenges are described next.

1.6 Challenges in Implementing Chipless RFID in the mm-Wave Spectrum

There are many challenging issues for implementing chipless RFID tags and reader systems in the mm-wave in real-world scenarios. These issues are explored below.

- *Detecting chipless tags in low-cost substrates is challenging.* Frequency and image-based systems depend on the tag resonances. The microwave substrates (such as Taconic and Rogers) provide higher backscatter reflections and less absorption, but those substrates are neither flexible enough nor cheap. In this book, we examine the screen printing technique used on plastic substrate for making chipless RFID tags, which results in fewer backscattering signals compared to the microwave substrates, so tag detection and decoding is much more challenging.

- *Letter tags on plastic substrates have low Q-factors.* There are a few advantages in using alphanumerical tags (i.e., tags that are represented by letters and numbers). Letters and alphabets have already been used as chipless RFID tags in the frequency domain [33–35], but they were mainly developed for single-letter tags with limited encoding capacities, and they are not used in the mm-wave frequency range on printable materials.
- *Decoding a chipless tag in different angles/distances is not always possible.* Chipless tags are perfectly decoded in the following ideal conditions: if the distance and background noise are controlled thoroughly (such as inside anechoic chambers), and if the reading conditions are kept the same. In practice, however, changing the relative tag-to-reader distance and angle makes decoding hard in a dynamic environment because there will always be shifts in the frequency response and changes in the amplitude and phase. This is a common problem for all types of chipless tags regardless of their detection methods.
- *Chipless tag misdetection is a real issue.* Although current readers have an acceptable level of detection, there is always a possibility that the system will misdetect a tag. This simply means the reported tag identification (tagID) by the reader is wrong. This is quite a dangerous situation in sensitive areas such as finance and grocery stores. The problem comes from the fact that a chipless tag is simply a piece of printed conductive on a dielectric material and does not have any smart elements, and therefore there is no error correction handshake signaling between the tag and the reader.
- *Hardware implementation at 60 GHz becomes challenging.* The higher the frequency, the more difficult it is to work with radio frequency (RF) components. As an engineering approach, using a modular system is recommended, which enables the designer to evaluate each receiver section independently before going ahead to connect them in a circuit.

The modular design procedure also reduces the design and troubleshooting time significantly.

- *Machine learning (ML) needs higher computations costs.* ML needs massive amounts of data to learn and generally has higher computation costs compared to traditional chipless readers. Data should be kept safe but also has to be shared with other readers if needed. Cloud storage/computing is an answer to all of these challenges, as will be described later.

As we will see in Chapter 4, the detection method in this book is based on pattern recognition and machine learning, which requires massive datasets for the training, testing, and verification phases. A large combination of different tags can be made by different combinations of individual alphanumeric tags (so a letter tag of "5ENST" has a different frequency response to "ES5ET"). Another advantage of using letters in this particular application comes down to reader hardware architecture. The proposed reader in Chapter 3 uses separated transmitter (Tx) and receiver (Rx) antennas (bistatic) in a cross-polar configuration. This configuration provides the advantage of suppressing copolar waves, but at the same time, the tags need to have components in both the x- and y-directions, as otherwise no cross-polarization scattering reflection is possible. Letters normally have components in both polarizations. And last but not least, letters are human- and machine-recognition friendly. This feature is quite important in a tag's double decoding verification, which is covered in Chapter 5.

Figure 1.8 shows a generic response of the typical low-cost tag. Detecting these highly underdumped resonant frequencies is difficult and also lowers the encoding bits in the frequency response. In general, for a specific bandwidth and detection algorithm, the higher the Q-factor the more encoding capacity there is. Using letters on paper and plastic substrates can lead to very low Q-factors and therefore less encoded bits in the frequency response.

Table 1.1 summarizes the current challenges in both frequency and image-based tags and their proposed remedies. The following section presents the approaches to mitigate these challenges.

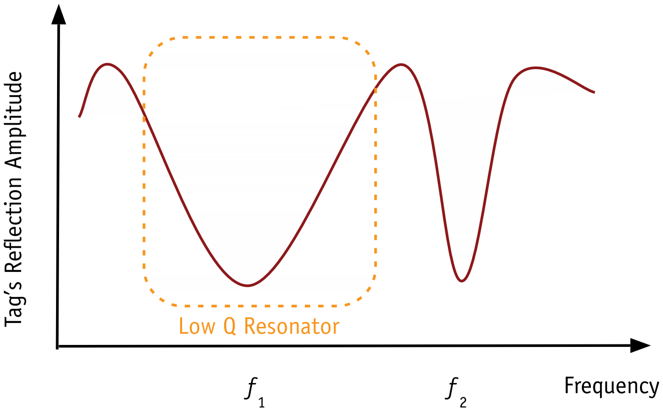


Figure 1.8 Chipless tags on paper and plastic substrates have low-Q resonator responses. The sharper the null in the frequency response, the higher the Q factor.

Table 1.1
Challenges in the Chipless RFID Domain and Their Proposed Solutions in This Book

Challenges	Details	Proposed Solutions
1. Tag detection	Not enough backscatter signal from the letter plastic tag	Use 5-letter tags to increase RCS. Alphanumeric tags can be also better detected in cross-polarization modes, where the aim is to suppress in-band inteference (Chapter 2)
2. Data encoding	Fewer data can be stored in plastic tags as Q-factor is low	Tags are detected by their difference in the response patterns not their nulls. Letter-based optimization is done to make tags more distinguishable. A 6 bits/cm^2 capacity per letter, or 30 bits for 5-letter tag is expected (Chapter 2)
3. Orientation insensitivity	Letter tags are not symmetrical	Using ML helps with mis-orientation reading (Chapter 4)
4. Mis-decoding	tagIDs might be decoded wrongly	Using two parallel decoding methods: image-based and frequency-space pattern recognition will reduce detection errors (Chapter 4)
5. 60-GHz reader hardware	RF design in mm-wave is difficult	Using a modular hardware design approach (Chapter 3)
6. Expensive processing and data storage per reader	ML and image processing need powerful computers, and data sharing between readers is essential	Using cloud storage/computing to share the data between readers and reduce processing cost per device (Chapter 5)

1.7 Book Outline

The book is divided into six chapters.

This chapter has listed the current problems in the chipless RFID domain by focusing on the 60-GHz spectrum, introduced the relevant research questions, and has made the goal of this book as using intelligent techniques in the tag decoding phase.

Chapter 2 describes the first part of the system, designing chipless tags in the mm-wave spectrum. After a short introduction, the design approach to letter-based tags, including backscattering theory and efficient tag designs, are discussed. This includes explanation of the tag's orientation dependency and data encoding capacity. Simulations are introduced for optimizing individual letter responses and choosing the best-performing combination of letters. RCS analysis and tag post-optimization are discussed. The findings are outlined in the conclusion.

Chapter 3 describes the second part of the system, the modular designed reader. After the introduction, the most important design considerations are highlighted, namely, link budget and calculating the maximum radiation power and reading distance, as these have the biggest impact on designing RF parts. After selecting the 60-GHz module, the chapter describes different components of the design. These include digital controlling board components, local oscillator (LO), and peripherals such as linear rail controllers and keyboards. For those modules, the integration problems are identified and solution steps are discussed. After integration of the whole board, final parameters of the system are discussed, which are the scanning time, frequency resolutions, and reader calibration. The chapter concludes with some brief results of the technical difficulties and possible improvements.

Chapter 4 is about using artificial intelligence (AI) and side-looking airborne radar (SLAR) implementation for the third part of the system, which is tag detection and decoding the tag's encoded data. Tag decoding techniques are closely related to the type of chipless tag developed in Chapter 2. This chapter starts with an introduction section, describes how system-level decoding simulations are carried out in MATLAB® and CST®, and how the experimental method is done for data collection. The AI decoding techniques start with feedforwarding networks and expand to recent pattern recognizers such as k-nearest neighbors (KNNs) and support vector machines (SVMs). This chapter also addresses the main problem of chipless RFID decoding, which is uncertainty of the results and misdecoding. A method of SLAR is developed in the next section to make a low-resolution image of the tag, which is combined with pattern classifiers to lower the misdecoding rate. The chapter ends with conclusions about both methods.

Chapter 5 describes expanding the capability of the reader developed in Chapter 3 and the decoding algorithms in Chapter 4. After an introduction, the overall architecture of the proposed cloud system is described, with the methodology and deep-learning architecture.

Chapter 6 summarizes the methods of the enhanced alphabetic tag designs in Chapter 2, the reader in Chapter 3, and algorithms in Chapters 4 and 5.

To make it easier to grasp the concept in simple terms, every chapter ends with an informal question and answer section. These were questions that came to us as we developed each chapter.

References

[1] Indvik, L., "Why Luxury Brands Are Putting Microchips in Your Clothes and Accessories," *Fashionista,* April 2016, https://fashionista.com/2016/04/moncler-ferragamorfid-counterfeiting.

[2] Solon, O., "Internet of Things: RFID-Powered Shoes Connect Wearers to Social Media," *Wired,* 2011, https://www.wired.co.uk/article/rfid-shoes.

[3] Smiley, S., "What Can RFID Laundry Tracking Do for You?" atlasRDID-store, March 2015, https://blog.atlasrfidstore.com/what-can-rfid-laundry-tracking-do.

[4] Nicasio, F., "6 Anti-Theft Devices You Can Use to Protect Your Retail Store," 2018, https://www.vendhq.com/blog/anti-theft-devices/.

[5] AB&R, *Evolution of RFID,* 2016, https://www.abr.com/evolution-of-rfid/.

[6] Violino, B., "The History of RFID Technology," *RFID Journal,* 2016, http://www.rfidjournal.com/articles/view?1338/2.

[7] Karmakar, N. C., "Tag, You're It Radar Cross Section of Chipless RFID Tags," *IEEE Microwave Magazine,* Vol. 17, No. 7, pp. 64–74, 2016, doi:10.1109/MMM.2016.2549160.

[8] Arjomandi, L. M., G. Khadka, Z. Xiong, and N. C. Karmakar, "Document Verification: A Cloud-Based Computing Pattern Recognition Approach to Chipless RFID," *IEEE Access,* Vol. 6, 2018, pp. 78 007–78 015.

[9] Roland, M., and M. Hölzl, "Evaluation of Contactless Smartcard Antennas," *arXiv preprint arXiv:1507.06427*, 2015.

[10] Li, H., B. Wang, M. Wu, J. Zhu, and C. Zhou, "Design and Analysis of Chipless RFID Tags Based on Retro-Radiators," *IEEE Access*, Vol. 7, 2019, pp. 148 208–148 217.

[11] Islam, M. A., and N. C. Karmakar, "Compact Printable Chipless RFID Systems," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 63, No. 11, 2015, pp. 3785–3793.

[12] Karmakar, N. C., P. Kalansuriya, R. E. Azim, and R. Koswatta, *Chipless Radio Frequency Identification Reader Signal Processing*, John Wiley & Sons, 2016.

[13] Herrojo, C., F. Paredes, J. Mata-Contreras, and F. Martın, "Chipless-RFID: A Review and Recent Developments," *Sensors*, Vol. 19, No. 15, 2019, p. 3385.

[14] Nguyen, D. H., M. Zomorrodi, and N. C. Karmakar, "Spatial-Based Chipless RFID System," *IEEE Journal of Radio Frequency Identification*, Vol. 3, No. 1, 2019, pp. 46–55, doi:10.1109/JRFID.2018.2887162.

[15] Forouzandeh, M., and N. C. Karmakar, "Chipless RFID Tags and Sensors: A Review on Time-Domain Techniques," *Wireless Power Transfer*, Vol. 2, No. Special Issue 02, 2015, pp. 62–77, doi:10.1017/wpt.2015.10.

[16] MathWorks, *Synthetic Aperture Radar (SAR) Processing*, https://au.mathworks.com/help/dsp/examples/synthetic-aperture-radar-sarprocessing.html.

[17] Özdemiir, C., *Inverse Synthetic Aperture Radar Imaging with MATLAB Algorithms*, John Wiley & Sons, 2012, doi:10.1002/9781118178072.

[18] Chen, V. C., and M. Martorella, *Inverse Synthetic Aperture Radar Imaging: Principles, Algorithms and Applications*, Institution of Engineering and Technology, 2014, doi:10.1049/SBRA504E.

[19] Li, J., and P. Stoica, *MIMO Radar Signal Processing,* Hoboken, NJ: Wiley-IEEE Press, 2009.

[20] Zomorrodi, M., *mm-Wave EM-Imaging Chipless RFID System,* PhD dissertation, Monash University, 2015.

[21] Herrojo, C., M. Moras, F. Paredes, et al., "Very Low-Cost 80-Bit Chipless-RFID Tags Inkjet Printed on Ordinary Paper," *Technologies*, Vol. 6, No. 2, 2018, p. 52.

[22] Vena, A., E. Perret, S. Tedjini, et al., "Design of Chipless RFID Tags Printed on Paper by Flexography," *IEEE Transactions on Antennas and Propagation*, Vol. 61, No. 12, 2013, pp. 5868–5877.

[23] Nimmi, M., S. Meenakshi, and R. Priyadarshini, "RFID Location System Based on Artificial Neural Networks," *International Journal of Computer Communication and Information System (IJCCIS)*, Vol. 2, No. 1, July 2010.

[24] Das, R., and P. Harrop, *RFID Forecasts, Players and Opportunities 2018-2028. The Complete Analysis of the Global RFID Industry,* https://www.idtechex.com/en/research-report/rfid-forecasts-players-and-opportunities 2018-2028/642.

[25] *RFID (Radio Frequency Identification) Technology Market Revenue Worldwide from 2014 to 2025*, 2018, https://www.statista.com/statistics/781338/global-rfid-technology-market-revenue/.

[26] Wikipedia, "ISM Radio Band," August 2022, https ://en.wikipedia.org/wiki/ISM_radio_band.

[27] W. S. 4. of the Internet of Things Alliance Australia, *Spectrum Available for IoT*, https://www.iot.org.au/wp/wp-content/uploads/2016/12/IoTSpectrumFactSheet.pdf.

[28] Murph, D., "WiGig Alliance Completes Multi-Gigabit 60 GHz Wireless Specification: Let the Streaming Begin," *Engadget,* 2009, https://www.engadget.com/2009/12/10/wigigalliance-completes-multi-gigabit-60ghz-wireless-specificat/.

[29] Liebe, H. J., *An Atmospheric Millimeter Wave Propagation Model,* U.S. Department of Commerce—National Telecommunications and Information Administration, Tech. Rep., 1983.

[30] P. Series, "Attenuation by Atmospheric Gases and Related Effects," *Recommendation ITU-R*, 2019, pp. 676–12.

[31] "Why Is the 60 GHz Band Not Good for Long-Range Communications?" *Everything RF,* August 2017, https://www.everythingrf.com/community/why-is-the-60-ghz-band-not-good-for-long-range-communications.

[32] Harris, W., "How Millimeter Wave Scanners Work," *HowStuffWorks,* https://science.howstuffworks.com/millimeter-wave-scanner1.htm.

[33] Singh, T., S. Tedjini, E. Perret, and A. Vena, "A frequency Signature Based Method for the RF Identification of Letters," in *2011 IEEE International Conference on RFID*, 2011, pp. 1–5.

[34] Tedjini, S., O. Boularess, T. Andriamiharivolamena, H. Rmili, and T. Aguili, "A Novel Design of Chipless RFID Tags Based on Alphabets," in *Microwave Symposium (IMS), 2017 IEEE MTT-S International*, IEEE, 2017, pp. 1561–1563.

[35] Boularess, O., H. Rmili, T. Aguili, and S. Tedjini, "Analysis of Electromagnetic Signature of Arabic Alphabet as RF Elementary Coding Particles," *Wireless Power Transfer,* Vol. 2, No. 2, 2015, pp. 97–106.

2

Chipless Tag Design

2.1 Introduction

Chipless RFID tags are the first part of any RFID system. In this chapter, we go through the design concepts and implementation of mm-wave chipless tags. Figure 2.1 shows a simplified illustration of a chipless RFID system.

There are five major considerations for designing tags in the microwave and mm-wave:

- Having maximum backscatter;
- Having high encoding capacity;
- Using cheap substrate and printing techniques (the final tag price should be far less than a cent);
- Tag flexibility;
- Symmetrical design (for orientation-insensitive tag reading).

Chipless tag detection is based on a backscattered signal from the tag to the reader, and if the signal level is below a certain threshold compared to the surrounding area, the tag might not be detected. Encoding capacity is normally defined as number of bits in each cm^2, which should be as high as possible. Having a cheap

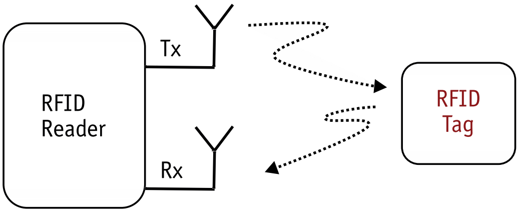
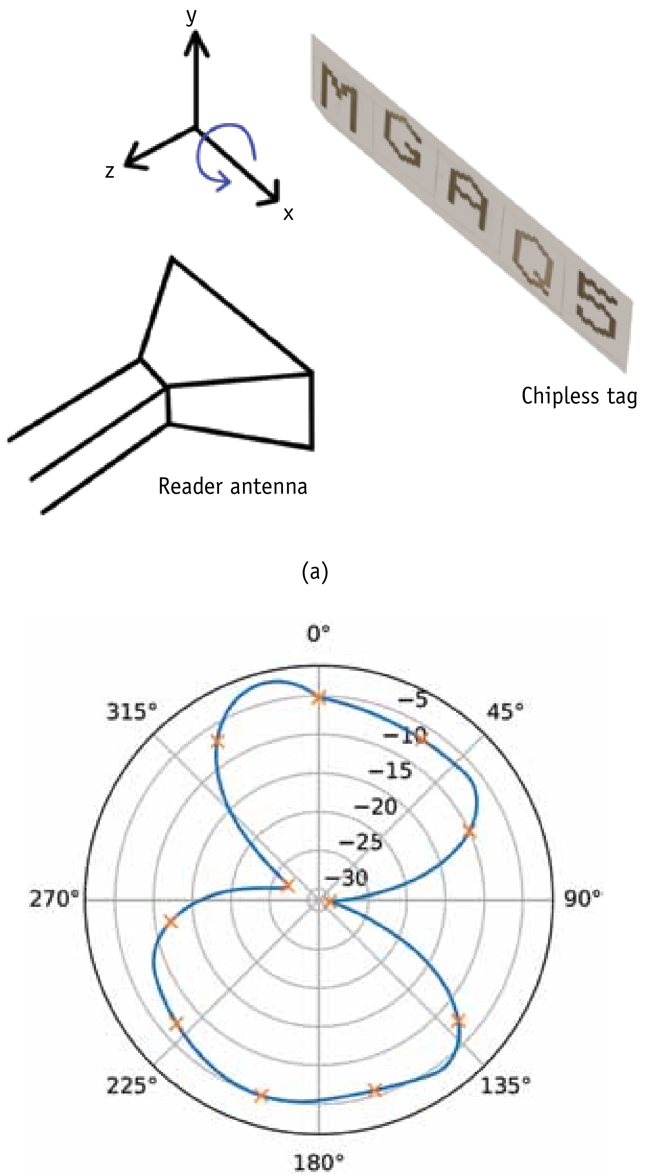


Figure 2.1 A simplified illustration of a chipless RFID system. Chipless tags, the focus of this chapter, act as backscatters and data encoders simultaneously.

and flexible substrate and printing technique to make cheap tags in high volume is a big requirement. A tag's flexibility by itself is also not dealt with in much of the research, but it is a main requirement in some applications in the real world. Having tags with a symmetrical design results in the same frequency response once the tag is read from different angles perpendicular to the tag's plane.

It is very important to note that all passive tags (tags that rely on reader power) suffer from the reading angle blindness problem as well. This means that the received backscatter signal will be highly attenuated if the reader view angle to the tag is not perpendicular. An illustration of this concept is shown in Figure 2.2. The plot of Figure 2.2(b) is based on the assumption that the tag has no grounding plane in the back of the tag. Although having a grounding plane is useful to reduce electromagnetic interference (EMI) and cross talk and results in higher-Q nulls, it has the disadvantage of the tag not being readable in some angles if the reader is facing the tag's ground plane. The code for the plot in Figure 2.2(b) is available from the authors' repository [1].

Conventional tags and ones available in the literature and market [2–5] cannot achieve all of the desired requirements at once but involve a compromise between tag cost, encoding capacity, size, and symmetrical design. In this chapter, we first go through the details of different chipless tags, and then introduce a new design method for chipless tags in the frequency (and image) domains. Our frequency tags are designed based on letters and symbols (alphanumerical), and will be part of a bigger RFID system that uses AI for decoding. The main goal is to use cheap and flexible substrates such as plastic and paper for mm-wave chipless RFID tag design with a target of obtaining a subcent price tag. This chapter demonstrates the design procedure for the letters so that



(a)

(b)

Figure 2.2 An illustration of angular sensitivity in RFID systems. In (a), the reader is shown perpendicular to the tag. If the tag rotates along the x-axis as shown by (a), the backscattered signal from the tag to the reader will have an attenuation similar to the measured pattern illustrated by (b). The attenuation values shown are typical for tags with no ground planes [1]. The tag has no ground plane; otherwise the received data from the tag between 90° toward 270° would be almost unreadable.

desired backscatter frequency response is achieved from the tag while enough backscatter signal and encoding capacity is assured.

The chapter is organized as follows: After this introduction, Section 2.2 reviews different tag categories, such as time-domain, frequency-domain, and image-based. Since this book is based on alphanumerical tags, this review will include letter-based tags and their printing techniques as well. Section 2.3 discusses the alphanumerical tags, including their encoding capacity and frequency response. A bit of handful backscattering theory and radar cross section (RCS) is discussed in Section 2.4, as these will be needed later for calibration purposes in the next chapter. Section 2.5 discusses the simulation procedure for alphanumerical tags, and their experimental results are shared in Section 2.6, followed by the conclusion.

2.2 Chipless RFID Tags

As mentioned before, the chipless tag detection on the reader side is based on the fact that tags act simultaneously as backscatters and data encoders. Based on the detection method, this backscattered signal is processed and the tag's data is extracted.

As shown in Figure 1.2, there are four categories of chipless tags: time-domain, frequency-domain, image-based, and chemical [6]. There are also hybrid tags, which are combinations of these methods, such as phase deviation-frequency [7], impedance-loading [8], polarization-phase, or space-frequency [2]. Chemical tags are not as common as the other types. Since chipless tags are still in the inception and development phases, there is no standard available for these types of tags. The only commercially successful chipless tags are SAW tags in the time-domain tags category, which is described in the following section, along with the other two main categories of tags, highlighting their advantages and disadvantages.

2.2.1 Time-Domain Tags

In time-domain tags, which are also referred to as time-domain reflectometry (TDR) tags, the tag reflects the incoming wave through some discontinuities or delays. The return signal from the tag has some trains of delayed pulses that correspond to the position of these discontinuities [9, 10]. SAW tags fit into this category. Figure

2.3 shows a schematic of a typical SAW tag. These tags consist of an IDT plus an antenna, piezoelectric (or diamond) surfaces [11], and a few reflectors that act as the encoders. The IDT structures attach to the piezoelectric substrates and convert electrical signals to mechanical vibrations and vice versa. The slot size for every bit is limited by the time difference the reader can distinguish between consecutive received pulses (25 ns in this case) [12].

There are several ways to encode data in time-domain SAW tags [13]. In on-off keying (OOK) modulation, for example, the reflection comes from capacitive impedance mismatches. The presence or absence of signals within a predetermined duration of time is considered as 0 or 1. In the pulse position modulation (PPM) technique, *n*-bit encoding is achieved by dividing each time slot into 2*n* pulse positions.

Although time-domain chipless RFID is a well-known concept, it is plagued with technical difficulties such as an inability to

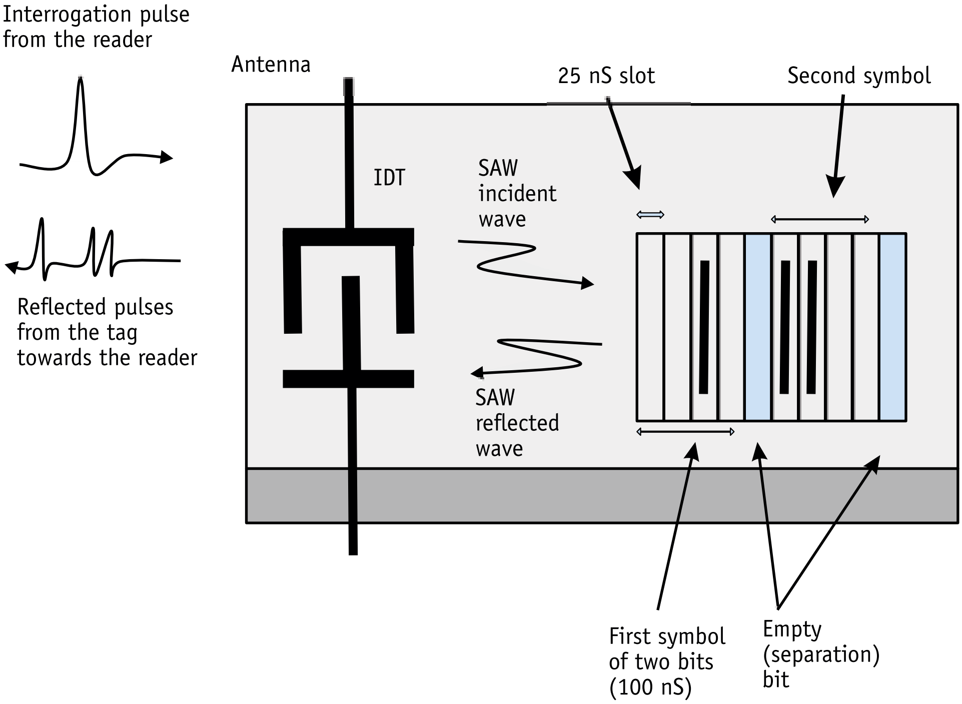

Interrogation pulse from the reader
Reflected pulses from the tag towards the reader
Antenna
IDT
SAW incident wave
SAW reflected wave
25 nS slot
Second symbol
First symbol of two bits (100 nS)
Empty (separation) bit

Figure 2.3 Time-position encoding in SAW substrates. The bytes (4 bits) are separated by an empty bit, as shown in blue. Each time slot creates a 25-nS delay, and 1s and 0s are represented by the presence or absence, respectively, of a metal bar.

make a sharp-edge interrogation signal in the reader, very rapid sampling rate requirements in A/D converters, and the need for UWB active and passive components. The SAW substrate is also too expensive for mass production of chipless tags.

2.2.2 Frequency-Domain Tags

The main purpose of frequency (or phase) domain tags is to produce resonant frequencies (or shifts in the phase) that are suitable for data encoding. In the frequency domain, the RFID reader sends a broadband signal, a chirp, or several distinct continuous wave (CW) frequencies, and listens to the echoes from the tag. The attenuated or missing information in the response signal indicates the presence of encoded data bits. This happens because the information is stored in the frequency signatures or resonant cavities. Frequency-based tags are so common that almost every tag not in the time domain will fall into this category.

As with time domain, there are many tag designs within the frequency domain. Space-filling tags, which use Peano or Hilbert curves to fill out gaps to reach to lower resonant frequencies in a smaller footprint [14], and spiral resonator tags, which are composites of a spiral passive filter and two monopole antennas [15], are examples of tags using the frequency domain.

Frequency-domain tags are the most prevalent type found in the chipless RFID arena. This is because of the simplicity of both the tags and the readers. Their disadvantages are substrate costs, lower Q-factor that reduces number of encoded bits, and frequency and amplitude change if the relative position of the tag to the reader changes. Figure 2.4 shows an example of frequency tag response. Higher Q-factors nulls and peaks result in more bits to be detected in any determined frequency range.

We present a few samples of frequency-based tags. Figure 2.5 shows a circular patch with slot resonators. Circular patch tags have the advantage of orientation/rotation independency from the reader to the tag. Moreover, the dominant spurious frequency of the circular patch is their third harmonic, in contrast to the majority of the frequency tags that have dominant second harmonics [16]. The fundamental resonant frequency of a circular patch with radius *r* is given by [17]:

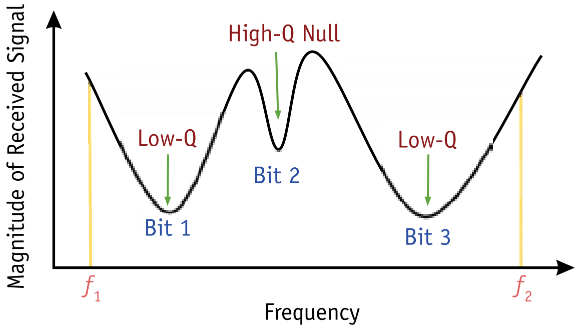


Figure 2.4 A typical response of the tag in the frequency domain, with high-Q and low-Q nulls (encoded bits).

$$f = \frac{1.841c}{2\pi r_e \sqrt{\varepsilon_r}} \tag{2.1}$$

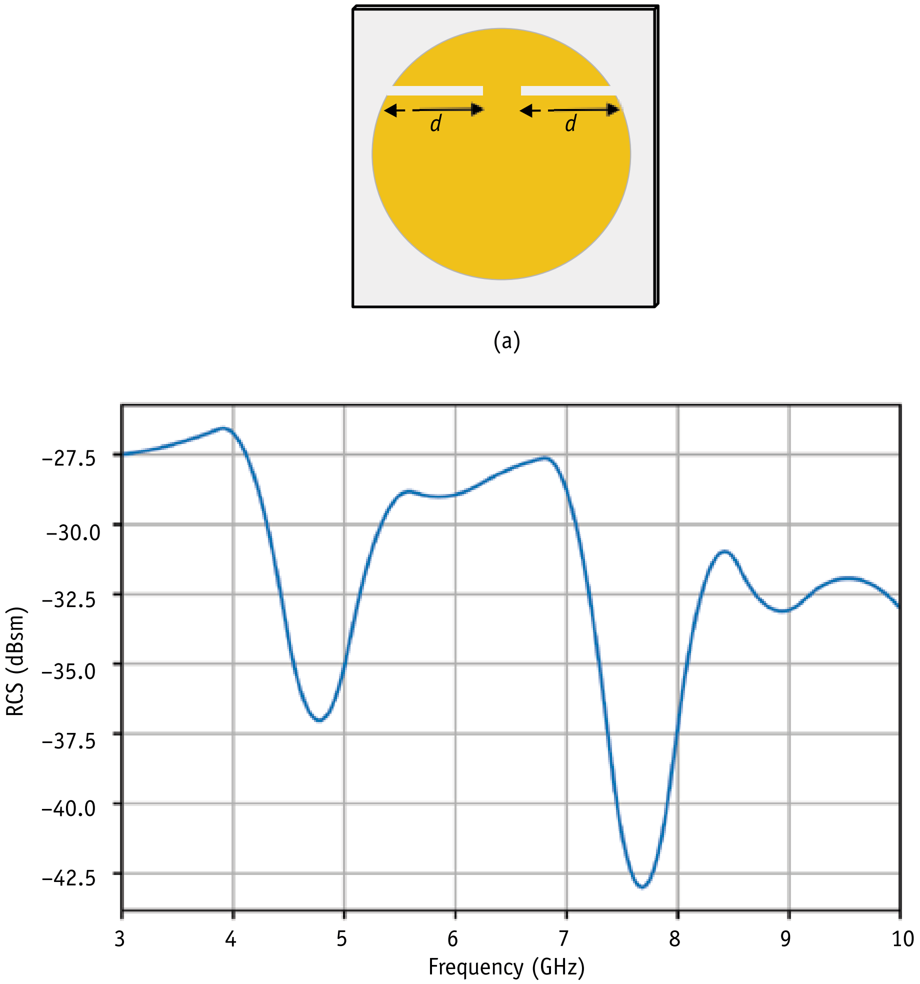
in which c is the speed of the light in the free space, ε_r is the relative permittivity of the substrate, and r_e is the effective radius of the circular patch which is given by [18]

$$r_e = r\sqrt{1 + \frac{2h}{\pi r \varepsilon_r}\left(\ln\left(\frac{\pi r}{2h}\right) + 1.7726\right)} \tag{2.2}$$

where h is the height of the substrate. In addition, putting two symmetrical quarter-length slots within the circular tags makes an extra bit [19]. To make more bits, more circular patches with different radii can be used, and/or more quarter-length added slots within circles.

2.2.3 Image-Based Tags

Although the technology for image-based tags is derived from SAR, which dates back to the 1950s, it is considered a new method in the chipless RFID domain. The concept of image-based tags is similar to airborne radars, where successive pulses are used to illuminate the target. The reader (radar) position changes compared to the tag (target) in a linear line. The resultant echoes from the tag are processed based on the relative position and angle of the reader to the tag to make an image of the tag. The resultant image



(a)

(b)

Figure 2.5 (a) A 2-bits quarter-wavelength slots circular patch with their typical planar incident wave perpendicular to the tag's surface response of (b). The radius of the patch and the length of the "d" slot in (a) are the causes of the two nulls in (b) [16, 19, 20].

resolution depends on the antenna's aperture size. This aperture size can become smaller using MIMO antennas and the concept of virtual aperture [21, 22]. A resolution of a few millimeters for airborne earth images was reported using this technology [23].

Image-based tags are basically image-friendly frequency-based tags that are scanned in a few angles. There are not many known image-based tags available in the literature, but two can be found in the following: a microwave readable dielectric barcode with a minimum length of $\lambda/2$ [24], and 45° stripline and meander-line polarizer tags by Zomorrodi et al. [25]. In Figure 2.6, the concept of a 4-bit tag is shown using meander lines. Each two-line pair represents one bit, which is done to increase the RCS, and cross-polarized antennas with the tags oriented 45° opposite from the reader are used to lessen the coupling effect.

For chipless tag imaging to be functional, frequency scanning of the tag should be done at different angles. One simple way of doing this task is putting the tag in a linear rail, moving it in front of Tx and Rx antennas, and capturing the frequency response in each position. Another method is using MIMO with SAR. MIMO-SAR

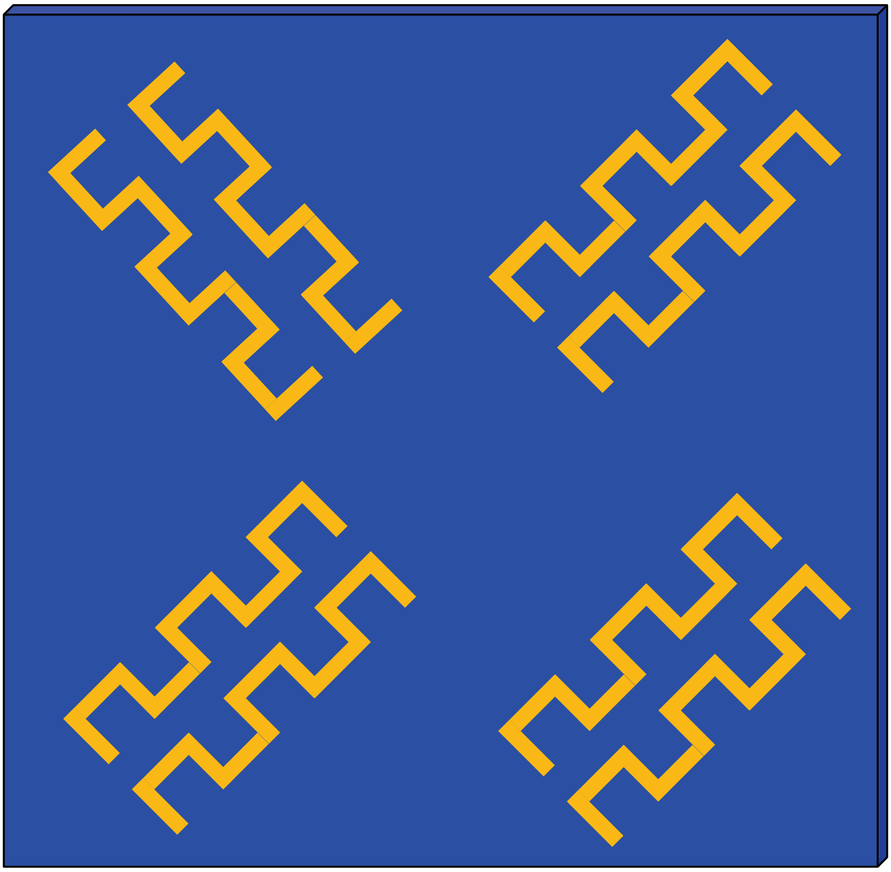


Figure 2.6 A 4-bit image-based tag using meander lines. Each meander line total length is $\lambda/2$ and the 45° orientation is used for a cross-polarized reader [26, 27].

is MIMO added to SAR technology to make the reading process easier and faster. There is no need to accommodate relative movements for the tags toward reader antennas in this scenario, as different angle readings are done simultaneously. A MIMO reader, similar to MIMO radar, transmits mutually orthogonal signals from multiple transmit antennas, and these waveforms can be extracted from each of the receive antennas by a set of matched filters. As an example, if a MIMO reader has two Tx antennas and three Rx antennas, a total of six different waveforms can be extracted because of the orthogonality of the transmitted signals [28]. Figure 2.7 illustrates the concept of linear rail and MIMO reading of the tags. Deploying MIMO normally means less physical movement of the tag, which is a great advantage.

In MIMO, the concept of virtual phase centers is defined, in which the location of the virtual transmitters/receivers are determined by the spatial convolution of the physical location of the transmitters/receivers. An illustration of phase centers in MIMO is shown in Figure 2.8. If the vector location of transmit and receive antennas are represented by $\mathbf{h}_{Tx}$ and $\mathbf{h}_{Rx}$, respectively, the placement vector of virtual phase centers would be [28]:

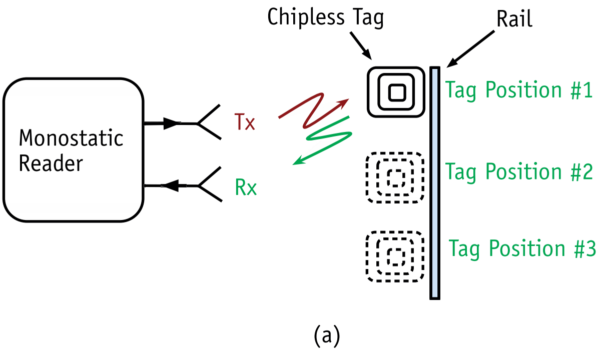
$$\mathbf{h}_{MIMO} = \mathbf{h}_{Tx} * \mathbf{h}_{Rx} \tag{2.3}$$

Although MIMO-SAR is an enhanced version of SAR, which eliminates relative antenna-to-tag movement requirements and facilitates faster microwave image capturing, it is coupled with a demand for more complicated hardware and higher synchronization and processing power to handle the orthogonal signals. These facts make MIMO reader implementation on handheld devices challenging.

2.2.4 Letter-Based Tags

Letter tags can be generally considered as frequency-based tags, but they are used in this book both as pure frequency-based and also as frequency-position based, enabling a developed SLAR algorithm to create their images. Because of their prominent use in this book, a brief background on letter tags is presented.

Letters have been used as tagIDs in the frequency domain since 2011. Singh et al. [30] showed how some individual letters



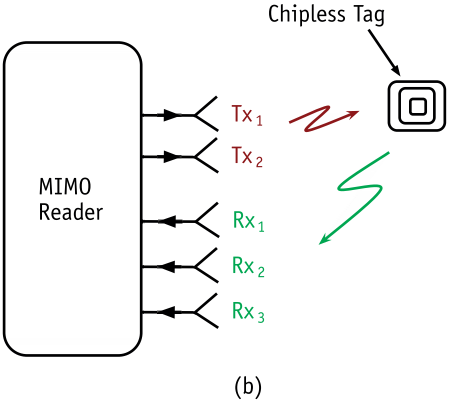


Figure 2.7 Multiple tag-reading scenario for image processing using (a) linear rail and (b) MIMO technique. In each position in (a), the tag is being read and then moved to the next position using a linear rail. For a three-position scenario, three-frequency capturing of the tag can be done. In (b), two transmitters send the orthogonal signals simultaneously and each receiver will get two signals, with a total of six signals received. Less or no tag movement is needed in the MIMO scenario.

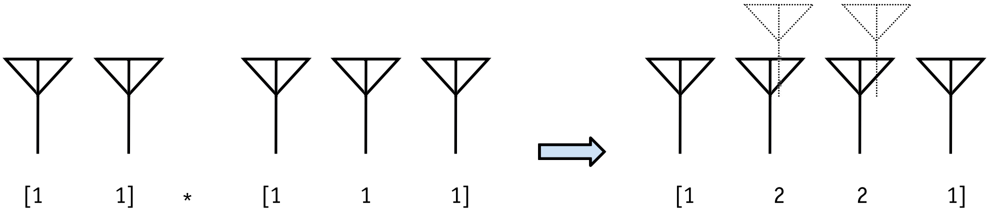


Figure 2.8 An illustration of the virtual phase center concept in MIMO. The location of the virtual transmitters/receivers are determined by the spatial convolution of the physical location of the transmit/receive antennas [29].

with calculated dimensions had dominant peak resonances in some specific frequencies. Tedjini et al. [31] used letters on a Taconic TLX-8 substrate, and showed that each letter could create specific frequency resonances. Their tag concept is shown in Figure 2.9. A tag is read in the vertical and horizontal planes. It is shown in their simulations that the letters can be decoded based on the frequency spectrum response of two copolar measurements: horizontal plane wave excitation with horizontal probe (HH) and vertical plane wave with vertical probe (VV). Boularess et al. [32] examined the frequency pattern of Arabic letters in a two-polarization configuration using a flexible Kapton substrate, and concluded that if both polarizations were used, the individual letters could be identified accurately. The tags were chosen from a set of 28 Arabic alphabets and the tag sizes were 37 by 37 mm^2.

In this book, alphanumerical tags are used as they can produce many different combinations in a five-letter tag, which is required to train AI networks. Less emphasis will be placed on a tag's null Q-factors, as the only requirement will be how the frequency responses of different tags are *distinguishable* from each other. These design concepts will be more expanded in Section 2.3.3.

A comparison of available letter-based frequency-domain tags and image-based tags is shown in Table 2.1. Tags are in time, frequency, or image domains. Their *data density*, which is defined as number of encoded bits per cm^2, is an important comparison column. It is important if the tags can be *printed* on low-cost substrates, such as paper or plastic, and if they are *flexible* enough to be attached as a label to the objects, similar to barcode or QR codes. The *expandable* column shows whether adding several tags would

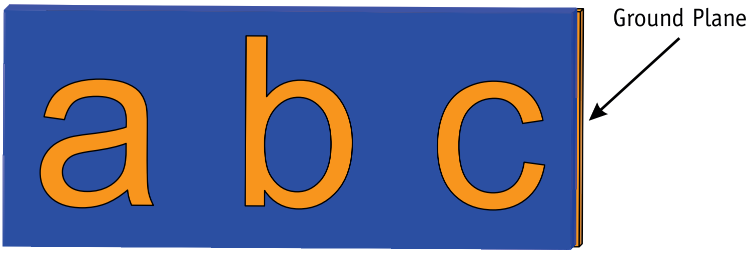


Figure 2.9 A letter tag on a ground plane. Letter sizes are selected to obtain resonance peaks in the frequency band of 6–10 GHz [30, 31]. Using a ground plane makes higher-Q nulls in the frequency response of the tag but restricts the reading angles as well.

Table 2.1

Comparison Table for Available Chipless Tags with Emphasis on Frequency- and Image-Based Tags

Domain	Type	Frequency (GHz)	Data Density (bits/cm^2)	Printable	Flexible	Expandable	Orientation-Independent
Time	SAW [34]	2.44 (F.M.)	>1	No	No	Yes	Yes
Time	TDR [35]	3.1–10.6	0.17	Yes	No	Yes	No
Frequency	Space filling curve [14]	3-4.5	0.55	Yes	No	Yes	No
Frequency	Multiband dipole [36]	2.4-2.5; 5-6	0.81	Yes	No	Yes	No
Frequency	Elliptical dipole [37]	3–14	0.36	Yes	No	No	No
Frequency	LC resonant [38]	5–8 MHz	1	Yes	Yes	Yes	No
Frequency	Split-ring resonator (SRR) array [39]	8–12	2.8	Yes	No	Yes	No
Frequency	Coplanar strip [40]	2–6	3.3	Yes	No	No	No
Frequency	Letter ID [30]	1–10	5	Yes	No	No	No
Frequency	Letters [31]	6–12	>3.1	No	No	No	No
Frequency	Arabic letters [32]	2–8.4	2.1	No	No	No	No
Frequency	Spiral resonator [41]	3–7	0.61	No	No	Yes	No
Frequency	Stub resonators [42]	2–4	0.16	No	No	Yes	No
Frequency	Multiple-resonant dipole antenna (MRDA) [43]	3–7	0.16	No	No	Yes	No
Frequency	Fractal [15]	1.8–2.8	0.03	No	No	Yes	No
Frequency	Slotted monopole [44]	3–7	1	No	No	No	No
Frequency	Wheel resonator [45]	3–6	1	Yes	No	Yes	No
Frequency	C-shaped resonator [46]	2–4	1.14	Yes	No	Yes	No
Frequency	THz tag [47]	100–900	3.5	No	Yes	No	NG
Frequency	BW division [48]	2–5	3.6	No	No	No	No
Phase	Stub loaded microstrip patch antenna (SLMPA) [49]	2.1–2.6	0.18	No	No	Yes	No
Phase	C-phase resonator [7]	2.0–3.5	2.88	Yes	No	Yes	No
Phase	Multilayer patch [50]	5.4–7.4	0.61	No	No	Yes	NG

Table 2.1 (continued)

Domain	Type	Frequency (GHz)	Data Density (bits/cm^2)	Printable	Flexible	Expandable	Orientation-Independent
Freq-polarization	Dual-polarized slot-loaded [52]	6–12	5.22	No	No	Yes	No
Freq-polarization	Hybrid dual Pol [4]	2–8	1.56	No	No	No	No
Image-based	SAR [26]	NG	>1	Yes	Yes	No	No
Image-based	Barcode [24]	10	NG	Yes	NG	Yes	No
Image-based	45° meander lines [10]	57–64	2	Yes	No	Yes	No
Freq-image	Letters [53]	57–64	6	Yes	Yes	Yes	No

F.M. = fundamental frequency; NG = not given.

result in higher encoding bit tags. The last row in the table shows whether the tags can be read correctly in different reader-to-tag angles. Table 2.1 is mainly based on the work of our former and current colleagues [4, 33], with additional columns and recently developed tags.

In Table 2.1, in the time domain, SAW and TDR tags are used for the comparison. The frequency range column has many tags, such as space-filling curves, MRDA, SLMPA, and multilayer patch. Phase domain is part of frequency tags, but that information is coded in the different part of the phase of the signal. There are hybrid tags, which combine frequency and phase to gain higher encoding capacities. Finally, image-based tags are a subdomain of frequency tags, in which frequency and spatial data are combined to make a 2-D representation of the tag.

The developed tags in this book are alphanumeric tags, which are scanned in the frequency domain and treated as frequency tags, and they will be image-scanned in Chapter 4, so they are freq-image tags, which is the last row of Table 2.1. Although the proposed tags, which are alphanumeric, are not symmetrical and hence would not have orientation independency, but the nature of the detection method (AI) would help to detect the misaligned tags in some content, as will be shown in Chapter 4.

The developed tags in this book are also screen-printable on low-cost substrates, have a data capacity of 6 bits/cm^2, and the combination of letters can multiply the encoding capacity. As AI and tag 2-D image formation using SLAR is used, the tags can be read from different angles with acceptable accuracy.

2.2.5 Screen Printing for the Chipless Tags

In this book we used alphanumerical tags on PCB and screen printing, which is why screen printing is briefly discussed here. Screen printing is one of the earliest (and relatively slowest) printing technologies used for printed electronics, including chipless tags. Screen printing produces the thickest printed layers in the order of 3–300 μm, which results in higher conductivity compared to the other printing techniques. These other suitable printing techniques are [54–56]:

- *Gravure printing,* which transfers inks from small engraved cavities on a gravure cylinder to a printing substrate, and is mainly used for magazines and newspapers;
- *Flexographic printing,* which uses a flexible relief plate (typically made of rubber) to print to the substrate;
- *Inkjet printing,* in which the ink streams directly from the nozzles in the printer onto the substrates.

Screen printing uses a mesh to transfer the ink from the screen onto the substrate. The mesh screen contains an emulsion that exposes the tag pattern to be deposited. First, the mesh is covered with conductive ink that is gently distributed using a squeegee. This is called the flooding phase. Second, the substrate is placed under the screen and the ink is pushed through from the top of the mesh to the bottom. The ink is transferred from the mesh to the substrate only in areas that are permeable. This is called the printing phase. The substrate should then be placed in an industrial oven (sintered) for a predefined period of time. Figure 2.10 illustrates the flooding and printing process. To obtain a uniform print on the substrate, printing pressure and ink viscosity should be carefully adjusted [57, 58]. The effect of heating on metallic ink tracks is shown by Figure 2.11.

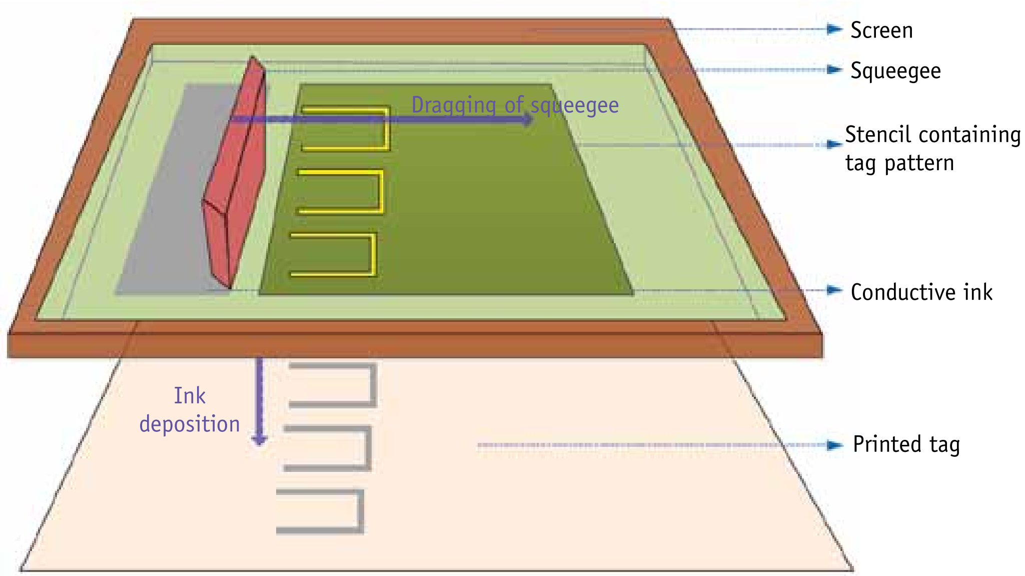


Figure 2.10 Screen-printing process for chipless RFID tags [57]. (Reprinted with permission.)

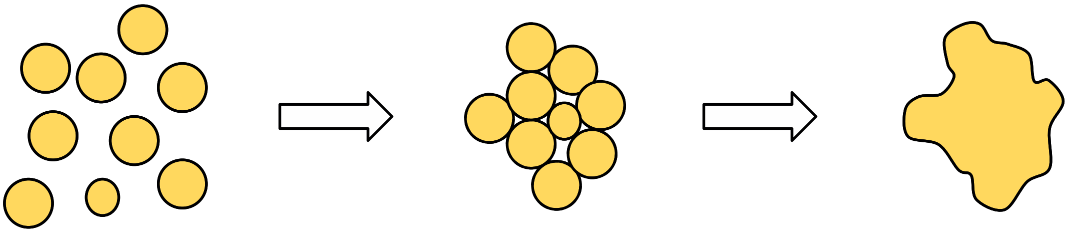


Figure 2.11 Heat sintering process in which metallic tracks attach together and form a connected shape [54].

2.2.6 Screen Printing Experimental Observations

Figure 2.12 illustrates one of the screen printings used in this book. Film is attached to a metal frame before being covered by the different conductive inks.

Several ways of implementing the tags are shown in Figure 2.13. The tags were printed on 100-μm Mylar polyester substrate film using two different inks, conductive InkTec TEC-PR-041 [59] and M-creative 118-09A [60]. They were both sintered at 140° for 20 minutes [57]. Here are some of the observations about the screen printing:

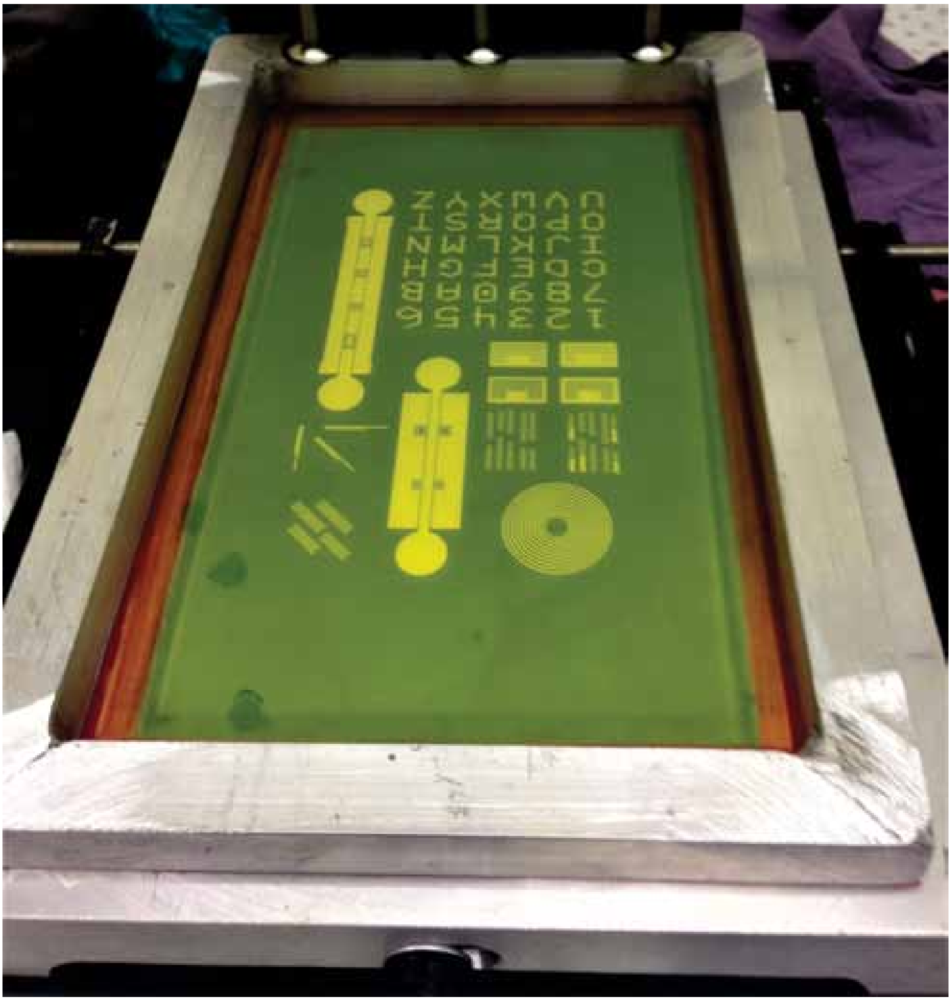


Figure 2.12 Screen printing film of the alphanumeric tags used in this book.

- *The quality of the ink as a conductance.* Ink quality is important as we could not see enough backscattering from some of the inks tested. InkTec has far less reflection than M-creative ink, although the conductivity after the oven was quite similar (Figure 2.13).
- *Quality of the ink.* Although the screen-printed films with different inks were all done at the same time, fading and oxidation of the InkTec ink after a few hours was observed.

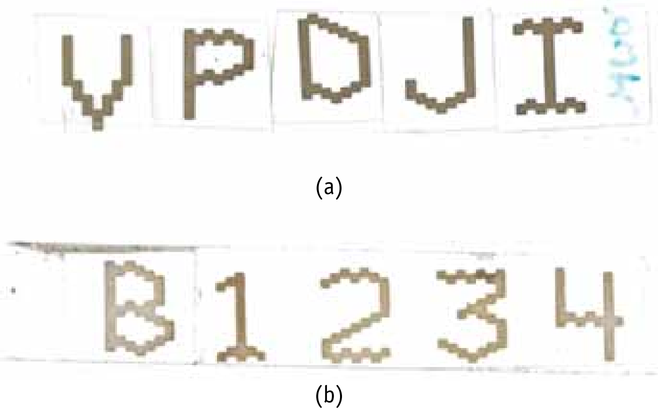


Figure 2.13 Letter tags screen-printed on Mylar polyester substrate using (a) M-creative ink, and (b) InkTec ink. Lower thickness (and hence less conductivity) and early metal oxidation are observed in (b) [61]. (Image courtesy of IEEE.)

- *Depth of the ink.* Although it is theoretically possible to use several additional layers of ink after the first layer dries out, in practice it ended up in disturbed inks over the cliffs and sharpness degradation at the corner of the letters. Even for one layer of printing, printing pressure and ink viscosity need to be carefully adjusted [57].
- *Viscous inks.* To avoid movement of the ink after being transferred to the substrate, normally inks should be rather viscous. The viscosity of the ink is the result of a large amount of additives to satisfy the rheological requirements for printing, but those additives degrade the electrical performance of the patterns [54].

Although screen printing on low-cost plastic substrates appears to be very attractive for RFID applications, special care should be taken in choosing the right conductive ink and sintering and curing process time and temperature. Correct handling and housekeeping of the inks is also a very important step in the final results. And note that screen printing is a very slow process. Even if the transparent films are ready, the whole production process takes a few hours in the lab.

The next section presents the design aspects of letter-based tags, which includes the effect of low-cost substrates on encoding capacity aspects. This is followed by Peyote tag design procedures.

2.3 Letter-Based Tag Design

The proposed chipless RFID system in this book is based on AI tag detection and decoding. To train a data-hungry network such as an AI system, many different tag samples are normally required. Using alphanumeric characters printed with conductive ink as tags can produce as many combinations as possible with very low cost. Current studies of letter-based tags are mainly focused on implementing single letters only, with data encoding 3.1 bits/cm^2 [31] and 2.1 bits/cm^2 [32]. Those tags are not expandable, are not orientation-independent, and cannot be printed on low-cost materials like Mylar polyester films [62]. Their operating frequency is normally in the S, C, and X bands of 3–10 GHz. In the next two sections two main considerations of chipless tags—substrate backscatter and encoding capacity for the tags in use—are analyzed in detail.

2.3.1 Effect of Substrate on Backscattered Signal

As nonmicrowave lossy substrates are used in this book, the efficiency of the materials are to be considered carefully. Figure 2.14 illustrates a low-cost letter-based tag screen-printed on a lossy substrate (Mylar laminate properties are considered similar to a thin polycarbonate sheet). For the tags to be detectable, there should be enough reflections available once an interrogation signal hits the tag's surface. In other words, for the best results in the frequency- or image-based domains, the tag's substrate has to have minimum absorption of the interrogation signal and the highest possible reflectivity index (backscattering from the tag).

The reflectivity of electromagnetic energy of a single-layer substrate as a function of frequency can be calculated using the following equation [63]:

$$R(dB) = \left| 20\log_{10} \frac{\mathrm{j}\,Z\tan(kd)-1}{\mathrm{j}\,Z\tan(kd)+1} \right| \tag{2.4}$$

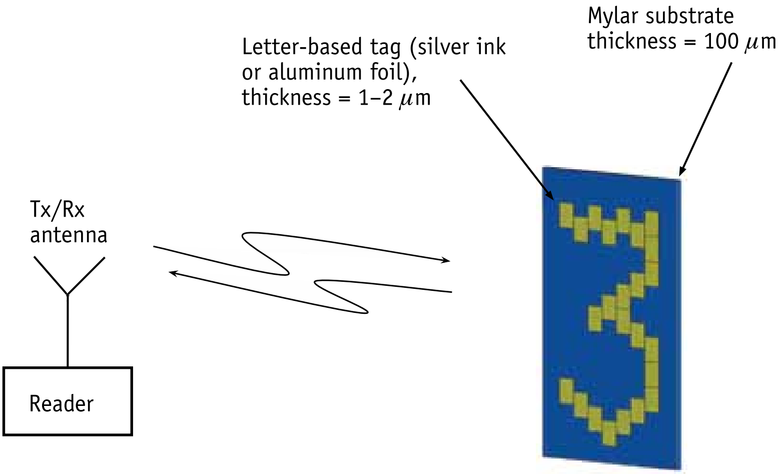


Figure 2.14 Isometric view of a letter-based tag printed on a lossy substrate (ε_r = 2.9, μ = 1, tand = 0.01).

where $Z = \sqrt{\mu/\varepsilon}$ is the intrinsic impedance of the substrate in ohms, and μ and ε are the complex permeability and permittivity of the material, respectively, $k = (2\pi f/c)$ is the wave number, f is the frequency of the incident electromagnetic wave, c is the speed of light in a vacuum, d is the thickness of the absorbing layer, and $j = \sqrt{-1}$. The permeability and permittivity might also be functions of the frequency [64].

A lower permittivity (ε_r in the range of 1 to 4) results in minimized cross talk and decreased propagation delays, whereas a medium or high permittivity (ε_r bigger than 8) results in smaller circuits but generates some phase changes. A low dissipation factor (DF) ($\tan\delta < 5 \times 10^{-4}$) results in the less heat generated and less signal attenuation [65]. The minimum absorption comes from the mismatch of the tag impedance to the incoming wave (antenna mode mismatch), as well as the substrate's intrinsic parameters.

Based on (2.4), Figure 2.15 shows a comparison between the reflectivity of a few materials used or considered [64–66]. Nonprofessional mm-wave substrates, like the ploy-carbonate and FR-4, have a lower reflectivity index compared to usual mm-wave substrates, meaning that they are far less suitable for mm-wave applications. This is the same for the paper and plastic.

Some techniques are therefore needed to increase the reflection, or increase the RCS, when using high abstract substrates like

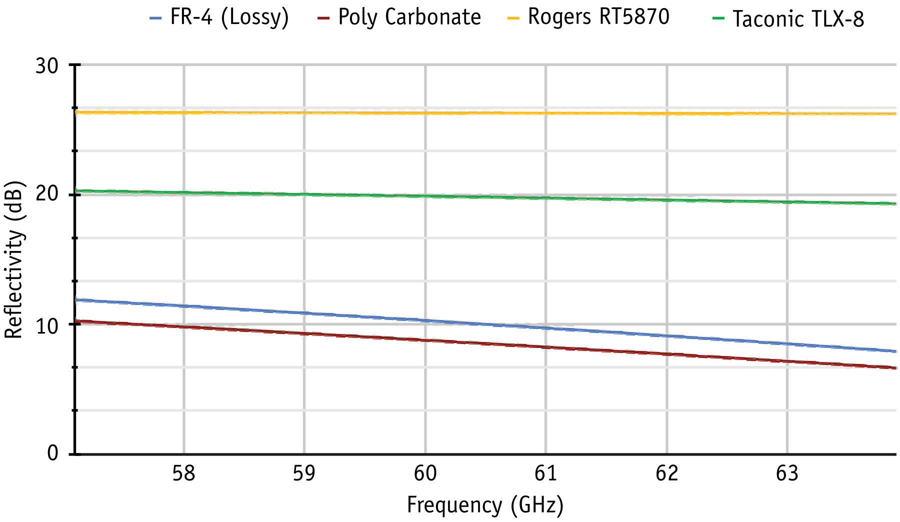


Figure 2.15 Reflectively comparison graph for the ISM band of 57–64 GHz. The higher the reflectivity, the lower the absorption of the substrate. Highly reflective substrates are more suitable for mm-wave chipless tags [61]. (Image courtesy of IEEE.)

paper or plastic (polycarbonate) at the mm-wave frequency band. Low absorption mode of the substrates is especially important for screen printing, in which metal is placed on the surface as a mesh, and it is not uniform. More reflection (and less absorption) of the substrate behind a chipless tag results in high RCS of the tag. This low-cost substrate problem is addressed in later sections of this chapter by using combinations of a few letters as one tag, therefore increasing the reflected signal.

2.3.2 Encoding Capacity Considerations

The next important design consideration for chipless tags is their encoding capacity. In the frequency domain and image-based tags the Q-factor will determine the number of bits in the frequency range. The higher the Q-factor of the responses, the more bits can be encoded in any particular frequency spectrum. Figure 2.16 shows the generic frequency response of a multiresonator passive tag. As can be seen, the 3-dB bandwidth is inversely proportional to the Q-factor of the tag.

As the detection method in this book is based on a tag's frequency differences, an important factor in the capacity concept is how distinguishable a tag response is from that of the other tags.

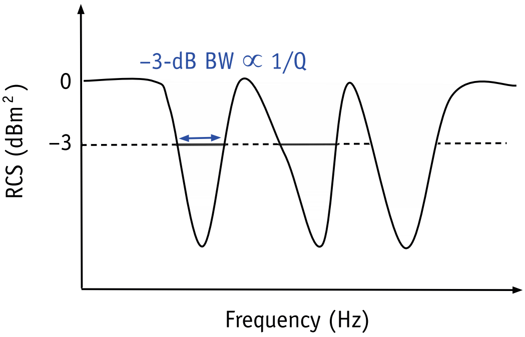


Figure 2.16 Generic frequency response of a multiresonator tag. If the nulls have high-Q, more bits can be encoded.

Hence, the focus will be on designing tags that have more changes in their frequency responses and thus a higher chance of being uniquely identified.

2.3.3 Tag Design Based on the Peyote Alphabet

Letter-based chipless tags printed on low-cost laminates are the main focus in this book. Making combinations of letters will lead to different tag responses, which is needed to train a data-hungry network of AI-based processing. There are many ways to make letter-based tags, but one of the most structured is Peyote beading alphabets [67, 68] because all the alphabets are made using small, adjustable-in-size building blocks. Figure 2.17 shows the design concept of the Peyote alphabet with a few alphanumeric characters.

The letters are made of small building blocks. Hereafter we use the terms *building block* and *cliff* as equivalent words. The tags are made in a mesh size of 11-by-7 cliffs. In this design the cliffs are chosen in the way that the tag fits in the 1-by-1 cm^2 space. The distance between the middle of every two tags is also considered to be 1 cm. Although the number of cliffs per tag is 11-by-7 as in Figure 2.17, they are considered 22-by-7 in practice as some cliffs need to be placed in the middle of the other cliffs (in the vertical dimension), and the space between them is 0.5 cliff-size wide (refer to the highlighted cliff in letter "B," where the cliff is halfway up from its neighbors).

These alphanumerical tags have the following characteristics:

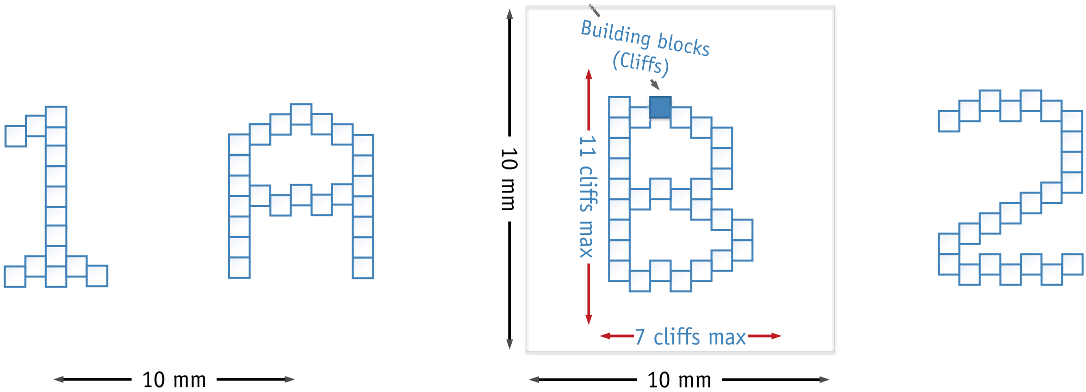


Figure 2.17 Peyote alphanumeric design method showing the building blocks (cliffs). Each character will occupy a 1-by-1 cm^2 space, so a four-character tag like this will be 4-by-1 cm^2 [61]. (Image courtesy of IEEE.)

- They are based on 30-by-30 mil^2 (0.762 cm^2) cliffs. Each character occupies 1 cm^2, considering the space between characters.
- They have increased data capacity in comparison to their microwave substrates counterparts. This is because in each symbol position of 1 cm^2 space, at least 52 different alphabetical symbols (26 uppercase and 26 lowercase letters) + 20 numbers/symbols can be placed. This results in a 72-symbols/cm^2 capacity. Considering $64 = 2^6$ symbols means 6 bits, the encoding capacity will be slightly higher than 6 bits/cm^2.
- They have components in the x- and y-planes. As will be reported in the reader design presented in Chapter 3, the reader in this work is based on a cross-polarized antenna configuration. Having components in the x- and y-planes makes the tags less sensitive to orientation issues compared to the meander-line and bar tags. The tag reflects sufficient cross-polarization backscattering signal to guarantee detection in any orientation.
- They are less sensitive to printing imperfections because of the AI detection method.
- In particular, any substrate can be used for the tags provided that the backscatter signal is large enough and the link budget condition is preserved.

Other factors that affect the overall designed tag performance are the resolution of the tag scanning, number of scans in different tag/reader angles and distances, and type of detection method [53]. These factors are covered in the next chapters, which go into detail of the hardware development and detection methods.

Figure 2.18 is a photograph of the tags on different substrates with different printing techniques. Out of these tags, the SATO printer results were unsatisfactory because of their lower and undetectable RCS in the measurements. The low RCS of the SATO printer happened because of poor conductivity of the ribbon used in the printing process. The tags shown are measured using a one- or two-port vector network analyzer (VNA) with horn antennas connected to the ports, and the results are discussed in the following sections.

2.3.4 Peyote Tag Frequency Response

Figure 2.19 illustrates the response of two 5-letter tags in the experiment. The VNA amplitude response of the tags "MGAQ5" (TagID = 03) and "RLFPV" (TagID = 06) were done in the monostatic case reader with a tag-to-reader distance of 5 cm.

There are a few characteristics for this design response illustration. First, the tag response is well above the noise level as the RCS is high. This is due to using a few letters at the same time to increase the reflection, and, hence, the RCS. Second, there is a 10-dB difference between the top peak and bottom null, which makes the tag response clear. Third, there are enough peaks and nulls and

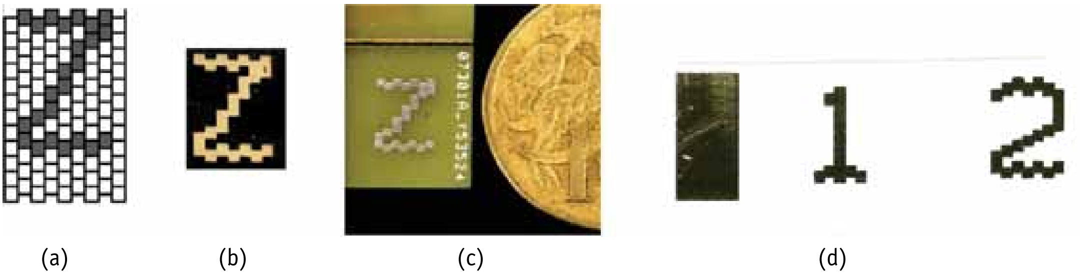


Figure 2.18 Making peyote symbols on different substrates: (a) initial Peyote alphanumeric plan [67], (b) ink printing using InkTec on Mylar plastic substrate, (c) on FR4 substrate and size compared to AUD $1 coin, and (d) letters printed using SATO printer (200 × 200 pixels/inch resolution) [53]. (Image courtesy of IEEE.)

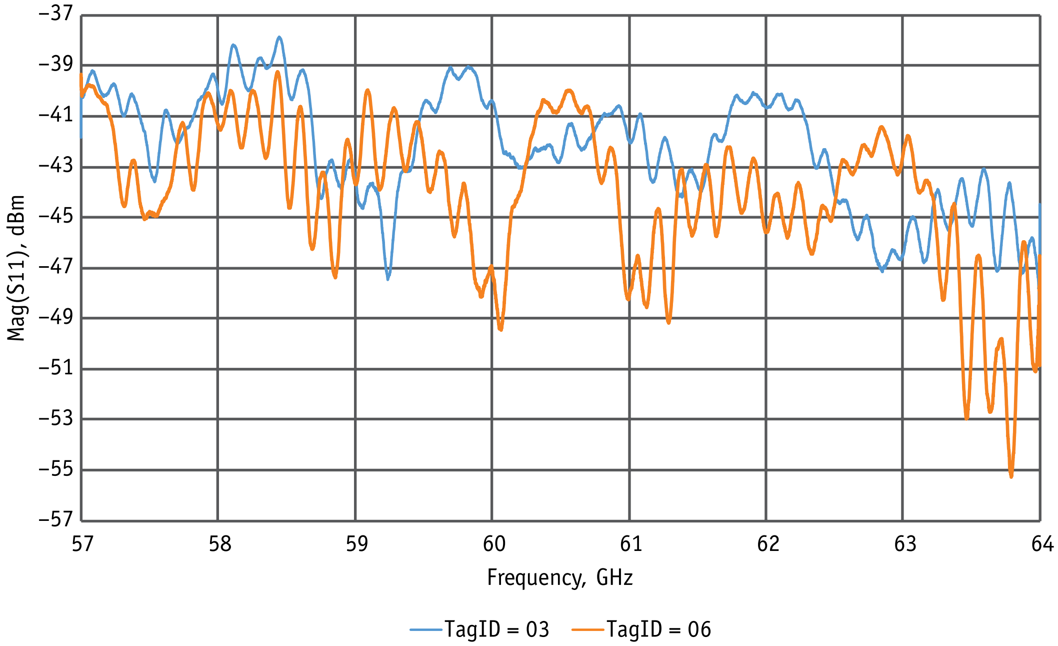


Figure 2.19 RCS comparison of two tags printed on low-cost Mylar substrate after background noise deduction, in the spectrum of 57 to 64 GHz. There is at least 10-dB difference between peak and nulls, and there are less flat areas in the response curve. Nonflat areas make it easier for the pattern recognizers to correctly identify the tag.

variations in the response to make the tag response unique and distinguishable compared to the other tags' frequency responses. The place of the nulls and peaks can be adjusted based on the chosen letters and their size, as will be shown in Section 2.5.1.

The next section presents the RCS analysis and measurement of backscattering responses of the tag using both monostatic and bistatic readers.

2.4 Backscattering Theory and RCS Calculations

In this section the theory and a practical path toward chipless tag's RCS measurement in monostatic (transmit antenna and receive antenna are the same or colocated) and bistatic (transmit and receive antenna are not colocated) reader cases are provided. Backscattering theory is the basis for the chipless tag design, as the tags need to be backscatterer and data encoder at the same time. The tag can be interrogated either in a bistatic or monostatic configuration.

In the development of the RCS theory, the main focus will be on the bistatic interrogation mode first, as shown by Figure 2.20, and thereafter the monostatic case will be explained.

In the mathematical mode analysis, once a CW signal of the form

$$x_i(t) = K_i \cdot \cos(\omega_i t) \tag{2.5}$$

from a reader interrogates the tag, where *i* index is the index of the sinusoidal sent at a particular time, the tag response, or the signal received by the receiving antenna in the time domain, can be expressed as

$$y(t) = y_s(t) + y_a(t) \tag{2.6}$$

where $y_s(t)$ corresponds to the tag's structural mode RCS response and $y_a(t)$ is its antenna mode response. In (2.6), for the sake of simplicity, index *i* has been omitted at the output compared to the interrogation signal in (2.5). The structural mode RCS is the part of the scattered signal from an incoming electromagnetic wave because of the scattering from tag's supporting structure, such as its ground plane. Some part of interrogation signal energy will be absorbed because of the antenna effect. Antenna mode scattering comes from the impedance mismatching [2]. In the chipped RFID, the mismatch is intentionally created with a switching electronics to vary the impedance of the tag, which in turn changes the tag's antenna mode RCS. Such RCS modulation is called antenna load modulation. Usually in the "on" state of the switch, antenna impedance is matched and provides the least backscattering. This state is equal to the binary 0. On the other hand, when the switch is off, the impedance mismatch reflects the maximum backscattering RCS to the reader. This state is referred to as the binary 1. A microcontroller that is programmed for coded tagID controls the switch and thus creates a stream of binary data.

The chipless tag, however, is not designed to have matching impedance to its structure, and the tag's data is stored in this $y_a(t)$ signal. Normally the antenna mode is much smaller than the structural mode, and in the time domain responses, the antenna mode

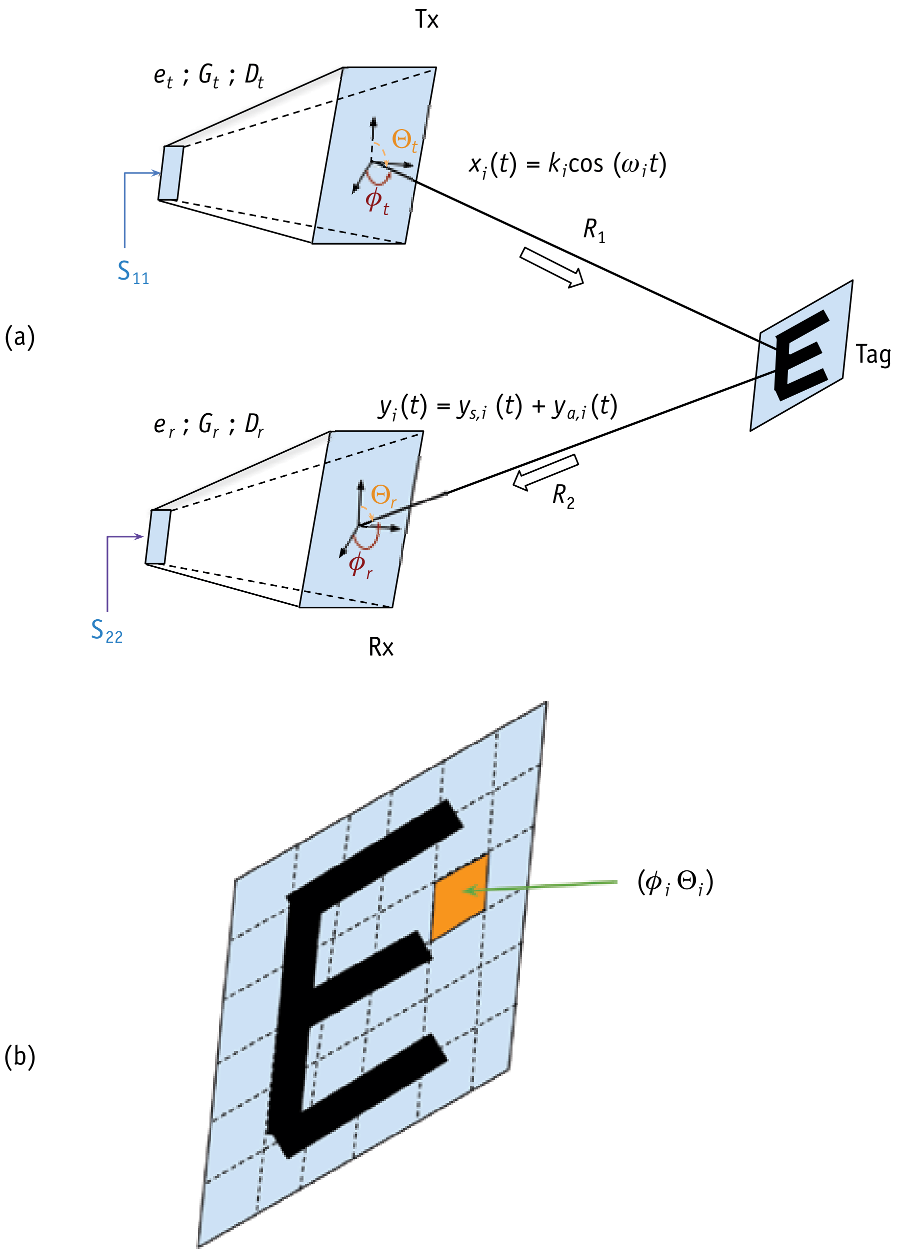


Figure 2.20 Backscattering theory explained on a chipless tag: (a) transmitter and receiver configuration, and (b) a closer view of a chipless tag.

is the delayed response compared to the structural mode response in the time sequence [69].

Hereafter, capital letters will be used to denote the signals in the frequency domain and lowercase letters will be used for the signals in the time domain. Figure 2.20(a) shows the bistatic reader. The reader response equation in the frequency domain is expressed as [70]

$$\frac{P_r}{P_t} = e_t\left(1-|S_{11}|^2\right)\frac{D_t\left(\theta_t,\phi_t\right)}{4\pi R_1^2}\times\sigma\frac{\lambda^2}{4\pi}\times\left(e_r\left(1-|S_{22}|^2\frac{D_r\left(\theta_r,\phi_r\right)}{4\pi R_2^2}\right)\right) \tag{2.7}$$

where P_r and P_t are the received and transmitted power, respectively, S_{11} is the reflection coefficient of the transmitting antenna, S_{22} is the reflection coefficient of the receiving antenna, e_t and e_r are the radiated efficiencies associated with the transmitter and the receiver, respectively, $D_t(\theta_t, \phi_t)$ is the directivity of the transmit antenna, $D_r(\theta_r, \phi_r)$ is the directivity of the receiving antenna, R_1 is the distance from the transmit antenna to the tag, R_2 is the distance from the tag to the receiving antenna, λ is the wavelength, and σ is the RCS of the tag. The Tx signals hit every point in the tag and the Rx antennas receive the backscatter signal from each point. Figure 2.20(b) shows a letter-based tag scanned at a highlighted angle of (ϕ_i, θ_i) that corresponds to $(\phi_{t,i}, \theta_{t,i})$, and $(\phi_{r,i}, \theta_{r,i})$. For simplicity in expression, the "i" subscript might be dropped.

Equation (2.7) can be written with a simpler radar response equation of [2]:

$$\frac{P_r}{P_t} = \frac{\sigma\lambda^2}{4\pi}\frac{G_t}{4\pi R_1^2}\frac{G_r}{4\pi R_2^2} \tag{2.8}$$

as the transmitter's gain toward the tag, G_t

$$G_t\left(\theta_t,\phi_t\right) = e_t D_t\left(\theta_t,\phi_t\right) \tag{2.9}$$

as the receiver antenna gain toward tag, G_r

$$G_t\left(\theta_r,\phi_r\right) = e_r D_r\left(\theta_r,\phi_r\right) \tag{2.10}$$

as the reflections coefficients of the transmitter and receiver antennas (S_{11} and S_{22}, respectively) are relatively low (normally below

–13 dB), the terms (1–|S_{11}|^2) and (1–|S_{22}|^2) are sometimes ignored (we still consider those terms hereafter).

For this book, the alphanumerical tags are normally printed on low-cost plastic or paper, such as Figure 2.18, and as there is no ground plane, it can be assumed that tag's structural mode backscattering signal is not as significant as the antenna mode signal. A tag's response can be found by removing the background noise from S_{11} and S_{22} parameters. This means

$$S_{11} = S_{11}^{withTag} - S_{11}^{withoutTag} \tag{2.11}$$

and

$$S_{22} = S_{22}^{withTag} - S_{22}^{withoutTag} \tag{2.12}$$

We know

$$\frac{P_r}{P_t} = \frac{\left|y_s(t) + y_a(t)\right|^2 / Z_0}{\left|x(t)\right|^2 / Z_0} = \frac{\left|y_s(t) + y_a(t)\right|^2}{\left|x(t)\right|^2} = \frac{\left|Y_s(f) + Y_a(f)\right|^2}{\left|X(f)\right|^2} \tag{2.13}$$

where Z_0 is the characteristic impedance of the system. Substituting (2.13), (2.11), and (2.12) into (2.7), one gets

$$\begin{aligned}\frac{\left|y_a(t)\right|^2}{\left|x(t)\right|^2} = \frac{\left|Y_a(f)\right|^2}{\left|X(f)\right|^2} \approx \frac{P_r}{P_t} = K\left(\theta_t, \phi_t, \theta_r, \phi_r\right) \frac{\sigma\lambda^2}{\left(4\pi R_1 R_2\right)^2} \times \\ \left(1 - \left|S_{11}^{withTag} - S_{11}^{withoutTag}\right|^2\right)\left(1 - \left|S_{22}^{withTag} - S_{22}^{withoutTag}\right|^2\right)\end{aligned} \tag{2.14}$$

in which

$$\begin{aligned}K\left(\theta_t, \phi_t, \theta_r, \phi_r\right) &= \frac{1}{4\pi} e_t D_t\left(\theta_t, \phi_t\right) \times e_r D_r\left(\theta_r, \phi_r\right) \\ &= \frac{1}{4\pi} G_t\left(\theta_t, \phi_t\right) G_r\left(\theta_r, \phi_r\right)\end{aligned} \tag{2.15}$$

Equation (2.14) provides the magnitude of the tag response in each point of the tag, and it depends on transmitter and receiver antenna gains in the tag place. The final response of the tag will be a *superposition* of the signals received from all parts of the tag. In the CW sweep or frequency modulated continuous-wave (FMCW) mode, based on the resolution, there are a many $x_i(t)$ signals, (2.5), hitting the different parts of the tag's surface, and many $y_i(t)$ backscatter signals from diffident angles (θ_t,ϕ_t) and (θ_r,ϕ_r). The superposition of these received signals create the final tag response by the receiver, as illustrated in Figure 2.20. In the simulations, these time- or frequency-domain backscatter simulations require very high computational power if the size of the simulation workspace is large. For a computation mesh of 25×10^6 cells, for example, it took a few hours for this equation to be calculated in a finite element simulation software such as CST or HFSS.

As said, the tag information is actually stored in $y_a(t)$. In the time domain, the antenna mode response term $y_a(t)$ in (2.13) is weaker and is delayed behind the structure mode of the tag, so putting a time domain window in theory can result in recovering the $y_a(t)$ [71].

In the monostatic case, $R_1 = R_2 = R$, $\phi_t = \phi_r = \phi$, $\theta_t = \theta_r = \theta$ and $S_{11} = S_{22}$, $G_t(\theta_t,\phi_t) = G_r(\theta_r,\phi_r) = G(\theta,\phi)$ and (2.14) can be simplified to:

$$\frac{|y_a(t)|^2}{|x(t)|^2} = \frac{|Y_a(f)|^2}{|X(f)|^2} \approx \frac{P_r}{P_t} = K(\theta,\phi)\frac{\sigma\lambda^2}{(4\pi R^2)^2} \times \left(1 - \left|S_{11}^{withTag} - S_{11}^{withoutTag}\right|^2\right)^2 \tag{2.16}$$

in which

$$K(\theta,\phi) = \frac{1}{4\pi}G^2(\theta,\phi) \tag{2.17}$$

These equations can be used to calculate the RCS (σ) or an estimation of the received signal for the tag, $y_a(t)$ if the σ is known. This is useful for calibration of the developed reader in Chapter 3.

In the next section the designed tag response characteristics are discussed, and a technique is developed to enhance the backscatter patterns.

2.5 Tag Performance Simulations

The basis of simulations for the remainder of this chapter will be discussed in this section. A couple of simulations are carried out for various purposes through this study. These simulations, done with a time domain solver in CST, are as follows:

- *RCS simulations.* The main purpose of these simulations is to ensure that there is sufficient backscatter signal from the tag and how the size of the tag should be changed to match the desired response. This will be covered in detail in Section 2.5.1.
- *Rotation sensitivity analysis simulations.* As already mentioned, one of the most important aspects of every chipless RFID system is to make it less sensitive to the reading angle. Since alphanumeric tags are not inherently symmetrical, a few simulations are performed to show how the detection AI system can correct the misaligned tags in different reading angles. The results are also presented in Chapter 4.
- *iSAR simulations.* In these simulations, the tag (target) is moved in linear steps in a rail perpendicular to the Tx/Rx antennas (reader). This was used to create the image of the tag and create enough different angles for training the AI network. The results of these simulations will be discussed in detail in Chapter 4.

In the RCS simulation aspect, the main purpose is to design tags that satisfy the following backscratching conditions:

- The designed tag should be able to reflect enough signals in a detectable level. In these designs, the amount of the ink on the substrate, the conductivity of the ink, the size of each letter, and the distance are the influencing factors for better RCS. Having five letters for every tag improves detection rate drastically by increasing the RCS.
- Besides sufficient reflection, which makes the received signal well above the noise level and satisfies the link budget, the difference between peaks and nulls is very important in this design. This is because tags with flatter responses have

much less data encoding capability than tags with differentiable peaks and nulls once decoded by pattern recognizers.

- Letters can be designed in a way that they have dominant peaks/nulls at particular frequencies, which makes the frequency response nonflat.

The basic simulation setup is shown in Figure 2.21. The far-field distance for a horn antenna with an aperture size of 12 mm at 60 GHz is

$$\text{Far-field distance} = 2\frac{D^2}{\lambda} = 2\frac{0.012^2}{0.005} = 5.7\ \text{cm} \tag{2.18}$$

In this formula, D is the antenna aperture used in the experiments and CST simulations. A number of simulations for the performance are conducted and reported in this section. As a general procedure, tag-to-reader distance is chosen to be 5 cm minimum in simulations and 5 to 10 cm in different experiments in this and the following chapters. This distance is a near-far-end, or a far-end as per the distance calculation of (2.18) for the horn antenna used.

It must be mentioned that most of the experimental results in this book come from a cross-polarized system such as the one in Figure 2.21. When both the Tx and Rx use one polarization (or share the same antenna), it is called copolarized. The copolar situation in this book mostly happens when VNA is in use for a tag's

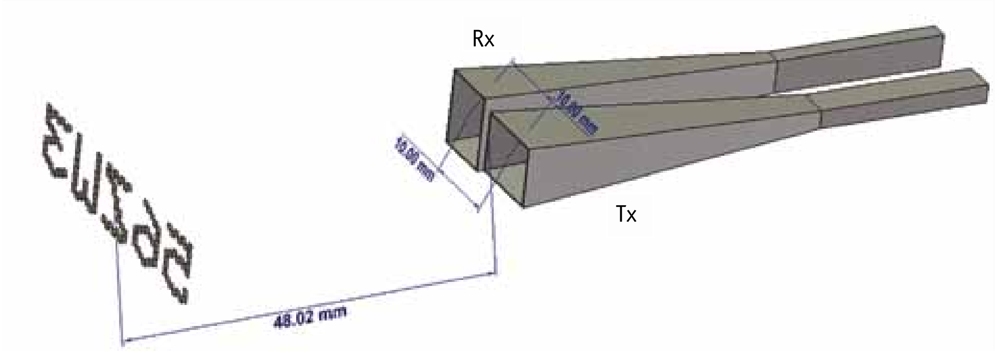


Figure 2.21 CST simulation showing the reader dimensions and distance of the reader and the tag.

reflection measurements, whereas cross-polar experiments are both done with VNA and the developed reader. As will be discussed in Chapter 4, the formulas for making a SLAR image of tags are similar for both co- and cross-polarized measurements. For long-distance radar imaging, such as polarimetric-SAR, the resultant image from these two types of antenna configurations are different for smooth surfaces (like lakes or deserts), and are similar for rough surfaces (like rocks, forests, and human-made objects) as most of them have corner reflector type features [72, 73].

In the following section, the first simulation is used to get the best-performing tags, in the sense of RCS and frequency tunability.

2.5.1 Tag Design Improvement

In this section, methods of improving tag response are described. As mentioned in Section 2.3.3, using a five-letter tag increases the RCS, but at the same time the frequency response of the individual letters may adversely affect the desired overall response of the tag. Figure 2.22 compares the overall response of the "56ZW3" tag using the black dashed line as the magnitude of S21, along individual responses of every letter.

In observing the overall tag response in Figure 2.22, two important notes must be pointed out. First, overall tag response should not be flat. A flat response means less data can be encoded or extracted, and distinguishing the tag with pattern recognizers will be more difficult. AI decoding methods (tag pattern recognition method and deep learning, which are explained in Chapters 4 and 5) can distinguish the final combination of tags; however, the more distinguishable the response of every tag is, the easier it is to recognize the total pattern. The higher peaks or deeper nulls in the individual tags are used as a signature verification means to increase the detection rate and to lower possible errors. Second, from the overall response curve it is not really possible to decompose the individual letters, and being able to extract or decompose each letter provides a higher level of decoding rates as well.

The first step in optimizing the frequency response is enhancing the letters so that their influence on the overall five-letter tag can be improved. To satisfy the letter tag design of Section 2.5, a few simulations were carried out. Here each letter is individually characterized first, and then its overall frequency response is

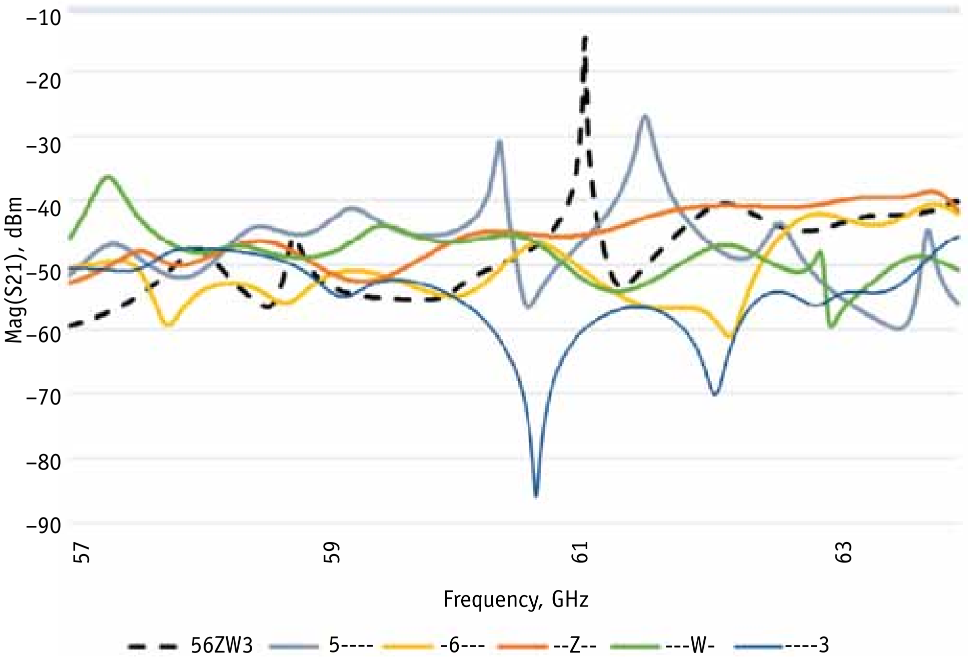


Figure 2.22 Simulation results for the frequency response of the individual letters and their overall responses for the "56ZW3" tag for the cliff size of 0.8 mm for all letters in the tag.

improved by optimizing the letter size by changing its cliff sizes. For now, our focus is how to increase the peak/null depths by optimizing the size of the building blocks in every letter.

Figure 2.23 shows the effect of changing building blocks in the RCS for letter "W." A sweep of 20 steps has been done from 0.1 to 2 mm in the cliff's size, and the effect of cliff size versus location of nulls in the frequency response is shown. From Figure 2.23, for example, it can be seen that a peak at frequency 57.504 GHz with a value of –18.9 dBm occurs for letter "W" if the size of its building blocks is chosen to be 1.2 mm. A similar simulation is carried on for character "6," as shown in Figure 2.24.

2.5.2 Discussion of the Results

Based on the findings of Figures 2.23 and 2.24 and similar graphs for the other letters, a configuration is found to optimize the effect of every character in the final five-tag combination. In the larger scale, once this experiment is done on all of the letters, a simple

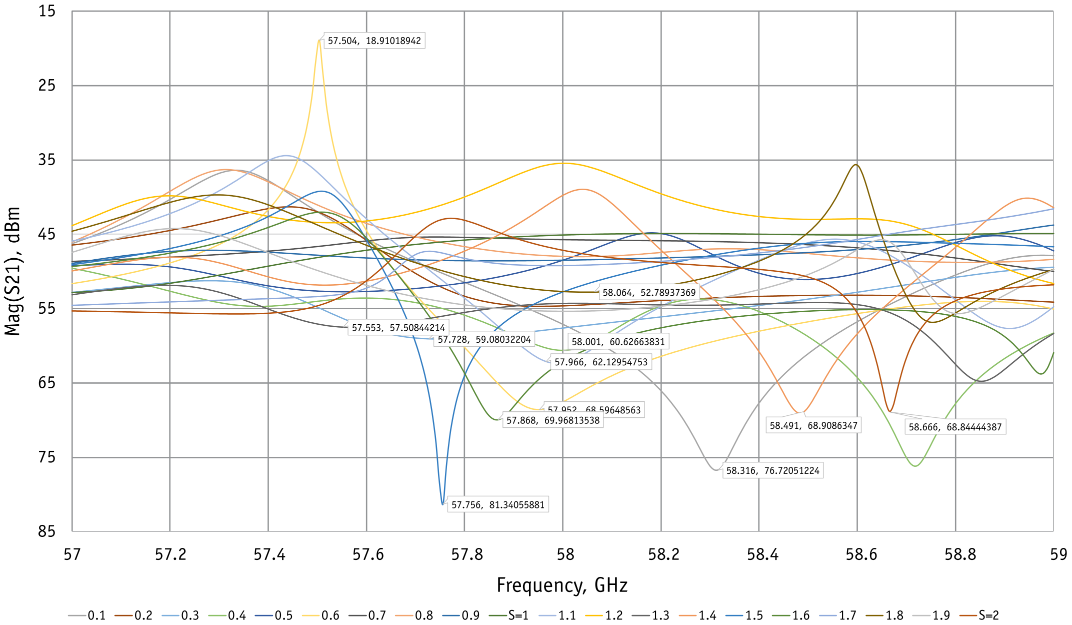


Figure 2.23 Frequency responses of the letter “W” tag by changing its building block size from 0.1 to 2 mm [53]. (Image courtesy of IEEE.)

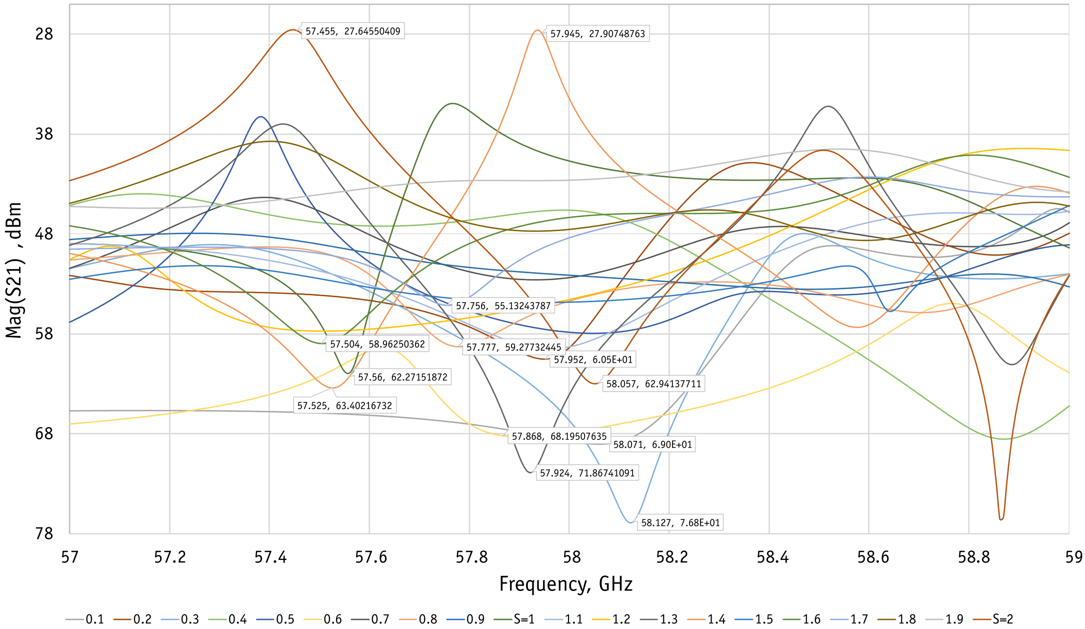


Figure 2.24 Frequency responses of the character “6” tag by changing its building block size from 0.1 to 2 mm.

program is used to determine the cliff's size for each letter in the letter combinations for maximum efficiency of null and peak placements. Figure 2.25 summarizes the cliff size versus frequency for making a peak or null for three letters, "Z," "6," and "W." To get reasonable results, a separation of few hundred megahertz should be considered between the null of one letter and peak of the adjacent letter in the five-letter tag. For example, to make a null at the frequency around 57.3 GHz for letter "Z," a cliff of size 0.5 mm will be used for letter "Z," and for a 57.6-GHz peak for the letter "W," cliff size should be 0.26 mm for the letter "W," and these two peaks and nulls have less interference with each other because there is a 30-MHz frequency separation between them.

Simulation using three tag combinations of "6ZW" with different cliff sizes for each character is done to verify the concept. Figure 2.26 shows the frequency responses with the initial tag size and updated tag (the tag with different-sized cliffs for every letter). The overall frequency response is improved in peaks and nulls (red lines), both in the phase and magnitude responses. This makes the tag more distinguishable once it is presented to an AI decoding system, which is sensitive to different distinguishable signatures.

Unlike other types of frequency tags (discussed in Section 2.2.2), the frequency response of the cliff-based alphanumeric tags

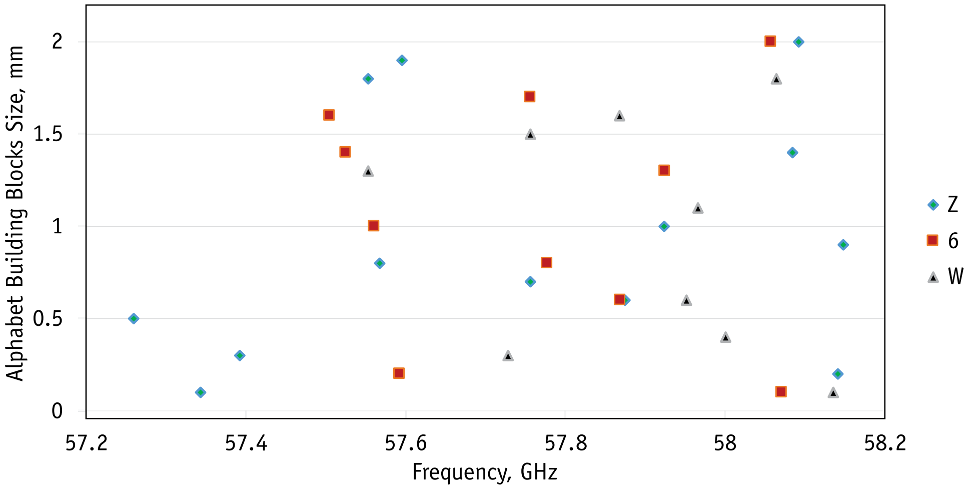
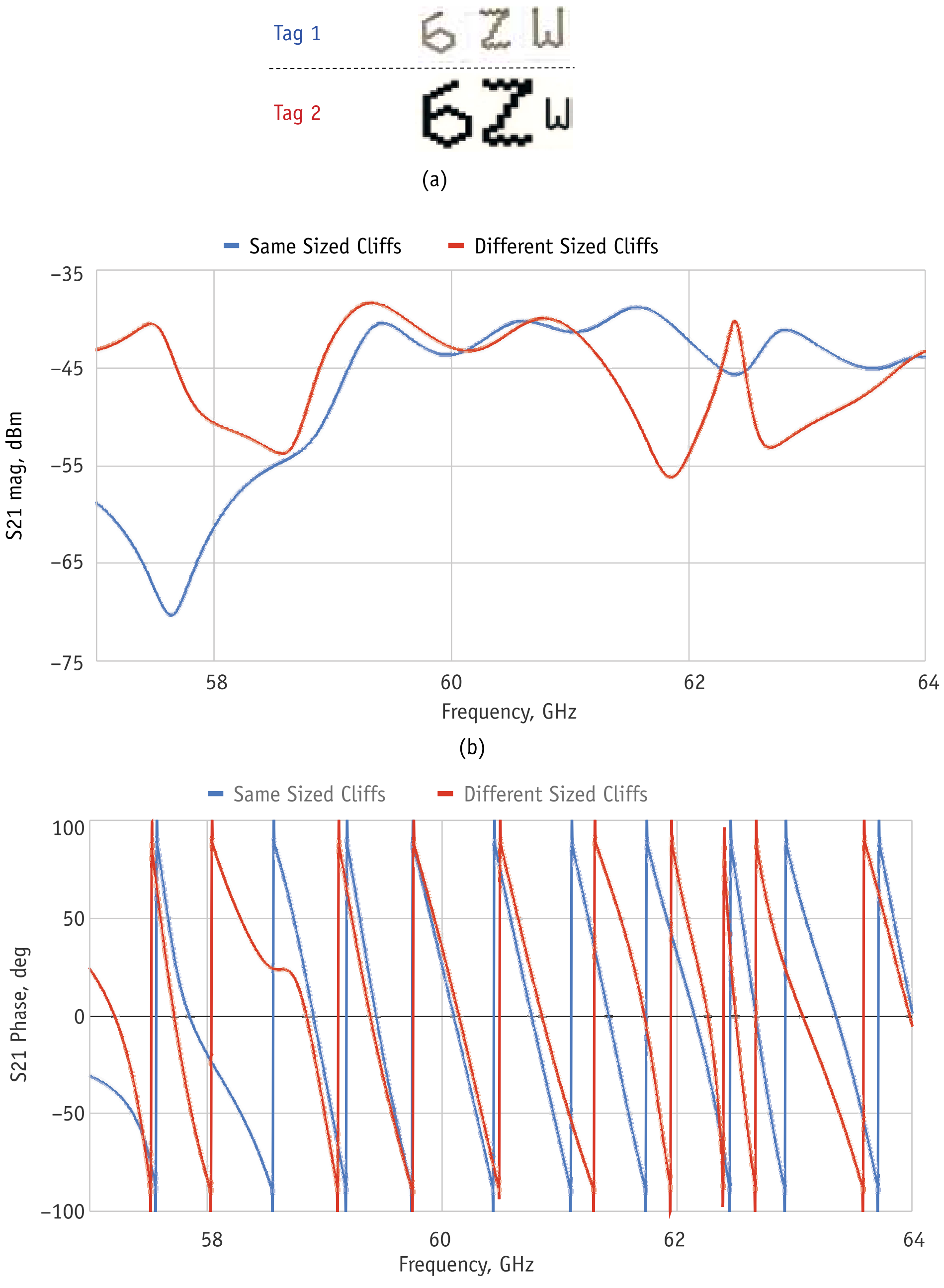


Figure 2.25 Changing building block size for optimized nulls in the tag frequency response of a few characters in the tag [53].



(c)

Figure 2.26 (a) "6ZW" tag with the same building block size for all letters and with different building block (cliff) sizes for each letter, (b) magnitude comparing the tags in (a), and (c) phase comparison. More changes in gain and phase are observed after building block adjustments per letter from Tag1 to Tag2. The blue lines in these graphs belong to Tag1 and the red lines belong to Tag2 [53].

is not easy to interpret. Each tag is made of a few characters, and each character is made of many cliffs. The combination response of the cliffs is not easy to be interpreted in the mm-wave spectrum, as discussed in the scattering theory in Section 2.4. Some of this interpretation difficulty comes from non-mm-wave substrates (plastic or paper) used in this book, as their material absorption and reflectivity are not necessarily uniform/linear within the mm-wave spectrum. Another notable observation in the simulations is shown in Figure 2.27. This figure shows the current density around two of the tags in the simulations, alphanumerics 3 and 6. In letters that have a loop (like 6, 0, P, A), a circular current is observed around joint cliffs, which might be corresponding to the deeper nulls in the frequency response.

The next section explains the experimental setup and practical design considerations.

2.6 Tag Response Measurements

The experimental setup is demonstrated in Figure 2.28. The experiments are not done in fully interference-free environments such as an anechoic chamber, but several microwave absorbers are set

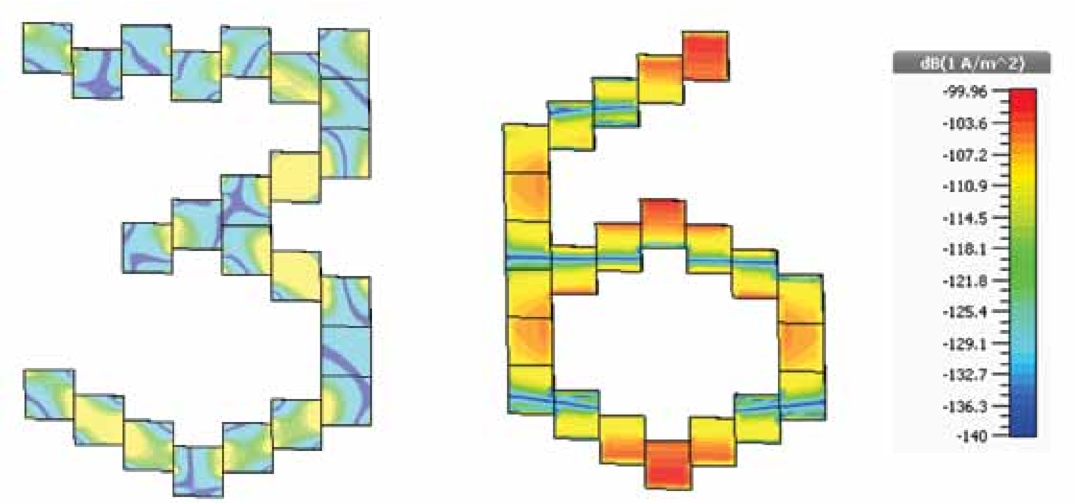


Figure 2.27 The current density around tags. For letters with a closed loop, such as "0" and "6," a higher loop current is observed, which can lead to the deeper (higher-Q factor) nulls in the frequency response [53]. (Image courtesy of IEEE.)

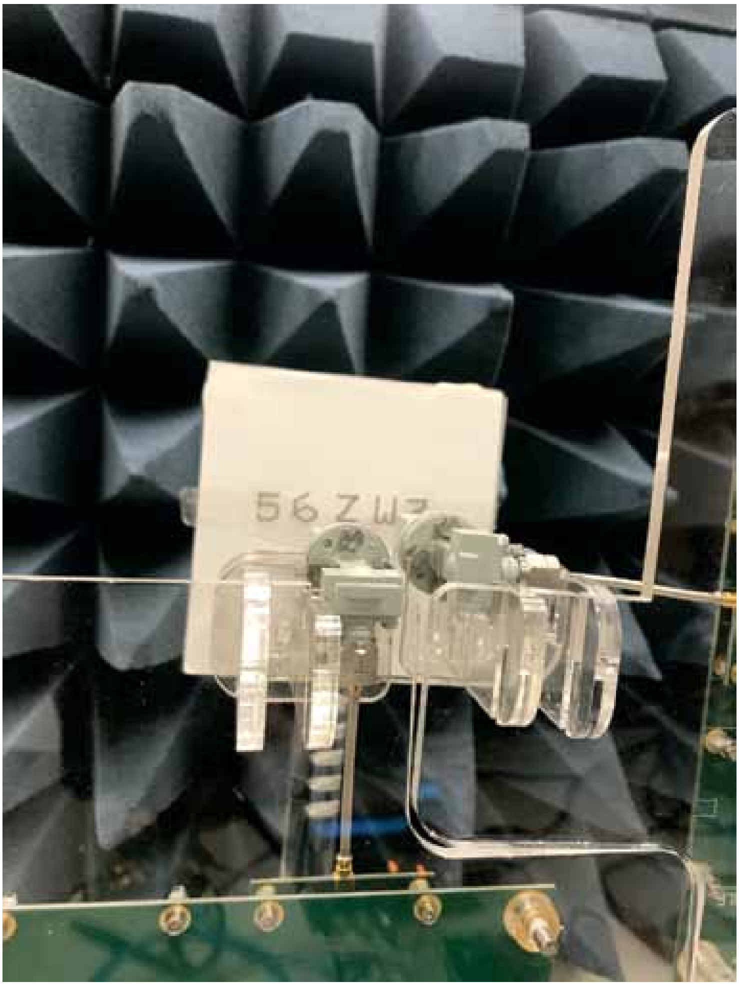


Figure 2.28 Experimental measurement setup using cross-polarized antennas for the Tx and Rx. Microwave absorbers are used to reduce undesired environment reflections [53]. (Image courtesy of IEEE.)

around to reduce unwanted environmental reflections. The responses of the five-letter tag “56ZW3” in the simulations and experiment are shown in Figure 2.29. In the simulations the block size was 33 mil, and in the screen printing it was 30 mil on a 100-μm Mylar polyester film. The size of the total tag is 5 by 1 cm^2, which is roughly the size of an optical barcode. The tag’s building blocks are quite visible in the experiment setup in Figure 2.28.

In the experiment, the background backscatter is measured first with the presence of a jig, and then the measurements are

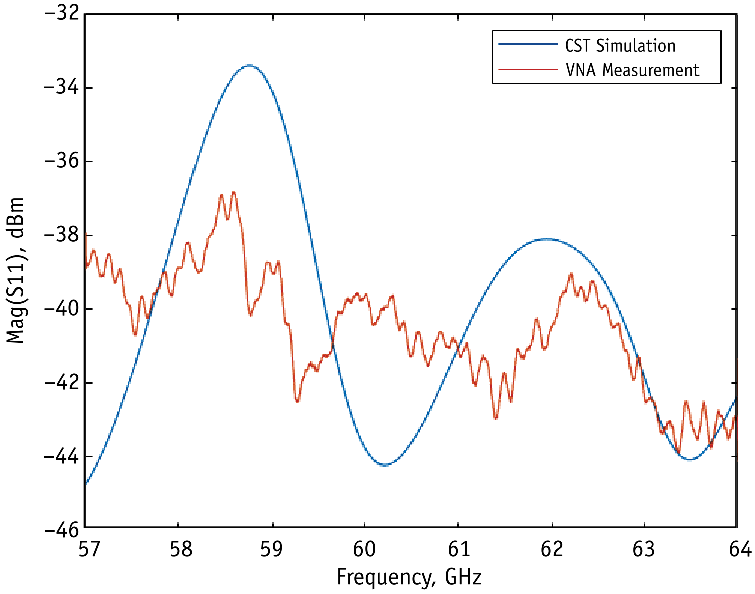


Figure 2.29 CST and VNA responses of the tag "56ZW3" in the cross-polar configuration. The dashed line is the smooth average of the VNA output. The mismatch is normally less than 2 dB [61]. (Image courtesy of IEEE.)

subtracted from those with the tag placed on the jig (similar to the S_{11} deduction formula of (2.16)). The results of the simulations and experiment of a one-port system are shown in Figure 2.29. There is a 2- to 4-dB difference in the magnitude of the simulated and experimented results, and there is one phase difference around 60 GHz in that figure. This phase shift might be the result of perfect substrate consideration in the CST simulations, and not pure removal of background noise in the lab practices. Also note that the S11 signal is around –40 dBm, or –70 dB. As

$$S_{11}(dB) = P_{reflected\_signal}(dB) - P_{transmitted\_signal}(dB) \tag{2.19}$$

And we set the transmitter power to

$$P_{trasmitted\_signal}(dB) = 5\ dBm = -25\ dB \tag{2.20}$$

Then

$$P_{reflected\_signal}(dB) = S_{11}(dB) + P_{transmitted\_signal} = -70 - 25 = -95\ dB \tag{2.21}$$

So, the reflected signal = $10^{-9.5}$ W or $50 \times 10^{-9.5} \approx 0.12$ mV (as the impedance is 50Ω). The main goal here is to get a similar notion of the RCS level so that each tag can be assigned with a unique code.

2.7 Conclusions

Based on the research presented in this chapter, it appears that letter-based chipless tags screen-printed on the plastic substrate are of great interest, as they are cheap and flexible enough to be used as hidden barcodes. The problems are that they do not have enough backscatter reflections to be detected effectively, they are not symmetric in their design to be orientation-independent, and their frequency response is difficult to adjust for peak and null locations.

In this chapter these main problems were illustrated and possible solutions were proposed. Combining a few letters as one tag was utilized to multiply the encoding capacity and boost the backscatter signal in order to achieve sufficient backscattering. Several inks were tested for conductivity and backscatter response, and the best ones were chosen.

Because the system is based on AI, as will be covered in Chapter 4, the rotated tags could still be detectable with higher than 70% accuracy. This is inherent to the nature of intelligence added to the detection algorithm, similar to a human brain that can still read letters correctly if they are rotated. This provides some flexibility for tag orientation errors.

Although alphanumerical tags were perfect for applications that require many different chipless tag sets, their frequency response was not easily predictable if a combination of several characters were used together. To address the adjustment of the frequency response, a few simulations were conducted to tune the size of the cliffs. It was shown that with adequately sized cliffs in each letter, specific frequency peaks and nulls were achieved. Although each individual letter's frequency responses interfered with one another, if the frequency difference between nulls and peaks were high enough, the overall response of the tag could be adjusted. Although the proposed letter tags were frequency-based, the only requirement was to make their frequency responses distinguishable from each other by using different letters or adjusting their frequencies by changing their cliff sizes. Enormous flexibility in the tag design was shown to be possible compared to conventional frequency signature-based tags where a high-Q is a must for decoding.

2.8 Tag Design Questions and Answers

2.1 What is the purpose of alphanumerical tags?

Any machine learning needs lots of samples, and by using letters and alphabets many different combinations can be made. Having a few letters as one tag also increases the RCS, increasing the chances of the tag being seen by the reader. As well, reading alphanumerical tags is easy for humans, and these tags can be considered for imaging purposes too, unlike other types of frequency-based tags (such as resonator tags).

2.2 How do you calculate the encoding capacity of alphanumerical tags?

Imagine an empty position, X, which can be filled with any lowercase or uppercase letter, number, or special character. There are 26 uppercase characters (A to Z), 26 lowercase characters (a to z), 10 numbers (0 to 9), and many special characters (#, %, etc.). If we consider this to be 64 possibilities, then as $64 = 2^6$, we can encode 6 bits of data per each empty position. If the space of a letter is 1 cm^2, the encoding capacity will be 6-bits/cm^2. For a tag consisting of a combination of five letters, encoding will be five times more, or 30 bits.

2.3 Why are five letters considered a tag, and not three or six?

Each letter is made in a 1×1 cm^2 space, and five of them together will be 5×1 cm^2, almost the size of an optical barcode. Having more letters in a tag also means higher RCS (detectivity) and higher encoding capacity per tag.

2.4 Is a ground plane needed in a tag?

In the experiments, it is shown that adding a ground plane (the tag's back-plate is fully metal) increases the Q-factor of the nulls, and the tag will have a stronger backscattering signal toward the reader. A ground plane in the tag also helps with the depolarization effect, which is sensitivity of the reading tag in different polarizations [74]. Having a ground plane therefore makes it easier to separate the chipless tags from unwanted signals backscattered by the reflective background objects. But having a ground plane will increase the tag's sensitivity to the reading angle, and in a 180° rotation it might blind the reader. It is also not a feasible option in tags on plastic substrates, such as tags produced with inkjet or screen-printing techniques.

2.5 Why was a screen-printing technique used?

We tried printed tags using a SATO printer, but the RCS was very low. Screen printing can have much higher conductivity on coated aluminum paper compared to using a SATO printer. Additionally, screen printing is a simple task that can be accomplished in the lab using small laboratory furnaces.

References

[1] Arjomandi, L. M., *A Google Colab (Jupiter Notebook) Python Code for the Angular Sensitivity Plot*, March 2021, doi:10.26180/19387412.

[2] Karmakar, N. C., "Tag, You're It Radar Cross Section of Chipless RFID Tags," *IEEE Microwave Magazine,* Vol. 17, No. 7, 2016, pp. 64–74, doi:10.1109/MMM.2016.2549160.

[3] Herrojo, C., M. Moras, F. Paredes, et al., "Very Low-Cost 80-Bit Chipless-RFID Tags Inkjet Printed on Ordinary Paper," *Technologies*, Vol. 6, No. 2, 2018, p. 52.

[4] Babaeian, F., and N. C. Karmakar, "Hybrid Chipless RFID Tags–A Pathway to EPC Global Standard," *IEEE Access*, Vol. 6, 2018, pp. 67415–67426.

[5] Preradovic, S., and N. C. Karmakar, "Multiresonator Based Chipless RFID Tag and Dedicated RFID Reader," in *2010 IEEE MTT-S International Microwave Symposium*, IEEE, 2010, pp. 1520–1523.

[6] Karmakar, N. C., R. Koswatta, P. Kalansuriya, and R. E-Azim, *Chipless RFID Reader Architecture,* Norwood, MA: Artech House, 2013.

[7] Vena, A., E. Perret, and S. Tedjini, "Chipless RFID Tag Using Hybrid Coding Technique," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 59, No. 12, 2011, p. 3356.

[8] Ni, Y.- Z., X.- D. Huang, Y.- P. Lv, and C.- H. Cheng, "Hybrid Coding Chipless Tag Based on Impedance Loading," *IET Microwaves, Antennas & Propagation*, Vol. 11, No. 10, 2017, pp. 1325–1331.

[9] Rezaiesarlak, R., and M. Manteghi, *Chipless RFID: Design Procedure and Detection Techniques*, Springer International Publishing, 2015, pp. 1–159, doi:10.1007/978-3-319-10169-9.

[10] Zomorrodi, M., "mm-Wave EM-Imaging Chipless RFID System," PhD dissertation, Monash University, 2015.

[11] "Diamond-Based Surface Acoustic Wave Devices: A Reverse Fabrication Design," http://www.av.it.pt/jisis/saw.html.

[12] Plessky, V. P., and L. M. Reindl, "Review on SAW RFID Tags," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, Vol. 57, No. 3, 2010, pp. 654–668.

[13] Forouzandeh, M., and N. C. Karmakar, "Chipless RFID Tags and Sensors: A Review on Time-Domain Techniques," *Wireless Power Transfer*, Vol. 2, No. Special Issue 02, 2015, pp. 62–77, DOI: 10.1017/wpt.2015.10.

[14] McVay, J., A. Hoorfar, and N. Engheta, "Space-Filling Curve RFID Tags," in *2006 IEEE Radio and Wireless Symposium*, 2006, pp. 199–202, doi:10.1109/RWS.20061615129.

[15] Preradovic, S., I. Balbin, N. C. Karmakar, and G. Swiegers, "Chipless Frequency Signature Based RFID Transponders," in *38th European Microwave Conference (EuMC 2008),* 2008, pp. 1723–1726, DOI: 10.1109/EUMC.2008.4751808.

[16] Forouzandeh, M., *Design and Implementation of Ultra-Wideband Frequency-Domain Chipless RFID Readers,* PhD dissertation, Monash University, March 2020, DOI: 10.26180/5e7c08eea81ea.

[17] Derneryd, A., "Analysis of the Microstrip Disk Antenna Element," *IEEE Transactions on Antennas and Propagation*, Vol. 27, No. 5, 1979, pp. 660–664.

[18] Freyens, B. P., M. Loney, and M. Poole, "Wireless Regulations and Dynamic Spectrum Access in Australia," in *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, IEEE, 2010, pp. 1–12.

[19] Rezaiesarlak, R., and M. Manteghi, "A Space-Frequency Technique for Chipless RFID Tag Localization," *IEEE Transactions on Antennas and Propagation*, Vol. 62, No. 11, 2014, pp. 5790–5797.

[20] Khadka, G., *Development of a Chipless RFID System for Internet of Things (IoT) Applications,* PhD dissertation, Monash University, October 2020, DOI: 10.26180/13031204.v1.

[21] Li, J., and P. Stoica, *MIMO Radar Signal Processing,* Hoboken, NJ: Wiley-IEEE Press, 2009.

[22] Khawar, A., A. Abdelhadi, and T. C. Clancy, *MIMO Radar Waveform Design for Spectrum Sharing with Cellular Systems, A MATLAB Based Approach*, Springer International Publishing, 2016, doi:10.1007/978-3-319-29725-5.

[23] Krieger, G., I. Hajnsek, K. Papathanassiou, et al., "The Tandem-L Mission Proposal: Monitoring Earth's Dynamics with High Resolution SAR Interferometry," in *2009 IEEE Radar Conference,* IEEE, 2009, pp. 1–6.

[24] Grishin, A. M., and R. M. Mays, *Microwave Readable Dielectric Barcode*, U.S. Patent 7,205,774, 2007.

[25] Karmakar, N. C., M. Zomorrodi, and C. Divarathne, *Advanced Chipless RFID: MIMO-Based Imaging at 60 GHz - ML Detection*, Hoboken, NJ: John Wiley & Sons, 2016.

[26] InkSure Technologies, *Progress Report on Chipless RFID Applications SARcode Development by InkSure Technologies,* https://www.slideserve.com/addisonglenn/progress-report-on-chipless-rfid-applications-sarcodedevelopment-by-inksure-technologies.

[27] Zomorrodi, M., and N. C. Karmakar, "Optimized MIMO-SAR Technique for Fast EM-Imaging of Chipless RFID System," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 65, No. 2, 2016 pp. 661–669.

[28] Wikipedia, "MIMO Radar," https://en.wikipedia.org/wiki/ MIMO_radar.

[29] EeSeanLiu, W. C., *Mimo Radar,* June 2022, https://en.wikipedia.org/wiki/MIMO_radar#/media/File:Virtual_array.png.

[30] Singh, T., S. Tedjini, E. Perret, and A. Vena, "A Frequency Signature Based Method for the RF Identification of Letters," in *2011 IEEE International Conference on RFID*, 2011, pp. 1–5.

[31] Tedjini, S., O. Boularess, T. Andriamiharivolamena, H. Rmili, and T. Aguili, "A Novel Design of Chipless RFID Tags Based on Alphabets," in *Microwave Symposium (IMS), 2017 IEEE MTT-S International*, IEEE, 2017, pp. 1561–1563.

[32] Boularess, O., H. Rmili, T. Aguili, and S. Tedjini, "Analysis of Electromagnetic Signature of Arabic Alphabet as RF elementary Coding Particles," *Wireless Power Transfer*, Vol. 2, No. 2, 2015, pp. 97–106.

[33] Islam, M. A., Compact Printable Chipless RFID Systems, PhD dissertation, 2014.

[34] Hartmann, C. S., "A Global SAW ID Tag with Large Data Capacity," in *2002 IEEE Ultrasonics Symposium, 2002. Proceedings.*, IEEE, Vol. 1, 2002, pp. 65–69.

[35] Shao, B., Q. Chen, Y. Amin, S. M. David, R. Liu, and L.-R. Zheng, "An Ultra-Low-Cost RFID tag with 1.67 Gbps Data Rate by Ink-Jet Printing on Paper Substrate," in *2010 IEEE Asian Solid-State Circuits Conference*, IEEE, 2010, pp. 1–4.

[36] Jalaly, I., and I. Robertson, "RF Barcodes Using Multiple Frequency Bands," *IEEE MTT-S Digest*, Vol. 4, 2005.

[37] Blischak, A., and M. Manteghi, "Pole Residue Techniques for Chipless RFID Detection," in *2009 IEEE Antennas and Propagation Society International Symposium*, IEEE, 2009, pp. 1–4.

[38] Lim, N., J. Kim, S. Lee, N. Kim, and G. Cho, "Screen Printed Resonant Tags for Electronic Article Surveillance Tags," *IEEE Transactions on Advanced Packaging*, Vol. 32, No. 1, 2009, pp. 72–76.

[39] Jang, H.- S., W.- G. Lim, K.- S. Oh, S.- M. Moon, and J.- W. Yu, "Design of Low-Cost Chipless System Using Printable Chipless Tag with Electromagnetic Code," *IEEE Microwave and Wireless Components Letters*, Vol. 20, No. 11, 2010, pp. 640–642.

[40] Vena, A., E. Perret, and S. Tedjini, "Design of Compact and Auto-Compensated Single-Layer Chipless RFID Tag," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 60, No. 9, 2012, pp. 2913–2924.

[41] Preradovic, S., and N. C. Karmakar, "Design of Fully Printable Planar Chipless RFID Transponder with 35-Bit Data Capacity," in *2009 European Microwave Conference (EuMC)*, IEEE, 2009, pp. 013–016.

[42] Nijas, C., R. Dinesh, U. Deepak, et al., "Chipless RFID Tag Using Multiple Microstrip Open Stub Resonators," *IEEE Transactions on Antennas and Propagation*, Vol. 60, No. 9, 2012, pp. 4429–4432.

[43] Balbin, I., and N. Karmakar, "Novel Chipless RFID Tag for Conveyor Belt Tracking Using Multiresonant Dipole Antenna," in *2009 European Microwave Conference (EuMC)*, IEEE, 2009, pp. 1109–1112.

[44] Balbin, I., and N. Karmakar, "Radio Frequency Transponder System," *Australian Provisional Patent DCC, Ref*, Vol. 30684143, 2008.

[45] Bhuiyan, M. S., and N. Karmakar, "Chipless RFID Tag Based on Split-Wheel Resonators," in *2013 7th European Conference on Antennas and Propagation (EuCAP)*, IEEE, 2013, pp. 3054–3057.

[46] Vena, A., E. Perret, and S. Tedjini, "A Fully Printable Chipless RFID Tag with Detuning Correction Technique," *IEEE Microwave and Wireless Components Letters*, Vol. 22, No. 4, 2012, pp. 209–211.

[47] Hamdi, M., F. Garet, L. Duvillaret, P. Martinez, and G. Eymin-Petot-Tourtollet, "New Approach for Chipless and low Cost Identification Tag in the THz Frequency Domain," in *2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, IEEE, 2012, pp. 24–28.

[48] El-Awamry, A., M. Khaliel, A. Fawky, M. El-Hadidy, and T. Kaiser, "Novel Notch Modulation Algorithm for Enhancing the Chipless RFID Tags Coding Capacity," in *2015 IEEE International Conference on RFID (RFID)*, IEEE, 2015, pp. 25–31.

[49] Balbin, I., and N. C. Karmakar, "Phase-Encoded Chipless RFID Transponder for Large-Scale Low-Cost Applications," *IEEE Microwave and Wireless Components Letters*, Vol. 19, No. 8, 2009, pp. 509–511.

[50] Mukherjee, S., and G. Chakraborty, "Chipless RFID Using Stacked Multilayer Patches," in *2009 Applied Electromagnetics Conference (AEMC)*, IEEE, 2009, pp. 1–4.

[51] Vena, A., E. Perret, and S. Tedjini, "A Compact Chipless RFID Tag Using Polarization Diversity for Encoding and Sensing," in *2012 IEEE International Conference on RFID (RFID)*, IEEE, 2012, pp. 191–197.

[52] Islam, M. A., and N. C. Karmakar, "A Novel Compact Printable Dual-Polarized Chipless RFID System," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 60, No. 7, 2012, pp. 2142–2151.

[53] Arjomandi, L. M., G. Khadka, Z. Xiong, and N. C. Karmakar, "Document Verification: A Cloud-Based Computing Pattern Recognition Approach to Chipless RFID," *IEEE Access*, Vol. 6, 2018, pp. 78007–78015.

[54] Zheng, L.- R., H. Tenhunen, and Z. Zou, *Smart Electronic Systems: Heterogeneous Integration of Silicon and Printed Electronics*, John Wiley & Sons, 2018.

[55] Vena, A., E. Perret, S. Tedjini, et al., "Design of Chipless RFID Tags Printed on Paper by Flexography," *IEEE Transactions on Antennas and Propagation*, Vol. 61, No. 12, 2013, pp. 5868–5877.

[56] Ali, Z., E. Perret, N. Barbot, et al., "Authentication Using Metallic Inkjet-Printed Chipless RFID Tags," *IEEE Transactions on Antennas and Propagation*, Vol. 68, No. 5, 2020, pp. 4137–4142, doi:10.1109/TAP.2019.2948740.

[57] Shrestha, S., "Practical Implementation of Chipless RFID System," PhD dissertation, Monash University, March 2019, doi:10.26180/5c8ecf1a3ea58.

[58] Custom Planet Co., *What Is Screen Printing? A Step-by-Step Guide,* https://www.customplanet.co.uk/what-is-screen-printing-stepby-step-i50.

[59] InkTec, *Inktec Introduction,* http://www.inktec.com/down/pdf/(E)ElectronicInk_v3.0.pdf.

[60] I. Creative Materials, *118-09A/B119-44 Solvent-Resistant Electrically Conductive Ink,* https://server.creativematerials.com/datasheets/DS_118_09. pdf.

[61] Arjomandi, L. M., G. Khadka, and N. C. Karmakar, "mm-Wave Letter-Based Chipless RFID Tags on Cheap Plastic Substrates," in *2020 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP)*, IEEE, 2020, pp. 1–4.

[62] Arjomandi, L. M., and N. C. Karmakar, "An Enhanced Chipless RFID System in 60 GHz Using Pattern Recognition Techniques," in *2018 48th European Microwave Conference (EuMC)*, 2018, pp. 973–976, doi:10.23919/EuMC.2018.8541656.

[63] Balanis, C. A., *Advanced Engineering Electromagnetics*, John Wiley & Sons, 1999.

[64] Folgueras, L. C., M. A. Alves, and M. C. Rezende, "Dielectric Properties of Microwave Absorbing Sheets Produced with Silicone and Polyaniline," *Materials Research*, Vol. 13, No. 2, 2010, pp. 197–201.

[65] Baker-Javis, J., Janezic, M. D., B. Riddle, C, L. Holloway, N. G. Paulter, and J. E. Blendell, "Dielectric and Conductor-Loss Characterization and Measurements on electronic Packaging Materials," *NIST Technical Note 1520,* 2001.

[66] Rogers Corp, *Rogers Product Selector Guide and Standard Thicknesses and Tolerances,* https://www.rogerscorp.com/documents/2532/acs/ACS-ProductSelector-Guide-and-Standard-Thicknesses-and-Tolerances.pdf.

[67] Kerrigan, S., *Peyote Alphabet Pattern*, 2015, http://www.bellaonline.com/articles/art42985.asp.

[68] ALYA, *Peyote Beading Alphabet Patterns,* http://www.domcon.co/peyote-beading-alphabet-patterns/.

[69] Kalansuriya, P., N. C. Karmakar, and E. Viterbo, "On the Detection of Frequency-Spectra-Based Chipless RFID Using UWB Impulsed Interrogation," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 60, No. 12, 2012, pp. 4187–4197, doi:10.1109/TMTT.2012.2222920.

[70] Balanis, C. A., *Antenna Theory: Analysis and Design*, Third Edition, Hoboken, NJ: John Wiley, 2005.

[71] Islam, M. A., and N. C. Karmakar, "Compact Printable Chipless RFID Systems," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 63, No. 11, 2015, pp. 3785–3793.

[72] Oka, S., H. Togo, N. Kukutsu, and T. Nagatsuma, "Latest Trends in Millimeter-Wave Imaging Technology," *Progress in Electromagnetics Research Letters,* Vol. 1, 2008, pp. 197–204.

[73] Sheen, D. M., D. L. McMakin, and T. E. Hall, "Near Field Imaging at Microwave and Millimeter Wave Frequencies," in *IEEE/MTT-S International Microwave Symposium, 2007,* IEEE, 2007, pp. 1693–1696.

[74] Ali, Z., O. Rance, N. Barbot, and E. Perret, "Depolarizing Chipless RFID Tag Made Orientation Insensitive by Using Ground Plane Interaction," *IEEE Transactions on Antennas and Propagation*, 2022.

3

Chipless Reader Design

3.1 Introduction

In the preceding chapter the details of chipless RFID tag design, encoding, and decoding methods and practical implementation challenges were presented. Because a chipless tag is just a piece of metal, it is the reader's responsibility to send the interrogation signal, capture the backscattered signal, and process it further to decode the information. The reader hardware is the next stage in the development of a chipless RFID tag detection and decoding system. Chipless reader development is based on the type of chipless tag and detection method.

The method of detection can be categorized into those for time domain tags and frequency-domain tags [1–3]. Image-based tags can be considered as a special category of frequency tags in which scanning over the surface is done in a few positions or by using multiple transmitters/receivers.

Time-domain readers normally produce a sharp pulse/burst as the interrogation signal using a pulse generator [4]. The resultant signal is an UWB one since sharp pulses require very high bandwidth. As a general rule of thumb, calculation for the required bandwidth (BW) of a 1-ns pulse with rise time of 0.1 ns, a bandwidth of

$$Required\ bandwidth = 0.35/(0.1ns) = 3.5\,GHz \tag{3.1}$$

is needed [5]. The echoes from tags are delayed in a matter of nanoseconds. After detection of the backscattered signal, the reader starts signal processing based on the delay times to decode the bits in the tag. The interrogation and reception process is quite fast thus requiring a very fast ADC as well; however, the pulse timing process and the rest of the decoding hardware is not very complicated.

In frequency-based readers, a voltage-controlled oscillator (VCO) produces a CW or linear frequency modulation continuous wave (LFMCW) signal in the whole available spectrum, so the scanning time takes a bit longer compared to the time domain readers. The scanning time is a function of the frequency resolution in the UWB interrogation signal bandwidth. For a CW signal, the ADC does not need to be fast, while for a FMCW signal, the only ADC speed requirement is to effectively capture the scattered signal. However, the complexity arises in the interpretation of the detected signal, which requires high to very high levels of signal processing.

In image-based tag readers, a burst of pulses, CWs, or LFMCWs is sent to the tag. Either several antennas are used to record the scattered signal (MIMO), or the tag or reader are moved or rotated using the iSAR detection technique. These signals are processed based on the time and location of the tag/readers to provide a 2-D or three-dimensional (3-D) representation of the tag.

This chapter presents the design of a frequency and image-based reader based on a modular design approach. Since a 60-GHz system on a chip was not available in the market at the time of this writing, it was a safe option to create the reader with a modular design, including separate transmitter, receiver, and control electronics. This is the first implemented mm-wave CW chipless RFID reader to the best knowledge of the authors. The main goal is to develop a fully automated 60-GHz chipless RFID using a modular design approach. This reader is a part of the first practical system, and uses a cross-polarized antenna for Tx and Rx to better suppress in-band interference.

Developing a 60-GHz chipless RFID reader is a nontrivial task. It needs sound planning, rigorous link budget calculations, appropriate selection of the transceiver modules at 60 GHz and then integrating them into the system, development of in-house custom-built modules, mechanical considerations for linear rail and its control circuit, development of machine learning algorithms for tag detection and decoding, and, finally, development of an appropriate graphical user interface (GUI) for the end user. Each of these tasks in turn need rigorous planning, logistics, conducting of experiments, developing system blocks and peripherals, and, finally, integration into a functional system. Figure 3.1 shows the flow chart of the complete hardware design procedure for the proposed 60-GHz RFID system.

The chapter is organized as follows: After this introduction, the overall blocks of the system are described in Section 3.3. Choosing the right RF module is critical, and that will be discussed in Section 3.4. Next, Section 3.5 details the design and integration of RF, LO, gain/phase cooperator, digital controller, and peripheral circuits such as display and linear rail stepper motor controller. Specifications of the overall hardware, such as scanning time and resolution and calibration procedure, are described in Section 3.6, followed by the conclusions.

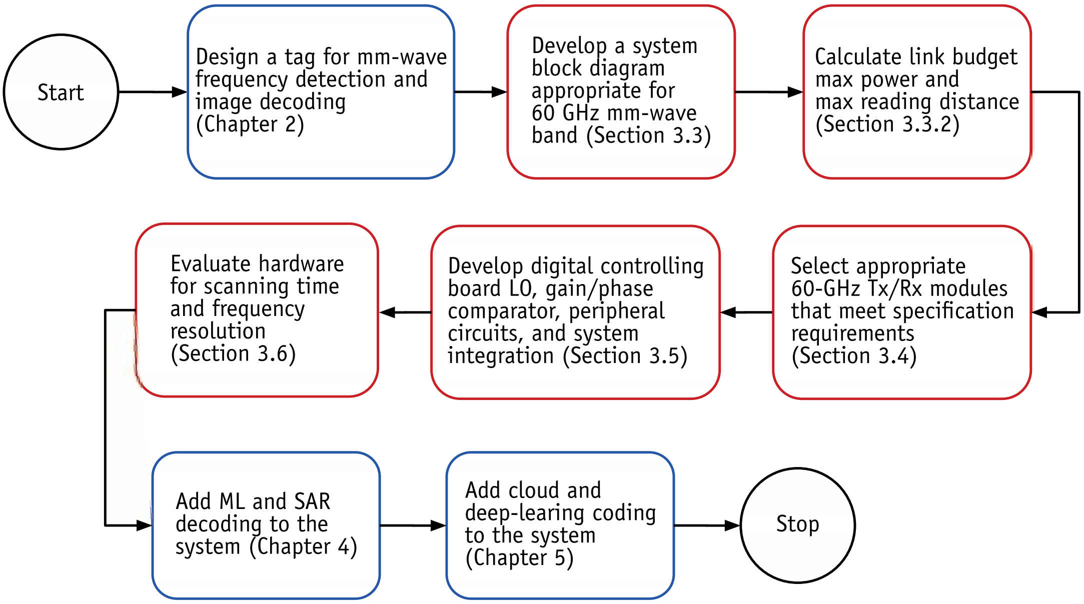

Start
Design a tag for mm-wave frequency detection and image decoding (Chapter 2)
Develop a system block diagram appropriate for 60 GHz mm-wave band (Section 3.3)
Calculate link budget max power and max reading distance (Section 3.3.2)
Select appropriate 60-GHz Tx/Rx modules that meet specification requirements (Section 3.4)
Develop digital controlling board LO, gain/phase comparator, peripheral circuits, and system integration (Section 3.5)
Evaluate hardware for scanning time and frequency resolution (Section 3.6)
Add ML and SAR decoding to the system (Chapter 4)
Add cloud and deep-learing coding to the system (Chapter 5)
Stop

Figure 3.1 Different sections of development for the mm-wave reader system. The red squares indicate topics covered in this chapter.

3.2 Chipless RFID Readers

This book is based on frequency and image-based systems, so this section mainly focuses on these readers.

3.2.1 Frequency-Based Readers

Frequency-based tags are essentially multiresonators. A typical frequency-based reader block diagram is shown in Figure 3.2 [1, 6].

In Figure 3.2, the microcontroller uses a digital-to-analog converter (DAC) to generate a ramp signal with a VCO. This signal goes through a power amplifier and a bandpass filter (BPF) and propagates toward the tag. The received signal is amplified and then compared with the pilot signal, and a gain/phase detector (AD8302) provides the change in the gain and phase.

The background interference and self-interference are the main problems with frequency-based chipless RFID readers. Removing the background interference—the noise that comes from the reflection of other objects near the tag—normally requires a calibration process, in which background data (no tag) is saved and then subtracted from the tag's measurements [7, 8]. As the background may change from time to time, a periodic recalibration is recommended.

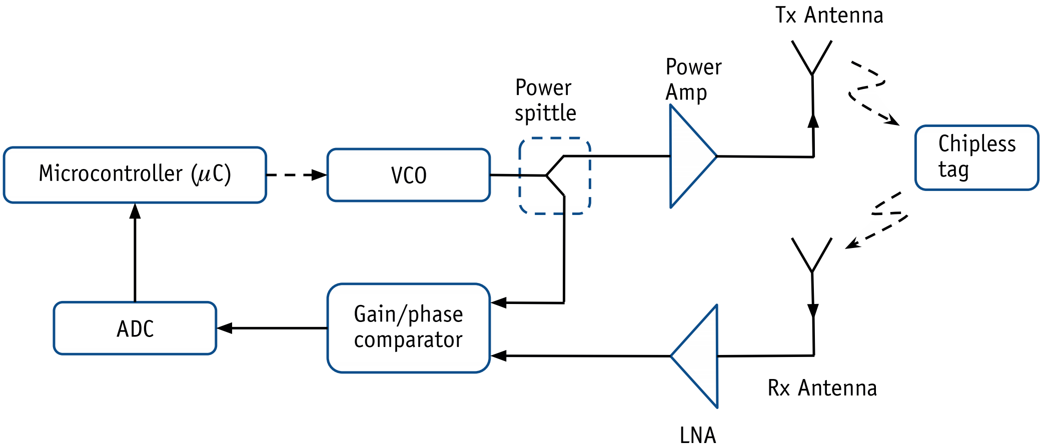


Figure 3.2 A typical frequency-based chipless RFID reader for magnitude and phase detection. A microcontroller (μC) sends a ramp signal to the VCO to sweep the whole frequency spectrum. The sent/received signals are compared to visualize the tag's response [1].

Self-interference comes from a leaked transmitter signal to the receiver. This interference significantly degrades the sensitivity of a reader. Forouzandeh et al. [9] recently used a lookup table for the optimum compensation vector in each frequency to reduce this noise. In this book we use cross-polarized Tx/Rx, which suppresses the noise from the transmitter to the receiver.

3.2.2 Image-Based Readers

Image-based readers add frequency spectrum data to the relative location of the reader-to-tag to make a 2-D image of the tag. This process normally requires the tag or the reader to be placed in a linear rail, the use of a few antennas to compensate for the physical movement, or antenna arrays and adaptive beamforming techniques.

At least three readers have been reported in industry [2, 10, 11]. Due to commercial concerns, the technical information is not that clear from the patents.

The first reader was developed by Somark Innovations Inc. [2, 10]. Somark has used its image-based barcode-like chipless tag reader using mechanical steering or switched beam transmit antennas. The tags are printed like a tattoo into livestock ears. The ink used is compatible with multiple frequencies ranging from 100 KHz to a few GHz [12].

The second reader is made by Vubiq Inc. [13]. Vubiq has used the phase and polarization in the reader and its tags for making 2-D representation and revealing the ID of the tag. The third reader is made by InkSure Technologies [2, 11]. Inksure has developed an antenna array system to read the tag in different positions. The tag used in their system is printed with special shapes, such as lines and triangles, with their proprietary conductive ink.

The reader developed for this book is different than all the abovementioned technologies. It uses a 60-GHz band for the first time, cross-polarized antennas, and a linear rail to scan the tags and to form 2-D representations of the tag using SLAR and augmented data. The advantages of our reader to the available market solutions are its higher frequency band (so smaller active components, antennas, and tags), modular design, and open code, which makes it easier to replicate.

3.3 A 60-GHz System Block Diagram

A simplified version schematic of the developed reader is illustrated in Figure 3.3. A mm-wave VCO (or digital frequency synthesizer) makes the required main RF frequency in the 60-GHz band with 1.5-GHz steps. So, the main mm-wave VCO frequencies are 56.5, 58, 59.5, 61, and 62.5 GHz. The LO (which is a digital synthesizer, ADF4350) produces the required continuous wave for the precision sweep, which is added to the main frequency with a mixer in the transmitter. The generated frequency of the LO is between 500 MHz to 2 GHz. The LO functionality is to make higher-resolution signals within the mmVCO frequency steps of 1.5 GHz. This modulated signal is amplified and sent to the Tx antenna and toward the chipless tag.

The backscattered signal is received by the receiver, amplified by low noise amplifier (LNA) and downconverted by the original mm-wave VCO signal. The output of Mixer 2 is in the intermediate frequency (IF) band (the same as the LO frequency). A gain/phase comparator compares the sent and received signals, and then sends their gain/phase difference to an analog multiplexer (AMux). A high-precision 20-bit ADC converts this analog signal

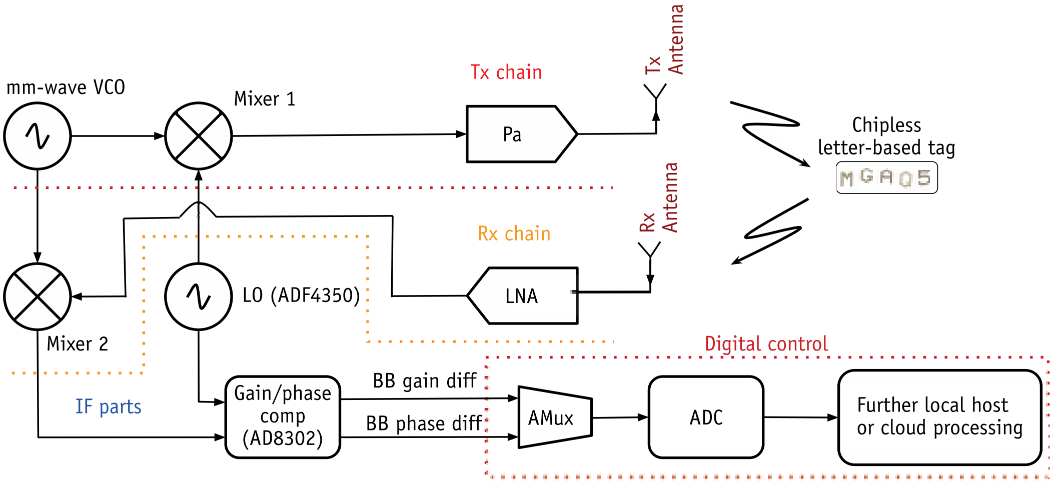


Figure 3.3 The simplified reader schematic diagram implemented in this book. The gain/phase comparator creates two analog baseband gain and phase signals from the sent and received signals, which is used to decode the tag.

to digital. These digitized gain and phase differences are processed by local or remote cloud computers.

Before explaining the component details, two main RF design considerations of the reader need to be investigated. These are the maximum allowed power or equivalent isotropically radiated power (EIRP) of the reader, and the desired detection reading distance between the reader and tag. These two parameters determine the link budget requirements and derive the specification requirements of the Tx and Rx modules, and choose adequate power levels for the voltage gain amplifier (VGA), power amplifier and LNA stages, and the desired antenna gains. In the following section, link budget calculation is performed.

3.3.1 Maximum Reader Power and Link Budget Calculations

Tag detection for chipless RFID in frequency, time, and image-based domains normally needs a wideband frequency spectrum (3–10.5, 22–26.5, and 57–64 GHz) [14, 15]. This wideband spectrum is needed in the time-domain to make a sharp pulse, in the frequency domain to detect more frequency nulls (in higher-Q multi-resonator tags), and in the image-based domain to reconstruct the image from time or frequency responses.

The EIRP, which is the product of transmitter power and the antenna gain in a given direction, is measured relative to the isotropic antenna of a radio transmitter (with antenna gain in dBi units). The EIRP can be calculated as follows [16]:

$$\mathrm{EIRP} = P_t + G_t - L_c \tag{3.2}$$

where P_t is the transmitted power level in dBW or dBm, G_t is transmitter antenna gain in dBi, and L_c is the cable and connector loss. If P_t is not known, a measurement should be done based on the following formula:

$$P_r = P_t + G_t + G_r - L_c - L_{Atten} \tag{3.3}$$

in which P_r is the measured received power level in dBW or dBm, G_r is the gain of the Rx antenna in dBi, and L_{Atten} is the free-space propagation path loss in dB, which can be calculated from the following formula:

$$L_{Atten} = 20\log(f) + 20\log(d) + 32.4 \tag{3.4}$$

in which the frequency f is in GHz and the measurement distance d is in m. The path loss, L_{Atten} (or air attenuation as the media is air) can be a serious limitation factor in the higher mm-wave bands such as the 60-GHz spectrum. Using (3.4), the air attenuation versus distance is calculated and presented in Figure 3.4. The actual distance is the distance from the reader to the tag plus the distance from the tag to the reader.

From Figure 3.4, the air attenuation in the 60-GHz spectrum is almost 10 to 20 dB higher than other UWB of the frequently used lower bands. The RFID is not restricted to these three major bands. In the United States, some other radio location bands used for RFID include 2.9–3.65, 5.25–5.925, 8.5–10.55, 33.436, 59–64, 76–81 GHz, 92–100, and 231–241 GHz [15]. The EIRP limitation is, however, not the same in different countries. Based on Australian spectrum regulations [17, 18], there is a limitation of 20 mW/cm^2 (averaged over 6 min) for occupational exposures and 5 mW/cm^2 (averaged over 30 min) for public exposures. In our case, using an antenna with 10 dBi of gain with a 3-dB bandwidth of 55 [19], the maximum transmission power should be

$$\text{Max Tx power} = 13\text{ dBi} - 10\text{ dBi} = 3\text{ dBm} \tag{3.5}$$

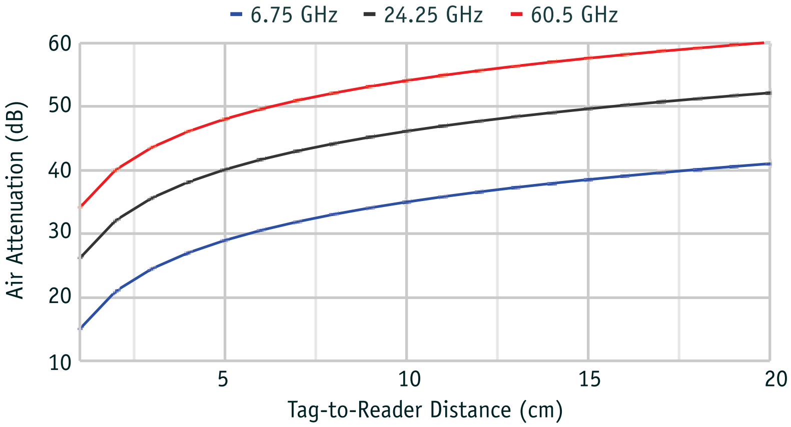


Figure 3.4 Air attenuation versus distance for the middle-band frequencies of the common chipless RFID spectrum bands based on (3.4). The 60-GHz spectrum has almost 10 to 20 dB (10 to 100 times) more attenuation than the other common bands.

This is because 20 mW = 13 dBmW.

In many circumstances, however, sending the higher power is not recommended as more unwanted backscatters from the environment are captured, and sending the lower power restricts the detection distance. So, there is always a compromise between the desired reading distance and the maximum available transmission power. In practice, power less than –5 dBm for the Tx in a 5- to 10-cm distances from the tag to the reader is used in this book, which is chosen based on the experiments in our lab.

3.3.2 Maximum Reading Distance Calculations

Another consideration for chipless RFID is the reading distance. Ideally a tag's reading distance is desired to be as far as possible, except for some secured applications. Calculations for the maximum possible reading range are as follows: From the radar range equation, the ratio of the received and transmitted power from a target with RCS σ is [20, 21]:

$$\frac{P_r}{P_t} = \sigma \frac{G_{Ant,t} G_{Ant,r}}{(4\pi)^3} \left[\frac{\lambda \cos(\theta)}{R_1 R_2} \right]^2 \tag{3.6}$$

In this equation, P is the power, σ is the tag's RCS, G is the antenna gain, t and r represent the transmitter and receiver, and R_1 and R_2 are the tag distances to the Tx and Rx, respectively. The term $cos^2(\theta)$ is the polarization loss factor (PLF) between the tag and the reader antenna. Consider a stripline tag in front of the reader, as per Figure 3.5. If the Tx/Rx antennas use the same polarization (E_{Tx} and E_{Rx} are in the same direction) and the tag is set straight, then the PLF is 1. For 90° angle between the reader antenna and the tags, the PLF is zero and ideally no power will be received by the Rx antenna.

In a bistatic mode reader, in which the Tx and Rx antennas are placed at different distances toward the tags, R_1 and R_2 are different. In this work, the Tx and Rx are closely set to each other, so $R_1 = R_2$ and cross-polarized Tx/Rx is used, with tags in the straight position. If the tags were just striplines, no power would be received. Because the tags are alphanumerical, they would still be detectable, as they have components in both polarizations.

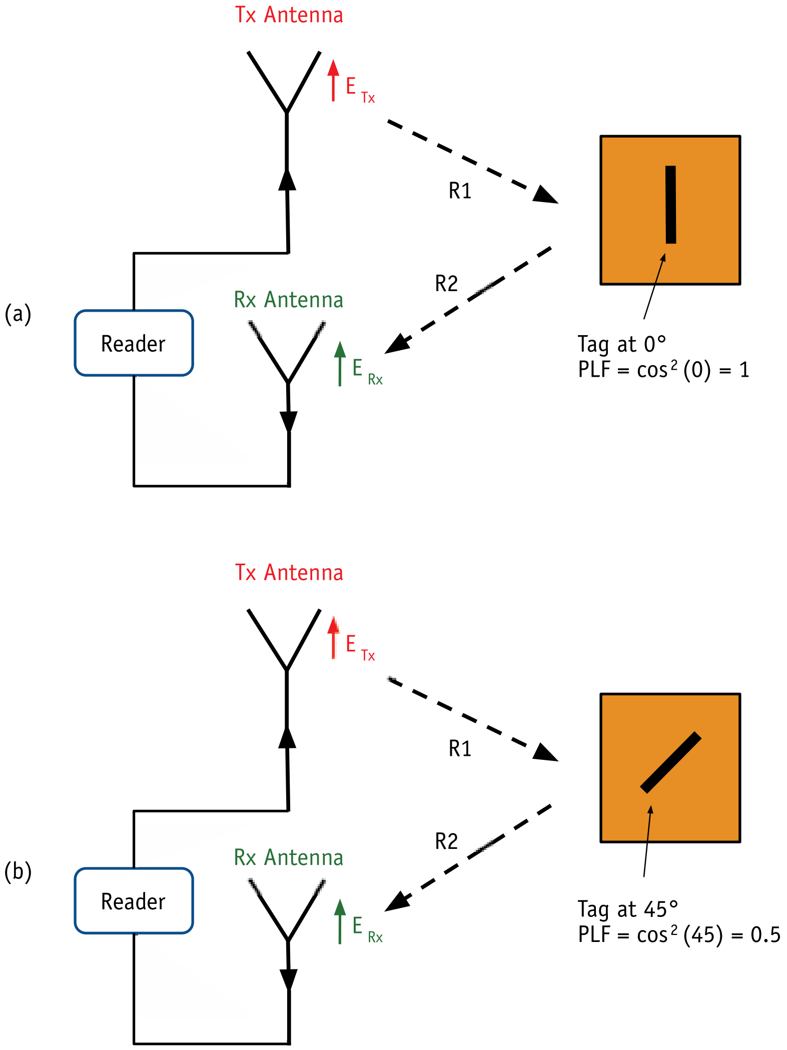


Figure 3.5 Illustration of the polarization loss factor concept in a bistatic reader: (a) the copolar Tx/Rx and the tag are in a straight position, and (b) the cross-polar Tx/Rx and the tag are in a 45° position (used in this book). In (b), the tag won't be seen if it is put in a vertical or horizontal position.

Table 3.1 summarizes the link budget, noise, signal-to-noise ratio (SNR), and maximum reading distance to the tag using (3.6). The tag's RCS is assumed[1] to be $\sigma_{tag} = 0.001$ m^2. The calculations are based on 60 GHz ($\lambda = 0.005$ m) and a reading angle of cos(45°) = 0.5. Based on these calculations, the measurements in this book are

[1] An insect's RCS is 0.00001 m^2 and a bird's RCS is 0.01 m^2 [22].

Table 3.1

Calculated Link Budget Parameters for Our 60-GHz Reader*

	Value	Unit	Comments
Transmit power	0	dBm	HMC6300 [23]
Antenna gain	13	dBi	[19]
Path length	0.1	m	
Free-space attenuation	–48.18	dB	Based on (3.4)
Oxygen attenuation	–0.0016	dB	Oxygen attenuation, 16 dB/km at 60 GHz
Received power	–35.18	dBm	
Receiver sensitivity	–60	dBm	Not provided [24]
Thermal noise	–82.07	dBm	10 log10(kBT), k is Boltzmann's constant
Noise figure	10	dB	HMC6301 [24]
SNR	46.89	dB	Received_power - noise_level
Maximum reading distance	4.5	cm	Using (3.6), with $\sigma_{\text{tag}} = 0.001$ and $\theta = 45^\circ$

*With a 0-dBm Tx, the tags at 5-cm distances should be detected with a received power of –35 dBm.

usually done for a 5-cm distance. For higher distances up to 10 cm between the tag and the reader, the power is adjusted accordingly to maintain the link budget.

In the next section, the details of our 60-GHz Tx/Rx boards will be explained.

3.4 60-GHz TX/RX Boards

After several designs of 60-GHz circuits and considering other parameters such as implementation time, flexibility, and risk in the development of a 60-GHz reader, it was decided to design the 60-GHz board from scratch with individual components. An HMC6350 Radio Link Transceiver [25] was selected as the only available candidate reader architecture at the time of this development.

The HMC6300/HMC6301 is a complete millimeter-wave transceiver that operates from the 57 to 64 GHz with up to 1.8 GHz modulation bandwidth. An integrated synthesizer provides tuning in 250-, 500-, or 540-MHz steps to support up to 64 quadrature amplitude modulation (QAM) modulation. The differential output provides up to 15-dBm linear output power into a 100-Ω termination. Single-ended operation is also supported up to 12 dBm. The Tx and Rx boards can be used for several applications, such as small cell backhaul, multi-Gb/s data communication, WiGig/802.11ad radio, high-definition video transmission, and radar or high-resolution imaging.

Moving on from these 60-GHz Tx/Rx boards, the next sections explain how different digital components and RF boards are utilized to implement a chipless RFID system.

3.5 Designing and Integration: RF, IF, Controller, and Peripheral Circuits

In this section more details of the developed reader are presented. Figure 3.6 illustrates the overall hardware system. The main focus in this figure is the interconnections between the digital and ana-

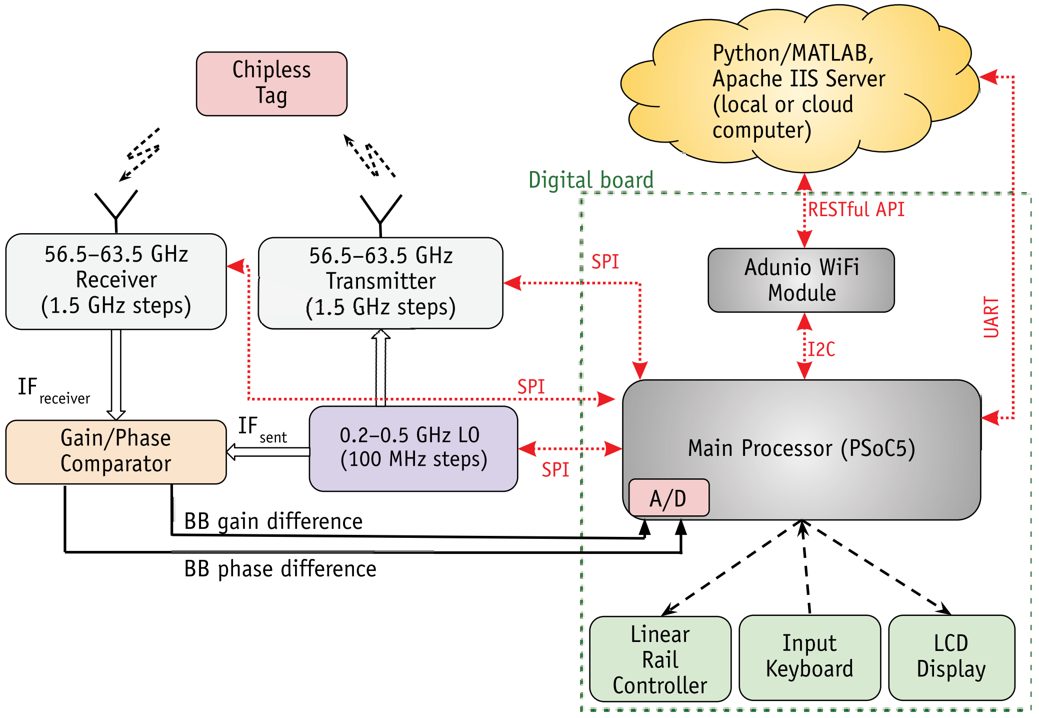


Figure 3.6 Cloud-based RFID reader architecture used in this book. The heart of the reader is a PSoC5 processor, which communicates with different boards through various protocols (SPI, I2C, UART). For cloud connectivity, an Arduino Wi-Fi module (ESP8266) is added to the PSoC5 through the I2C protocol, which can communicate to the http web server through RESTful API.

log parts, with Figure 3.7 showing the frequency steps of Tx and Rx over time.

The heart of the system is its main processor, which is accomplished here using a programmable system-on-chip (PSoC). In this work a programmable system-on-chip 5 low power (PSoC-5LP) is used as it can provide enough input/output (IO) pins and has some integrated ADC inside. Communicating to the RF and IF boards is done through serial peripheral interface (SPI) lines of the main processor.

A PSoC5 also controls less intelligent devices, such as the keyboard, the 20 × 4 lines liquid crystal display (LCD), and the linear rail controller through its different digital pins. There is a character LCD component in a PSoC5 to easily write on the LCD, and a C-based function is needed to run the stepper motor driver. A PSoC5 can directly send the data to the local computer through its universal asynchronous receiver transmitter (UART) component. As a PSoC5 does not have Wi-Fi connectivity, an Arduino-based Wi-Fi module is added (ESP8266). The communications between the PSoC5 and the Arduino ESP8266 is through the Inter-Integrated Circuit (I2C) protocol.

A tag's scanned data is put in a JavaScript Object Notation (JSON) file format to be sent to the computer running an Apache web server through a representational state transfer (REST) application programming interface (API). A REST API (also known as RESTful API) is an application programming interface that adheres to the constraints of REST architectural style and allows for interaction with RESTful web services.

In the next sections, further details are provided for these hardware parts.

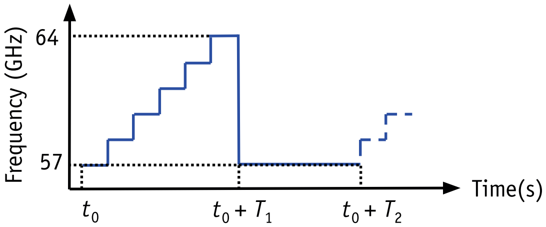


Figure 3.7 Illustration of the transmitter in Figure 3.6 over time. There are 1.5-GHz steps, and the LO produces the smaller steps (10 MHz) within these big steps. A time is also required to move the tag in the linear rail, which is in the [t_0+T_1, t_0+T_2] interval.

3.5.1 60-GHz Transmitter/Receiver

HMC6350 evaluation boards from Analog Devices [25] are the main RF parts of this book. Their main ICs are HMC6300 (for transmission) [23] and HMC6301 (for reception) [24] that operate from 57 to 64 GHz with up to 1.8 GHz of modulation bandwidth. Available frequency steps are 250, 500, and 540 MHz with low phase noises (lower than –93 dBc/Hz at 1 MHz and beyond). Output power is linear up to 15 dBm power for a 100-ohm load (using a MMPX cable and 1.85-mm antennas).

HMC6300 boards are initially designed for the half-duplex 60-GHz millimeter-wave links. The Tx and Rx boards are separated but they are installed on the same mainboards, which initially requires two different computers for the operation. Eventually using a virtual windows machine, controlling both boards via one physical computer is made possible. HMC6300 boards can have four baseband (BB) or frequency modulation (FM) in-phase and quadrature (IQ) signals as the inputs to the Tx circuit and can deliver those BB or FM signals in the Rx circuit after demodulation from the 60-GHz band.

HMC6300 boards are quite complicated, as they are mainly designed for backhaul data transfer for up to 2-GHz baseband signals. To more easily utilize them for a 60-GHz Tx/Rx board, a simplified schematic of the HMC6300 transmitter is illustrated in Figure 3.8. The baseband IQ signals are multiplied with their corresponding 16-GHz signal initially. An IF amplifier and a bandpass

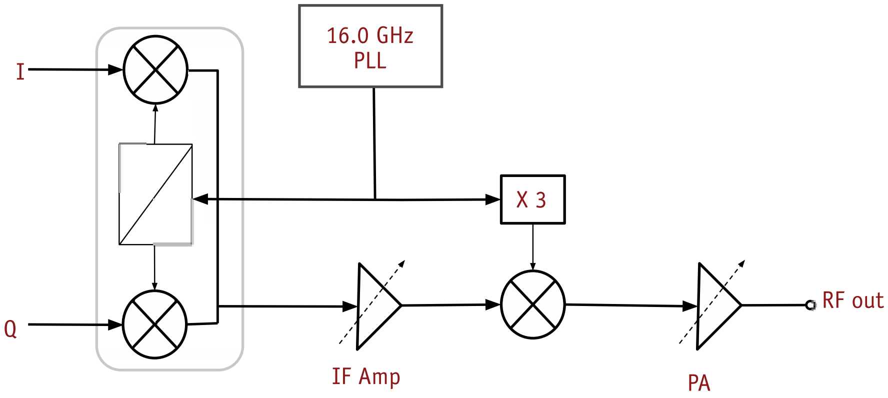


Figure 3.8 Simplified schematic of the 60-GHz transmitter. The IF amplifier and PA are programmed using the SPI port.

filter amplifies the signal and a frequency multiplier brings the signal to the 60-GHz range. The final signal is amplified by a power amplifier (PA) and goes to the antenna. The IF amplifier and PA are programmable using SPI, and their values should be determined for the best results, as this will be needed later in this section.

On the receiver side, which is illustrated in Figure 3.9, the signal after the antenna (RF in) initially amplifies with a variable gain LNA. The frequency then steps down to 16 GHz by multiplying the received signal with a 48-GHz clock. The I and Q signals are available after multiplication with an IQ 16-GHz mixer. There is a baseband voltage gain amplifier (BBVGA) in the end chain for further amplification.

The higher the transmitting power in the Tx, the more harmonics are available in the received RF wave, and also more phase noise. Higher Tx powers also create more background noise, as more reflection from peripheral objects will be received. On the Rx side, having high gains for the LNA or the IF amplifiers normally leads to more harmonics and nonlinearity in the IF received signal. Therefore, for a few centimeters' distance of the tag and Tx/Rx antennas, a trade-off has to be made between the Tx and Rx gains and the noise level.

The HMC6300 (Tx) has two variable gain controllers in the IF and RF stages, as shown in Figure 3.8. The higher the attenuation in the amplifiers, the lower the gain will be. The HMC6301 (Tx) has

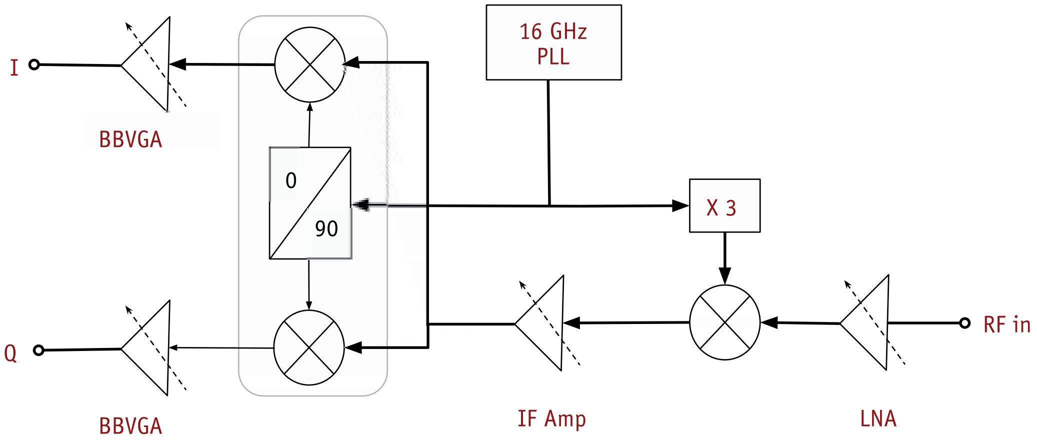


Figure 3.9 Simplified schematic of the 60-GHz receiver. The LNA, IF, amplifier, and BBVGA stages are programmed through the SPI port.

four variable gain controllers, in the IF and RF stages, as shown in Figure 3.9. The optimized gain values, chosen through trial and error, are found in Table 3.2.

Figure 3.10 shows how the 60-GHz board is used in the experiment. Utilizing these 60-GHz boards caused a few technical issues that are discussed in the next sections along with the proposed solutions. This is done so readers can overcome the hardware challenges presented by the Tx/Rx boards.

3.5.1.1 Using Virtual Machines

To get used to the HMC6300 (Tx) and HMC6301 (Rx) board initially, Analog Devices has developed SoC transmitter and SoC receiver programs associated with the Tx and Rx boards, respectively. These programs are supposed to be run on two individual computers, perhaps to reduce the interference between the Tx and Rx boards. However, using two separate computers to control the boards was inconvenient; in this project we needed to change the setting of the Tx and Rx boards often, and at the same time access to other peripherals had to be monitored through one of these computers.

As Tx and Rx boards use the same COM port number, a COM port conflict issue occurs if one computer is used to control the two Tx and Rx boards. As said in the manual [25], using two separate computers for controlling Tx and Rx boards is recommended. In this work, VirtualBox, a freeware virtual machine (VM) from Oracle is used inside the host computer to address this. VirtualBox is a general-purpose full virtualizer for x86 hardware, targeted for

Table 3.2
Optimized Amplifier Gains Used in Tx and Rx Circuits

System	Corresponding Stage Amplifier	Recommended Gain/ Attenuation (dB)
Tx	IF	5.2
Tx	RF (PA)	10
Rx	LNA	12
Rx	IF	10.4
Rx	BBVGA coarse attenuation	12
Rx	BBVGA I fine attenuation	5
Rx	BBVGA Q fine attenuation	5

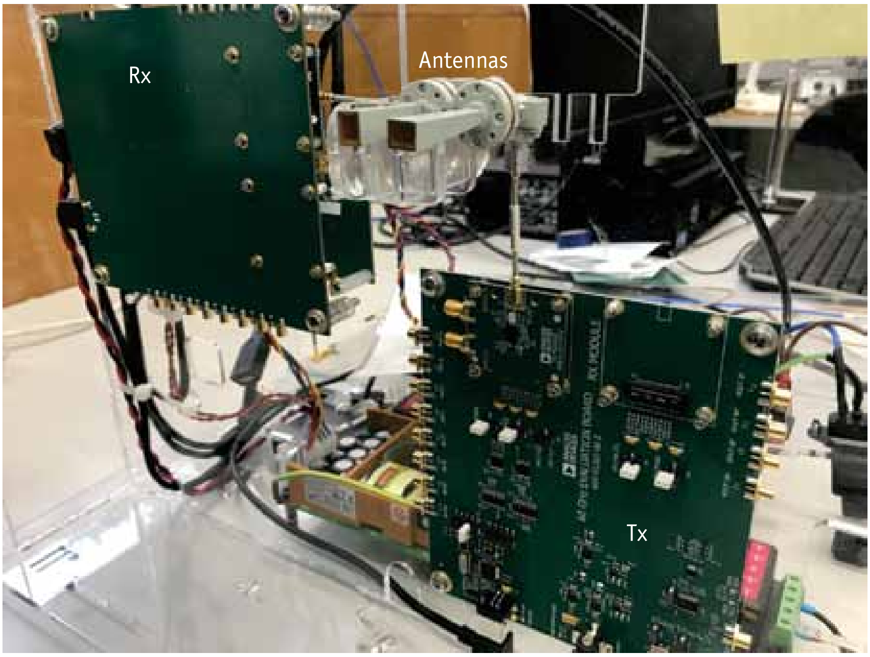


Figure 3.10 60-GHz Tx and Rx with cross-polarized antennas.

server, desktop, and embedded use [26]. Figure 3.11 illustrates a snapshot of the VM used.

Without going into detail, if your computer hardware and operating system (OS) supports Hypervisor technology using Hyper-V, VMware ESXi, or Citrix Xen-server, VMs are recommended.

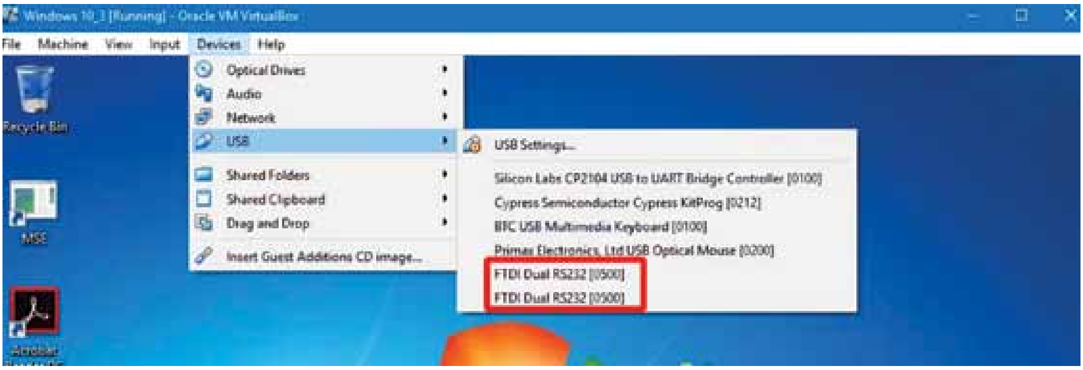

Windows 10_1 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Recycle Bin
MSC
Acrobat
Optical Drives
Audio
Network
USB
Shared Folders
Shared Clipboard
Drag and Drop
Insert Guest Additions CD image...
USB Settings...
Silicon Labs CP2104 USB to UART Bridge Controller [0100]
Cypress Semiconductor Cypress KitProg [0212]
BTC USB Multimedia Keyboard [0100]
Primax Electronics, Ltd USB Optical Mouse [0200]
FTDI Dual RS232 [0500]
FTDI Dual RS232 [0500]

Figure 3.11 Snapshot of the VirtualBox VM running inside the host computer, showing the conflicting ports for Tx and Rx boards. We ran the SoC receiver program within the VM and took one of these FTDI ports first. The SoC transmitter ran through the host OS and took the other FTDI port with no conflicts.

These are type 1 hypervisors. Otherwise, a type 2 hypervisor, such as Oracle's VirtualBox, will work. Type 1 hypervisors work directly with the computer hardware, whereas type 2 rely on the host OS to communicate with the hardware. Therefore, type 1 VMs normally consume less resources.

3.5.1.2 SPI Signaling

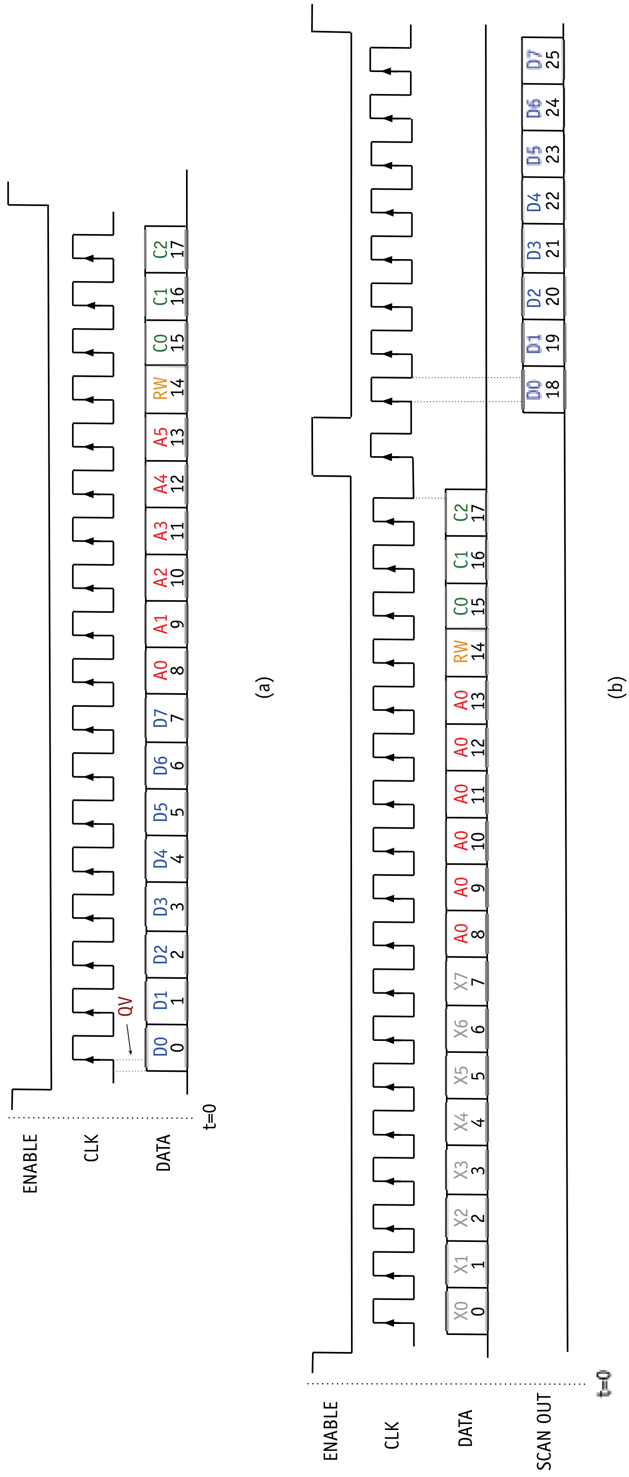
The next step in the hardware part of this book is how to control the 60 GHz Tx and Rx boards through the SPI lines. Using this step, the need for a computer is eliminated. As illustrated in Figure 3.6, both the Tx and Rx boards are connected to the main processor (PSoC5) through their SPI interfaces. The register arrays for both the Tx and Rx are organized into 32 rows of 8 bits. Using the SPI interface, the arrays are either written or read one row at a time.

An illustration of timing diagrams to write an 8-bit data to the Tx/Rx ICs of HMC6300 and HMC6301 using ENABLE, CLK, and DATA lines is shown in Figure 3.12. Initially the ENABLE line goes low, then the first of 18 data bits (bit 0) is placed on the DATA line, the CLK line goes high to clock in data bit 0. For the correct operation, the DATA line remains stable for at least 2 ns after the rising edge of the CLK signal.

The 18 data bits and 18 clock pulses are required for a write operation, as shown in Figure 3.12. These 18 data bits contain the 8-bit register array row data, with the LSB clocked in first. It is followed by the register array row address (ROW 0 through ROW 23,000000 to 010111, LSB first), the R/W bit (set to 1 to write), and finally the Tx chip address 110 (LSB first). For the Rx, the chip address is 111 instead.

The ENABLE line returns high after the 18th clock pulse to load the register array on the Tx/Rx. A minimum of 2-ns line stability is needed for the DATA line before the rising edge of the CLK line and also the CLK line prior to the rising edge of the ENABLE line [23]. In the reading mode the ENABLE line should go to 1 with a complete CLK and then go to 0 again to read the bits in the SCAN OUT line.

As the SPI is based on a 1.2V complementary metal-oxide-semiconductor (CMOS) technology, a voltage divider should be used to interface the PSoC5 and the Tx/Rx boards. A serial resistance of 1 kΩ can be used instead to compensate for the voltage level difference.



(a)

(b)

Figure 3.12 Timing diagram for (a) writing to (b) reading from Tx/Rx boards. D0–D7 are the register array row data bits (least significant bit (LSB) to most significant bit (MSB), respectively), A0–A5 are array address bits, C0–C2 are chip address, and X0–X7 represent dummy bits [23]. Each bit in the data line should remain stable for at least 2 ns before the rising edge of the CLK signal.

The trouble with the PSoC5 SPI master component is that it does not accept a customized SPI configuration, as the maximum bits its SPI component handles are 16 bits. According to Figure 3.12, a sequence of 18 CLKS (and DATA bits) is needed in the writing mode and 27 CLK (and DATA) bits for the reading/verification modes of the Tx and Rx boards.

To do this customized SPI, a few functions are added to the PSoC5 main C program. SDI_TX and SCK_TX pins of PSoC5 (Figure 3.22) are used for DATA and CLK lines, respectively. To write a bit 0, SDI_TX goes to zero, SCK_TX goes from 0 to 1 and then to 0 to make a complete clock. This is similar for the writing of a bit 1, as per the following code listing:

Code Listing 3.1: Write a bit 0 to the 60-GHz Tx board.

```
int  Putbit0_TX ( )
{
     SDI_TX_Write ( 0 ) ;
     SCK_TX_Write ( 0 ) ;
     SCK_TX_Write ( 1 ) ;
     SCK_TX_Write ( 0 ) ;
     return ( 0 ) ;
   }
```

With a PSoC5 clock speed of 80 MHz [27], the CLK pulse duration (>12.5 ns) satisfies the stability criteria of 2 ns mentioned previously for the DATA line. The complete C-language code list for this book has been provided in Appendix A. Table 3.3 provides the list of functions used to program the HMC6300 60 GHz Tx module from Appendix A.

Similar functions are developed for the 60-GHz Rx module if the Tx in the function name in Table 3.3 changes to Rx. For example, the Putbit0_RX function puts a bit 0 in the 60-GHz Rx module, and so on. The difference between Tx and Rx functions is the 3-bit chip address as seen in Figure 3.12. The Rx chip address is 111 whereas Tx is 110.

3.5.1.3 Locking Time Calculations

The HMC6350 Tx/Rx modules are initially intended as a backhaul transmitter/receiver [23]. Their minimum step frequency is 250 MHz in the 56.5- to 66.5-GHz range. It is observed that in the

Table 3.3
PSoC5 List of C Functions for the Tx Board in Appendix A

Function Name	Arguments	Description
Putbit0_TX		Puts a bit 0 into the 60-GHz Tx
Putbit1_TX		Puts a bit 0 into the 60-GHz Tx
WriteReg_TX	Reg_no, R	Writes 8-bit R data into register Reg_no
ReadReg_TX	Reg_no	Reads 8-bit R data from register Reg_no
WriteAndVerify_TX	Reg_no, R	Writes 8-bit R data into register Reg_no and then verifies it by reading from the same Reg_no

extreme range of this frequency spectrum the device is unable to lock properly, but it works fine in the ISM band of 57–64 GHz. If higher than 250-GHz resolution is needed, a VCO should be used in conjunction with the Tx/Rx modules, as illustrated in Figure 3.6.

As per our experimental observations, it takes a few seconds for the transmitter and receiver to update to the new frequency (register number 20 in HMC6300 and HMC6301). This time is more or less around 10 seconds, so it is generally advised to keep the step frequency of the Tx/Rx boards as high as possible, and use a LO to do the high-resolution frequency sweeping using the I or Q inputs of Tx module shown in Figure 3.8.

As there is a limitation of 1.8 GHz in the input IQ signals, in this work 1.5-GHz frequency steps are chosen for the Tx/Rx (five 1.5-GHz steps from 56.5 to 62.5 GHz) and the LO sweeps from 0.5 GHz to 2.0 GHz to cover the desired resolution (more details are provided in Section 4.3.2 in the next chapter). In practice, it is observed that one input to I or Q inputs for Tx board (Figure 3.8) is sufficient. The same I or Q should be used in the output of Rx (Figure 3.9) as in the received IF-band signal. Using one input (I or Q) leads to a bit higher phase noise in the received signal compared to sending/receiving both the I and Q at the same time, as per our observations, but the circuity will be simpler.

3.5.2 Local Voltage-Controlled Oscillator

The ADF4350 local oscillator from Analog Devices was used in this work to create IF frequency using its high-precision phase-locked loop (PLL)-VCO. Its fundamental frequency ranges from 2,200 to

4,400 MHz, and its divide-by-1/2/3/8 or 16 circuits allows for creating outputs from 137.5 to 4,400 MHz. A precision channel spacing as low as 500 Hz over the frequency range of interest (500 to 2,000 MHz) is possible. The ADF4350 has a simple SPI compatible for writing to the device. The CLK, DATA, and LE pins are used to control the data transfer via SPI lines. When the CE pin is high, the 32 bits that have been clocked into the appropriate register on each rising edge of CLK are transferred to the appropriate latch. The CE is always kept high through a pull-up resistor. The lock status is monitored via VCO_lock pin, as shown in Figure 3.13.

To fully operate the ADF4350 VCO, six registers of 32 bits are programmed. Table 3.4 shows the list of functions used in the PSoC5 for writing the data to the registers.

Unlike HMC6300 and HMC6301 (3.12), there is no way to verify if the registers are written correctly in ADF4350. The MAXOUT pin can be used as a lock detection indicator only, which is the input to the PSoC5, as shown in Figure 3.13. The developed LO circuit using ADF4350 is shown in Figure 3.14. A linear power supply is also used in the circuit to make a 3.3V supply for the board out of 5V input from the mainboard. The schematic circuit for this LO circuit is shown in Figure 3.15.

Figure 3.16 illustrates sent and received waveforms for a 61-GHz signal sent to a tag at the distance of 5 cm. This 61-GHz is created using a BB signal at 1 GHz from the LO after modulation with a 60-GHz Tx. Almost 30 dB (28.1 dB) gain loss is observed in this trial. The difference between amplitude and phase of these two will be detected by the gain/phase circuit described in Section

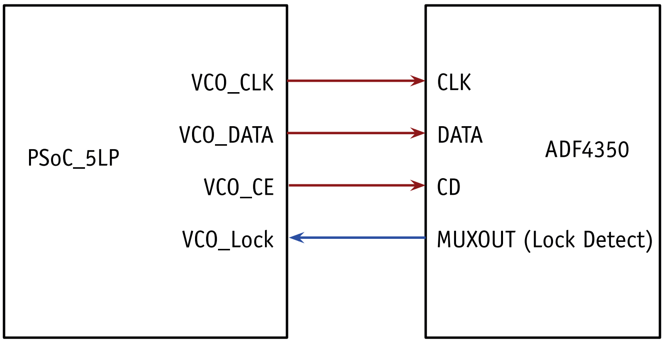


Figure 3.13 LO ADF4350 SPI connections with PSoC5. MAXOUT pin will indicate successful frequency lock after data has been received by the ADF4350.

Table 3.4
PSoC5 List of C Functions for the LO

Function Name	Input Arguments	Description
Putbit0_ADF4350		Inserts a bit 0
Putbit1_ADF4350		Inserts a bit 1
WriteReg_ADF4350	R*	Writes 32-bit R (including DATA and Reg_no)

*R is the register value.

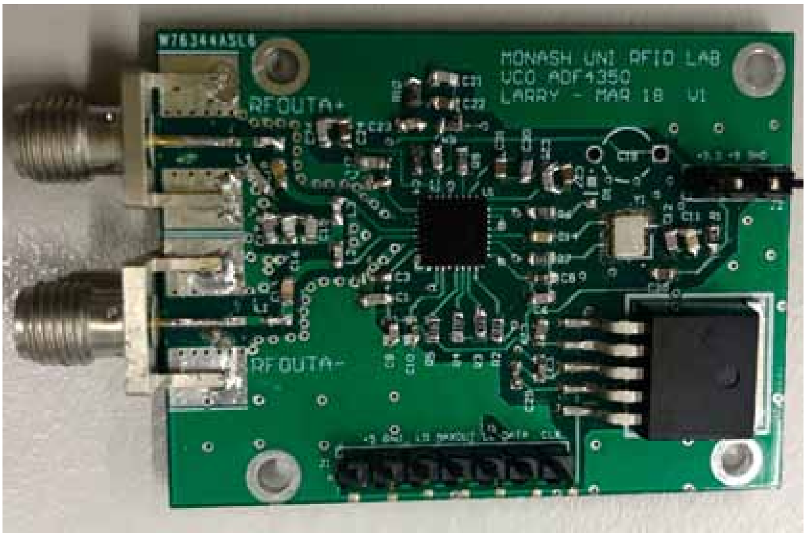


Figure 3.14 Developed LO board for this project using the ADF4350 IC.

3.5.3 and will be further processed as the whole architecture shown by Figure 3.6.

3.5.3 Gain/Phase Comparator

The gain/phase comparator in Figure 3.6 compares the sent and received signals and provides their difference. This is done using an AD8302 IC, which is a logarithmic gain/phase differential detector that accepts a wide input range of –60 to 0 dBm. A schematic of the comparator is illustrated in Figure 3.17.

The gain and phase accuracy are 30 mV/dB and 10 mV/Deg, respectively, with a typical nonlinearity of less than 0.5 dB for the gain and less than one degree for the phase [28]. The developed comparator board based on the AD8302 is illustrated in Figure 3.18.

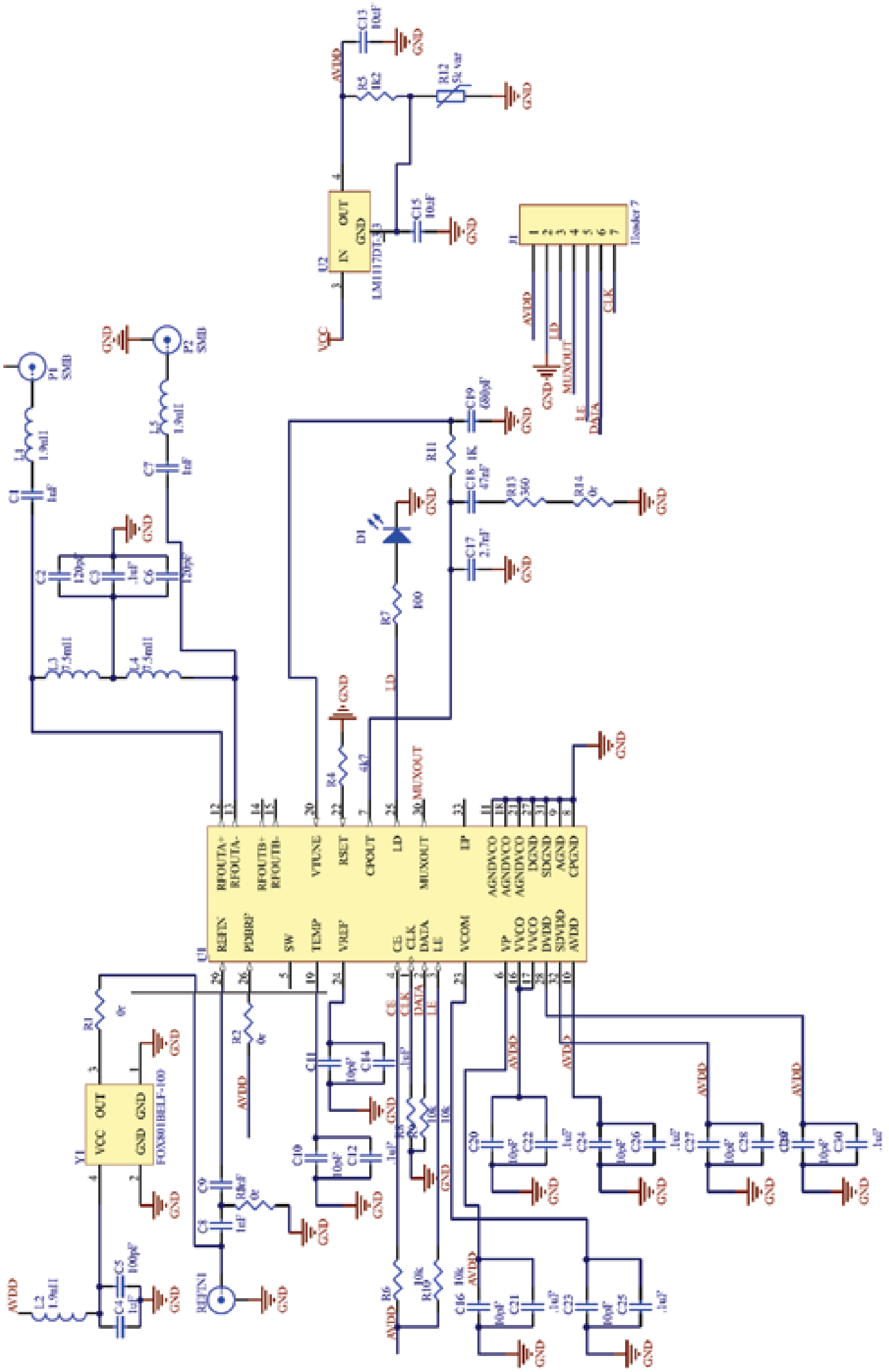
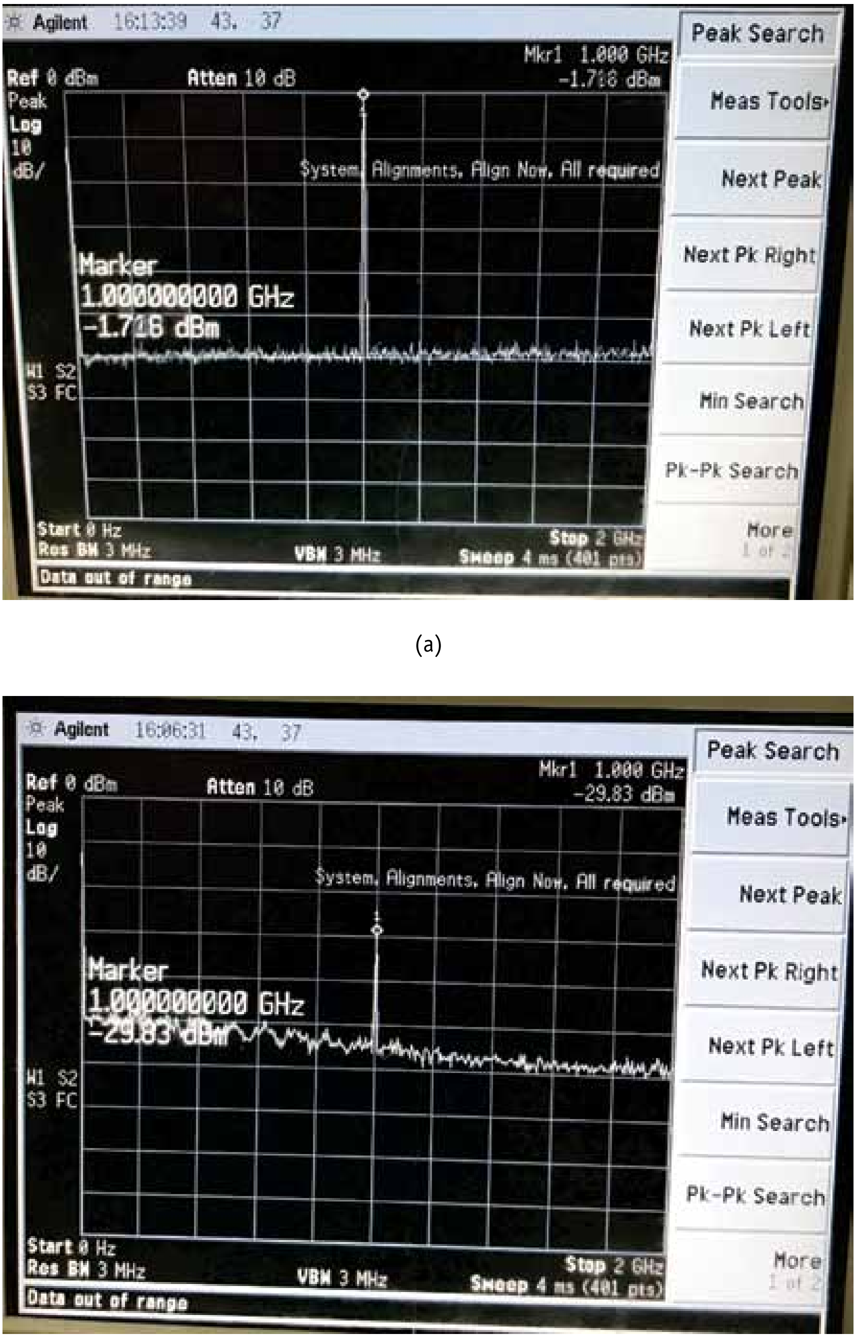


Figure 3.15 Schematic circuit diagram of Figure 3.14 used as a LO.



Agilent 16:13:39 43. 37

Ref 0 dBm Atten 10 dB

Mkr1 1.000 GHz
-1.716 dBm

Peak
Log
10
dB/

System, Alignments, Align Now, All required

Marker
1.000000000 GHz
-1.716 dBm

W1 S2
S3 FC

Start 0 Hz
Res BW 3 MHz VBW 3 MHz Stop 2 GHz
Sweep 4 ms (401 pts)

Data out of range

(a)

Agilent 16:06:31 43. 37

Ref 0 dBm Atten 10 dB

Mkr1 1.000 GHz
-29.83 dBm

Peak
Log
10
dB/

System, Alignments, Align Now, All required

Marker
1.000000000 GHz
-29.83 dBm

W1 S2
S3 FC

Start 0 Hz
Res BW 3 MHz VBW 3 MHz Stop 2 GHz
Sweep 4 ms (401 pts)

Data out of range

(b)

Figure 3.16 (a) LO sent and (b) received signals to the tag at 1 GHz, modulated by a 60-GHz signal (RF sent at 61 GHz). A loss of –1.716 dBm + 29.83 dBm = 28.1 dB is visible for the received signal.

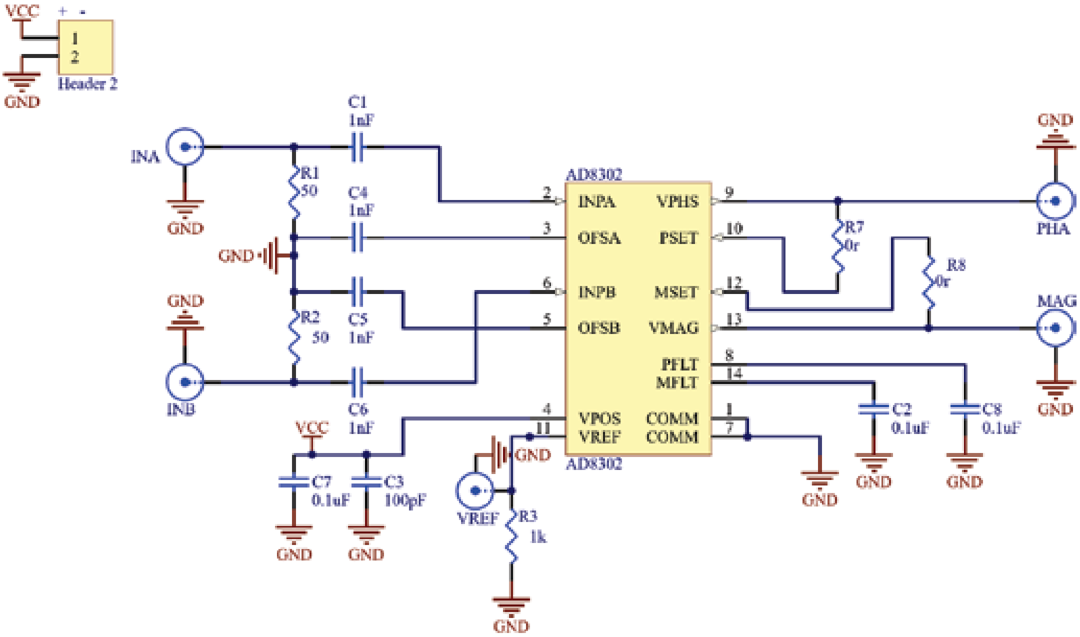


Figure 3.17 Schematic diagram of the AD8302 board (gain/phase comparator).

3.5.4 Digital Control Board

The digital board in Figure 3.6 controls the RF and LO circuits and processes the ADC signals before sending them to the local host or cloud computer. Three versions of digital boards have been developed in this book, each with added functionalities. The latest version is illustrated in Figure 3.19 with different components identified. In the next sections different hardware parts of the main digital control board are explained. The design evolution of the main digital circuit is shown in Figure 3.20.

3.5.4.1 Main Processor: PSoC-5LP

As mentioned, the Cypress PSoC-5LP (CY8C58LP) is used as the main hardware processor part (Figure 3.19). This chip includes a 32-bit ARM Cortex-M3 core and a mixed-signal array of configurable analog and digital peripherals chips. In particular, one delta-sigma ADC (8 to 20 bits, 192 kSamples per second (SpS) 12-bit) is of interest in this work. As there is only one 20-bit ADC used for higher required precision, an AMux (AMuxSeq in Figure 3.21) is used to connect the same ADC for multiple inputs, namely, gain and

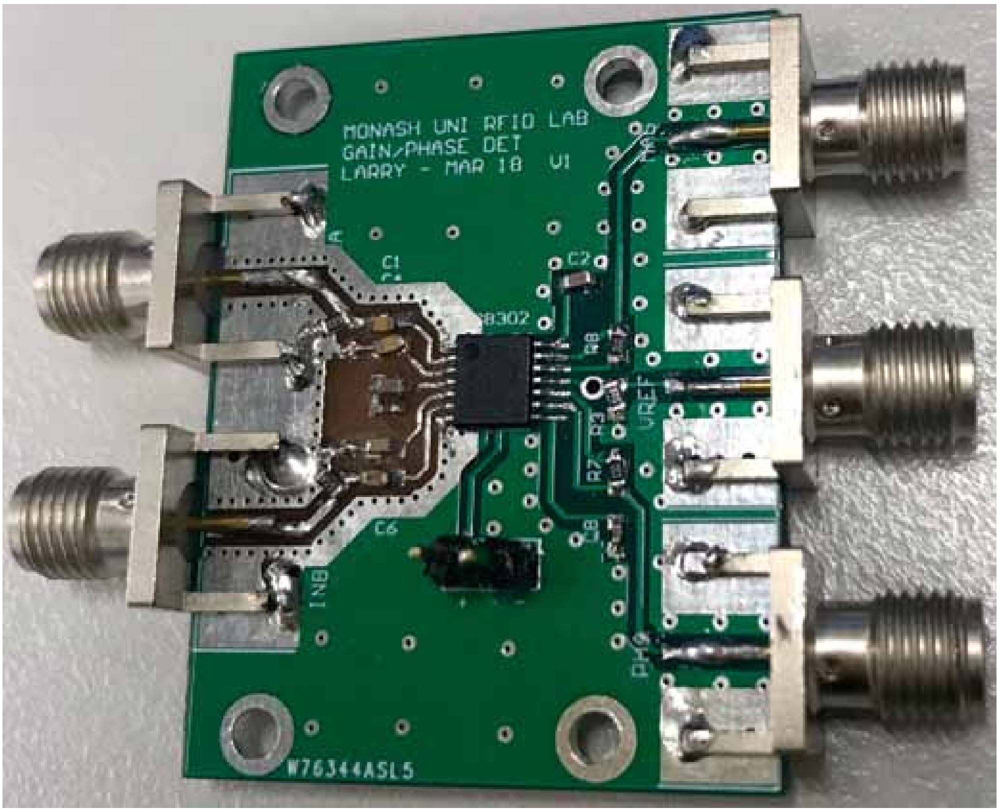


Figure 3.18 In-house developed AD8302 board (gain/phase comparator).

phase. The conversion rate of the used delta-sigma ADC depends on the resolution bits. With 12 bits resolution, the conversion rate is 192,000 SpS, and for 20 bits resolution, it goes down to 187 SpS [30].

Each PSoC5 pin can be assigned as a digital/analog input/output. As there is a capacitor (connected to the GND) for some pins, these pins should be avoided in higher-frequency digital communications.

The PSoC5 pin assignments for this book along with the other components are illustrated in Figure 3.21.

Figure 3.21 shows different PSoC5 pin allocations for controlling circuits. For example, to control the HMC6300 (60-GHz Tx) through the SPI interface, SEN_TX, SCK_TX, and SDI_TX pins are used as the output digital pins. The RST_TX pin resets the Tx when needed. Scan_TX is used as a bidirectional pin to verify that the register contexts are correctly loaded. The pin assignments are

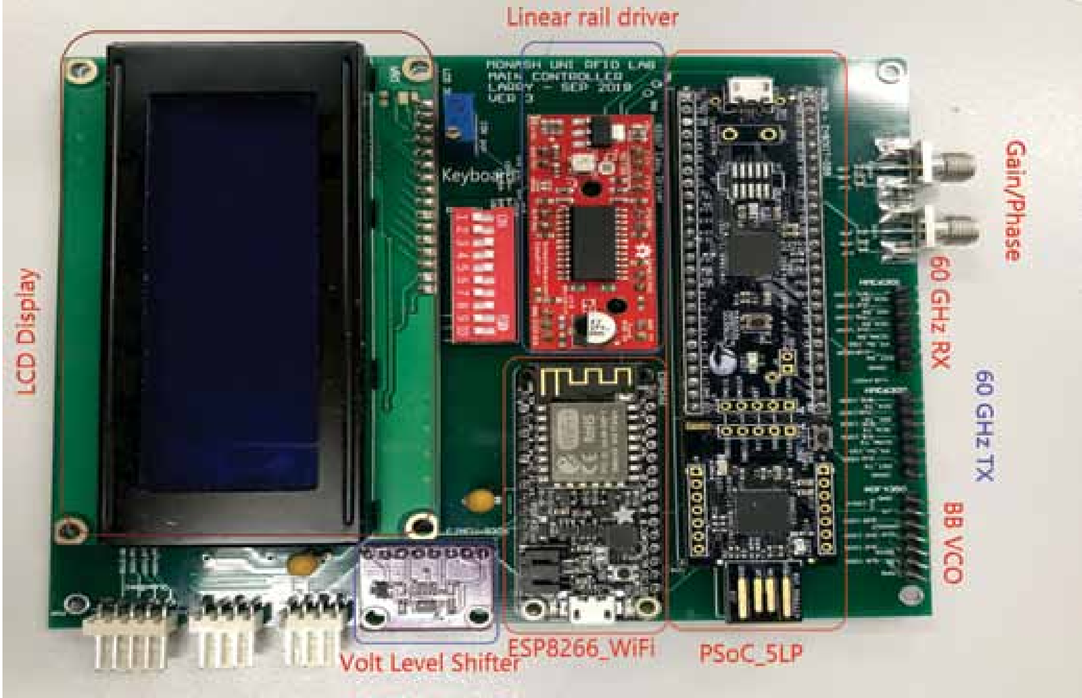
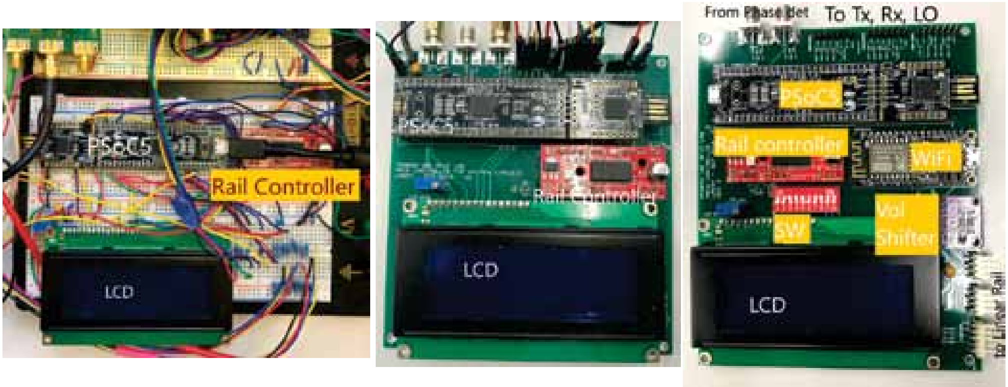


Figure 3.19 Latest version of the in-house developed digital control board with functional blocks identified [29].



(a) (b) (c)

Figure 3.20 Evolution version of the reader main digital boards. (a) The initial circuit on the breadboard, which contains a linear rail controller and an LCD to display data. (b) The first version on the PCB. (c) The latest version, with added functionalities such as a Wi-Fi module and DIP switches for data input. A level shifter circuit is used to match the logic voltage between the PSoC5 and Wi-Fi module (ESP8266).

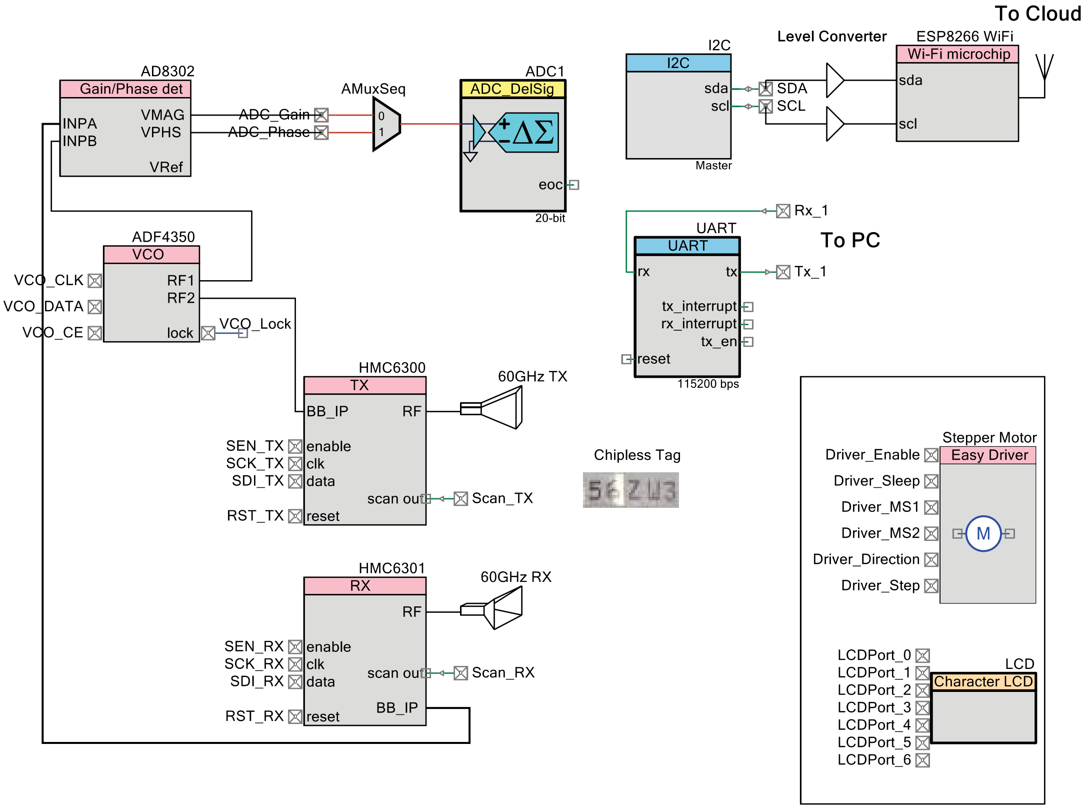


Figure 3.21 PSoC5 allocated pins for different components. These pins control different functions of Figure 3.6 using SPI, I2C, or direct commands (for LCD, linear rail controller, etc.) [29]. (Image courtesy of IEEE.)

tricky in PSoC5 because some pins are specialized for specific functions, like UART Rx_1 and TX_1 pins, and there are some lowpass filter capacitors connected to others. The complete list of pins and their assignments for this book is shown in Figure 3.22.

As mentioned, the HMC6350 Tx/Rx boards used require four quadrature inputs (baseband in-phase positive (BB-IP), baseband in-phase negative (BB-IM), baseband quadrature positive (BB-QP), and baseband quadrature negative (BB-QN)) for the lower phase noise outputs and higher transmission bit rates. Initially a test was carried out for the Tx/Rx using four inputs by a 5 GSpS arbitrary waveform generator. Afterward, a few attempts were made to produce those quadrature signals using four built-in PSoC5 DACs. Figure 3.23 clearly shows that the PSoC5 is not fast enough to

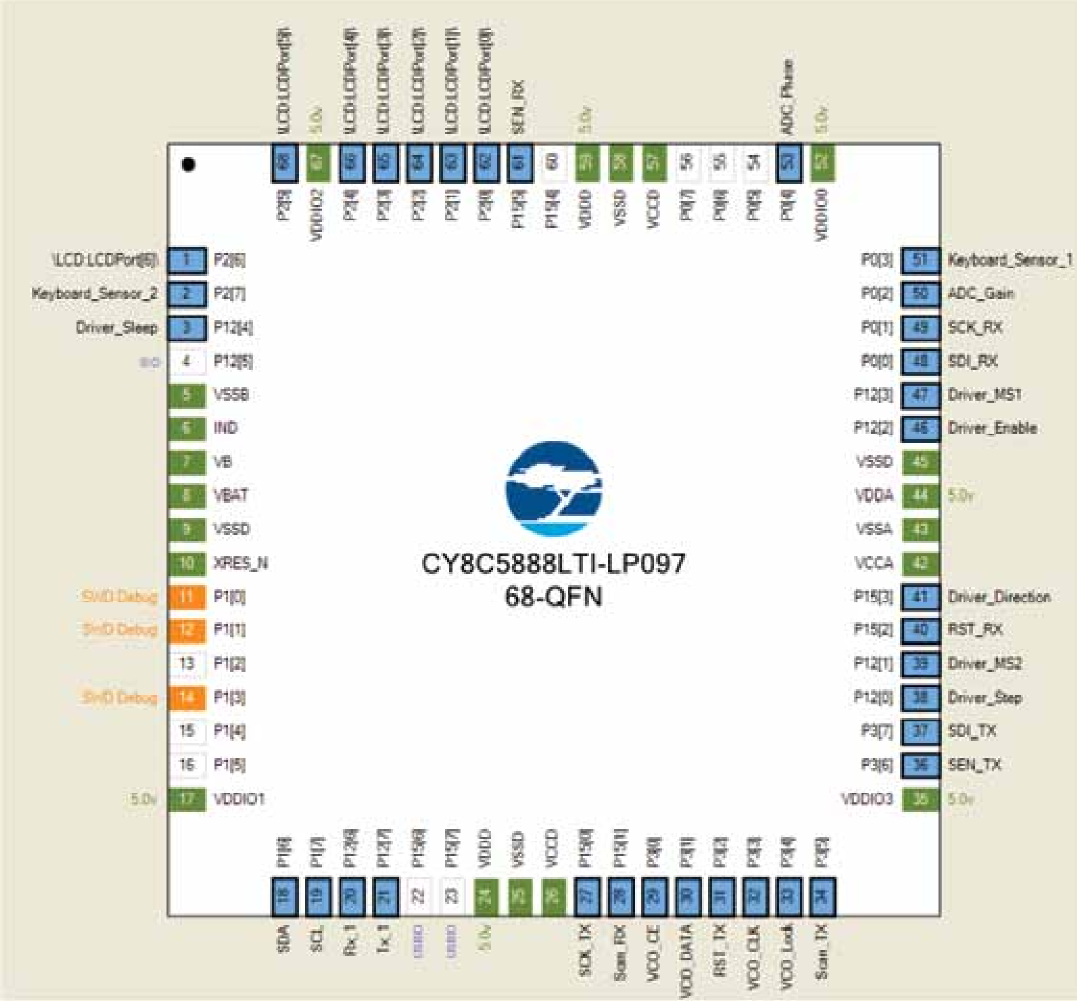
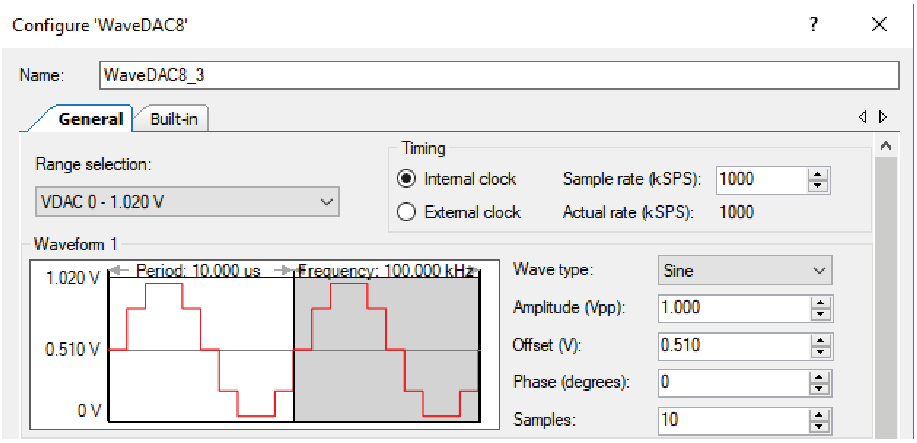


Figure 3.22 PSoC5 physical assigned pins. Careful pin assignment is needed to avoid using pins with lowpass filter (LPF) capacitors for the digital signals.

make the right sinusoidal signals faster than 10 kHz, even with its highest sample per second (1,000 kSpS).

Another PSoC5 limitation is that it cannot utilize more than one wave digital-to-analog converter 8-bit (WaveDAC8) component in the higher frequencies. The WaveDAC8 component enables waveform generation using sine/square/triangle/sawtooth/arbitrary waveforms and an 8-bit ADC. The dc-blocking capacitor in HMC6350 BB inputs board is 100 nF, with a 50Ω impedance at 31



Configure 'WaveDAC8'

Name: WaveDAC8_3

Range selection: VDAC 0 - 1.020 V

Timing

● Internal clock Sample rate (kSPS): 1000

○ External clock Actual rate (kSPS): 1000

Waveform 1

Wave type: Sine

Amplitude (Vpp): 1.000

Offset (V): 0.510

Phase (degrees): 0

Samples: 10

Figure 3.23 PSoC5 WaveDAC function block cannot make sine waves fast enough, so an external LO is used to make the IF signals.

kHz. This was one of the reasons the external LO is used (explained in Section 3.5.2).

3.5.5 Peripheral Circuits

Below is a list and functionality of the peripheral circuits used in the digital control board.

- *Stepper motor controller.* For the SAR scanning mode, a linear rail is needed to maintain the tag/antenna relative move positions. An aluminum rail is used in this work with an Easy Stepper motor driver (A3967) to drive the rail. The linear rail with the stepper motor is shown in Figure 3.24, and the linear rail controller module is the red block in Figure 3.19. A translator inside the A3967 controls the input to the driver DACs and the direction of current flow. A low-to-high transition on the STEP input pin sequences the translator and advances the motor by one increment. The size of the step increments is determined by the state of inputs MS1 and MS2. It provides four speeds (full/half/quarter/eighth step), according to the manual [31]. Step configurations for MS1 and MS2 pins are shown in Table 3.5.
- *Stepper motor in the linear rail.* There are 200 steps per circle, or each one step of stepper motor is 1.8 degrees. The down-

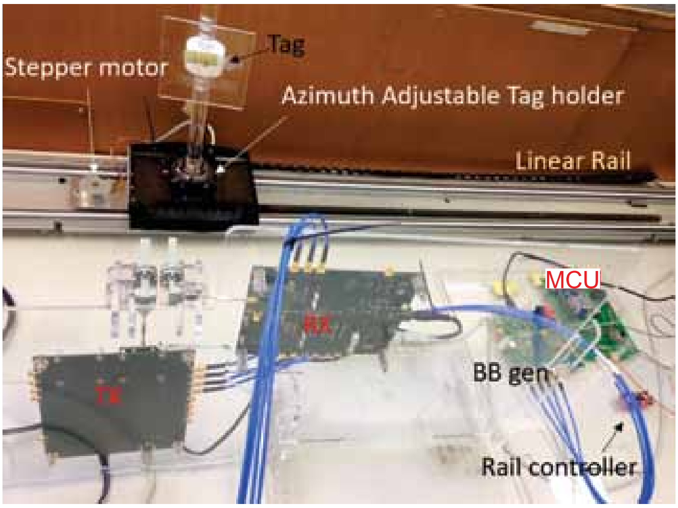


Figure 3.24 An early-stage hardware setup demonstration [29].

Table 3.5
A3967, Stepper Motor Controller Microstep Resolution Truth Table

MS1	MS2	Resolution
0	0	Full step
1	0	1/2 step
0	1	1/4 step
1	1	1/8 step

side for the stepper motor is that it does not have a positioning feedback. Stepper motors are continuous, which is not ideal for precision rail cases. Lack of position feedback can also cause big errors if the motor shaft is stuck. Most stepper motors have four controlling wires, and some have six. There is no polarity, so a simple ohmmeter can be used to find the connected poles. For the six-wire stepper motors, there are two separate coils, with three wires connected to each. Two wires are connected to the middle of each of these two coils, so all the wires can be easily identified with a resistance checking. Six-wire step motors are not very common. An early stage of the reader hardware configuration is shown in Figure 3.24, which shows the tags, 60-GHz reader, and linear rail.

- *LCD.* The LCD is the auxiliary output that shows the status of PSoC5 in the different stages for testing purposes. It is a 20-by-4 character display. To control the LCD, only 4 upper data bits (DB4-DB7) are used, together with E (Enable), RS (Register Select), and R/!W (Read/not Write) pins. The VEE is intended for backlight intensity control. The connects pins from the PsoC5 to the LCD are shown in Figure 3.21.
- *UART.* A UART with 115,200 bps is used to transfer the output of the ADC in different tag positions and different BB and carrier frequencies to the local computer. Although there are some hardware UART interfaces to send UART's serial data directly to the cloud, they are not recommended because of added cost and the size. In the next section we will go through how to use ESP8266 low-cost Arduino device and its Wi-Fi interface work for sending data directly to cloud servers using RESTful APIs.
- *Keyboard.* A 10-key dual in-line package (DIP)-switch is used as auxiliary inputs for defining tagIDs to the system (as in Figure 3.19). To reduce the number of PsoC5 pins as the input, a configuration similar to Figure 3.25 is used for every five keys. In this figure, one PsoC5 pin digitizes the analog input of "To ADC" for every five DIP keys, so just two pins are used in PsoC5 for 10 inputs. Pins P0[3] and P2[7] in

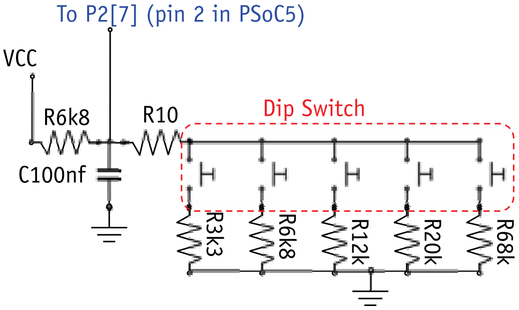


Figure 3.25 Keyboard layout. The output is connected to pin 2, which is configured as an ADC input in PSoC5 (port 2, pin 7 in Figure 3.22).

PsoC5 are used as the ADC inputs for this resistor ladder, as per Figure 3.22. A lookup table in the PsoC5 decodes the key sequences after the ADC digitization, a concept similar to R-2R ladders.

- *ESP8266 Wi-Fi module.* As said, PsoC5 does not have any Wi-Fi or local area network (LAN) modules, so a complementary circuit is needed for cloud connection and data exchanges. The ESP8266 module enables other microcontrollers to connect to 2.4-GHz Wi-Fi, using IEEE 802.11 bgn standards (up to 300 Mbps of network bandwidth). There are many other ways to connect PsoC5 to the cloud. Some of the other alternatives to ESP8266 as a Wi-Fi module can be a Wi-Fi-enabled Raspberry Pi (Rpi) device (such as Pico W or Pi 4 B), or with the Psoc6 with CYW4343W (Wi-Fi and Bluetooth module). We do not go into the details of ESP8266 at this stage other than to mention that it can be used as the slave in I2C communication with PsoC5 to get digitized amplitude and phase of the reflected signal from the tag in each frequency, and then send the data to the cloud server using a HTTP GET/POST in a RESTful API configuration with the cloud http web server.

In this book we did not use the capabilities of ESP8266, as data was collected using UART (Figure 3.6) to the local PC and then presented to the cloud using Google Drive. We will go into more details of data processing in the following chapters.

3.6 Reader Characterization

In this section some aspects of the developed reader characteristics are explained, such as scanning time, frequency response, and RCS calibration.

3.6.1 Scanning Time and Frequency Resolution Calculations

One important point in RFID system design is the time required to scan a tag in the frequency band of interest. The major bottleneck in the system is usually the ADC acquisition time.

There are two types of ADCs available in PSoC5, namely, successive approximation register and delta-sigma. The successive approximation register analog-to-digital converter (SAR ADC) has either 8, 10, or 12 bits and delta-sigma ADC has a maximum of 20 bits. Their difference in the operation is that SAR ADC uses sample-and-hold techniques for successive changes using a switch and a charging/discharging capacitor, whereas delta-sigma averages a series of binary outputs from a D-flip flop, which detects the difference between the output stage and input and an integrator that sums the analog input with a comparator in output. Because of its processing issues, delta-sigma is normally slower than SAR ADC, but it can have more accuracy.

Choosing between these two ADCs for any design comes down to which factor is more critical, reading speed or scanning accuracy. In this work, the voltage variations from gain/phase detector (AD8302) is in the mV range, and considering the ADC input range (–1.024 to +1.024V), at least 1,024 × 2 levels are needed, which corresponds to an 11-bit ADC. To provide extra allowance for safe operation, a 12 bit one may be chosen. So, any of the ADCs available are safe to use for a minimum resolution. In practice, however, the higher bits are more useful since noise averaging can be performed.

Assuming 12-bit sampling for ADC according to the PSoC creator software, the sampling rate will be 187,500 SpS maximum, or about 50 μs. Figure 3.26 magnifies the return path of the received backscatter signal after the RF stage of Figure 3.3.

According to Figure 3.26, for measuring each frequency, the LO should set up the frequency first, which takes 300 μs [32]. Receiving the backscatter is almost immediate, and it takes 60 ns for gain and 40 ns for phase to be processed in the AD8032 (although they might be done in parallel). Therefore, considering a safe delay

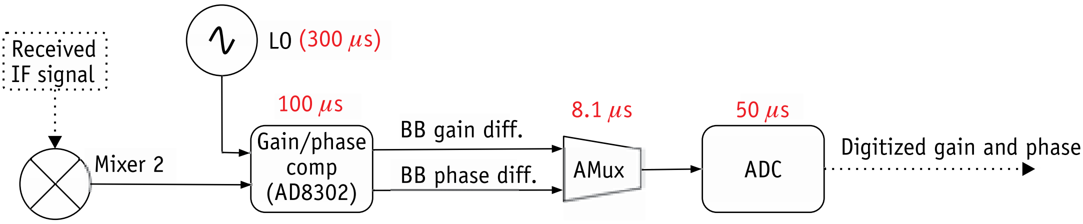


Figure 3.26 Time delay calculation for received signal in each component.

of 100 ns for AD8032 is logical [28]. The ADC needs 50 μs for each signal (gain and phase) [33], and so the total required time for gain and phase conversion is

$$300+100+(8.1+50)\times 2=516\ \mu s \tag{3.7}$$

With the resolution of 100 MHz in each 1.5-GHz RF step, there are 15 points to be sampled, so total time to scan a 1.5-GHz frequency span with 100-MHz resolution is

$$1.5\text{ GHz span}=15\times 516\ \mu s=7.74\text{ ms} \tag{3.8}$$

As discussed in Section 3.5.1.3, there is no information about the settling time of the HMC6300/HMC6301 as per their catalogs [23, 24]. From the experiments, a settling time of 10s is needed for these HMC6300/HMC6301 boards once the RF frequency changes. Having five 1.5-GHz steps in a 7-GHz frequency range of 57–64 GHz, the total scanning time for every tag in each position is around 50 seconds.

3.6.2 RCS Calibrations

The RCS is a property of the target's reflectivity, meaning that the level of transmitted power and the distance do not affect the measured RCS. In practical measurements, the reader power level normally needs calibration through the RCS calibration process. It is usually done using a standard object with a known RCS. In extended research at Palo Alto Research Center (PARC), a standard triangular trihedral corner reflector has been used whose RCS is 1 dB square meters (dBsm) at a distance of 1m at 60 GHz. Figure 3.27 illustrates the triangular trihedral shape. The scattering calculations for the trihedral objects can be found in [34, 35].

The RCS of the triangular trihedral reflector of Figure 3.27 can be calculated as [36]

$$\sigma=\frac{4\pi a^4}{3\lambda^2} \tag{3.9}$$

where a is the length of one of the sides, as shown in Figure 3.27. If $a = 5$ cm in (3.9), the RCS will be 1 at 60 GHz as $\lambda = 5$ mm. Knowing

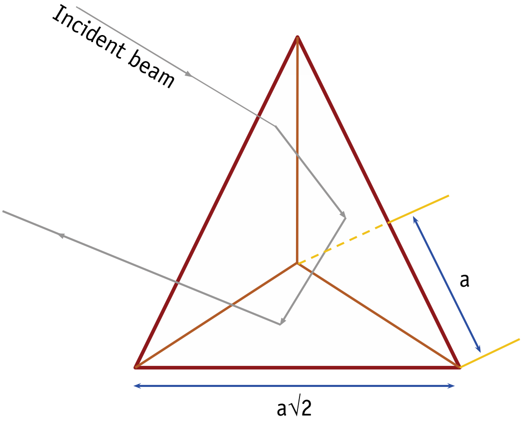


Figure 3.27 An illustration of a triangular trihedral. The measured RCS at a certain distance and frequency is a known value.

the RCS at different frequencies, the value of $K(\theta_t, \phi_t, \theta_r, \phi_r)$ in (2.14) can be calculated using the calibration outcomes. The calibration procedure is as follows:

- In the lab or an anechoic chamber, measure S_{11} for the case of monostatic configuration by (2.16), or S_{11} and S_{22} for a bi-static case by (2.14) just for the structure of the jig without a standard trihedral. The distance of the antenna to the jig that holds the tag should be 1m. Write down the Tx and Rx powers.
- Repeat the above step in the presence of a standard trihedral.
- Using (2.16) for monostatic case and (2.14) for a bistatic case, calculate K (at each frequency). σ is known as per (3.9).

Considering the graph with a known RCS value, the reader can be calibrated in a similar way. Figure 3.28 shows the measurement done with a standard trihedral at a distance of 1m from the antennas. Using (2.16), and knowing the P_t (transmitted power), P_r (received power), G (antenna gain from the catalog), S_11, and R (distance from antenna to the trihedral), the calculated RCS is shown in Figure 3.28. From (3.9), the RCS should be 1 at 60 GHz, whereas it is –6.73 dBsm. Therefore, +6.73 should be added to the measured RCS at 60 GHz for it to be corrected. Doing this for

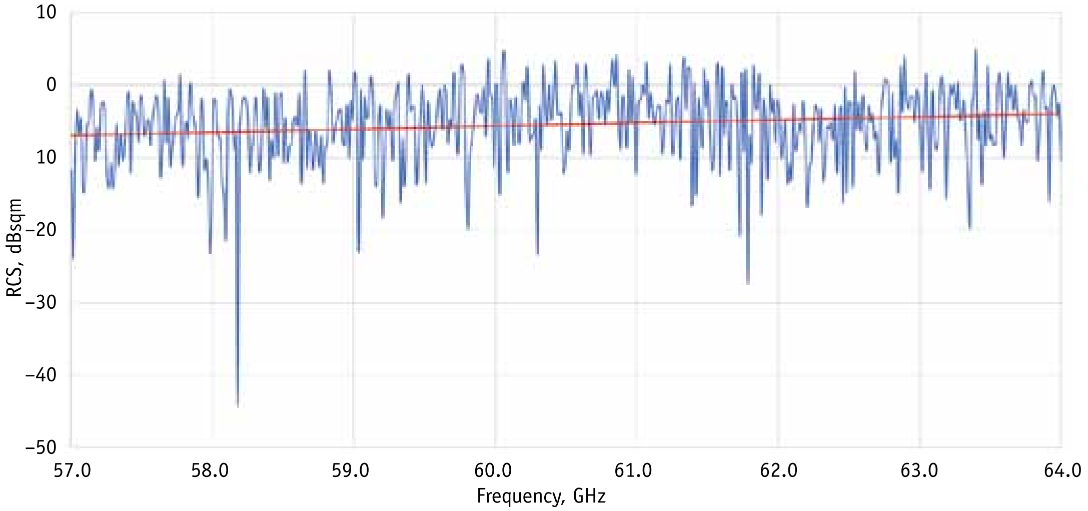


Figure 3.28 Modified RCS for the triangular trihedral corner reflector (blue) and its linear trend (red line). These values can be used for the calibration of the reader at different frequencies.

all of the frequency spectrum, the chipless reader values can be calibrated.

If the signal is considered to be narrowband, the average RCS trend, shown by the red line in Figure 3.28, can be used as the first estimation for the reader calibration. The bandwidth of 57–64 GHz is almost narrowband as

$$\frac{\Delta f}{freq\_center} = \frac{7}{60.5} \approx 12\% \tag{3.10}$$

3.7 Conclusions

A 60-GHz ISM band chipless RFID has been a research focus in recent years [37–40], but there was no RFID reader hardware developed for this spectrum. The problem with this higher spectrum is its much higher attenuation at the same distance for the received backscattered signals compared to lower microwave and mm-wave frequencies.

After a few design trials, a modular design around HMC6350 Tx/Rx modules was provided and developed. The HMC6350 initial intention design was for backhaul signal transmission, so many parameters had to be considered in order to integrate these boards into a chipless RFID system. In this chapter, those problems, such as USB port conflict, customized SPI signaling control, and low-resolution frequency scanning, were discussed, and proper solutions were provided.

Running the HMC6350 boards required a complete set of RF and digital controllers and peripherals. For the higher-frequency resolutions and to avoid timely mm-wave step changes in the HMC6350, a LO was developed to provide high-resolution frequency sweeps from 500 MHz to 2 GHz in the desired steps. A gain/phase comparator circuit was developed to compare sent and received signals, and a high-precision delta-sigma ADC was configured to digitize the output of this comparator.

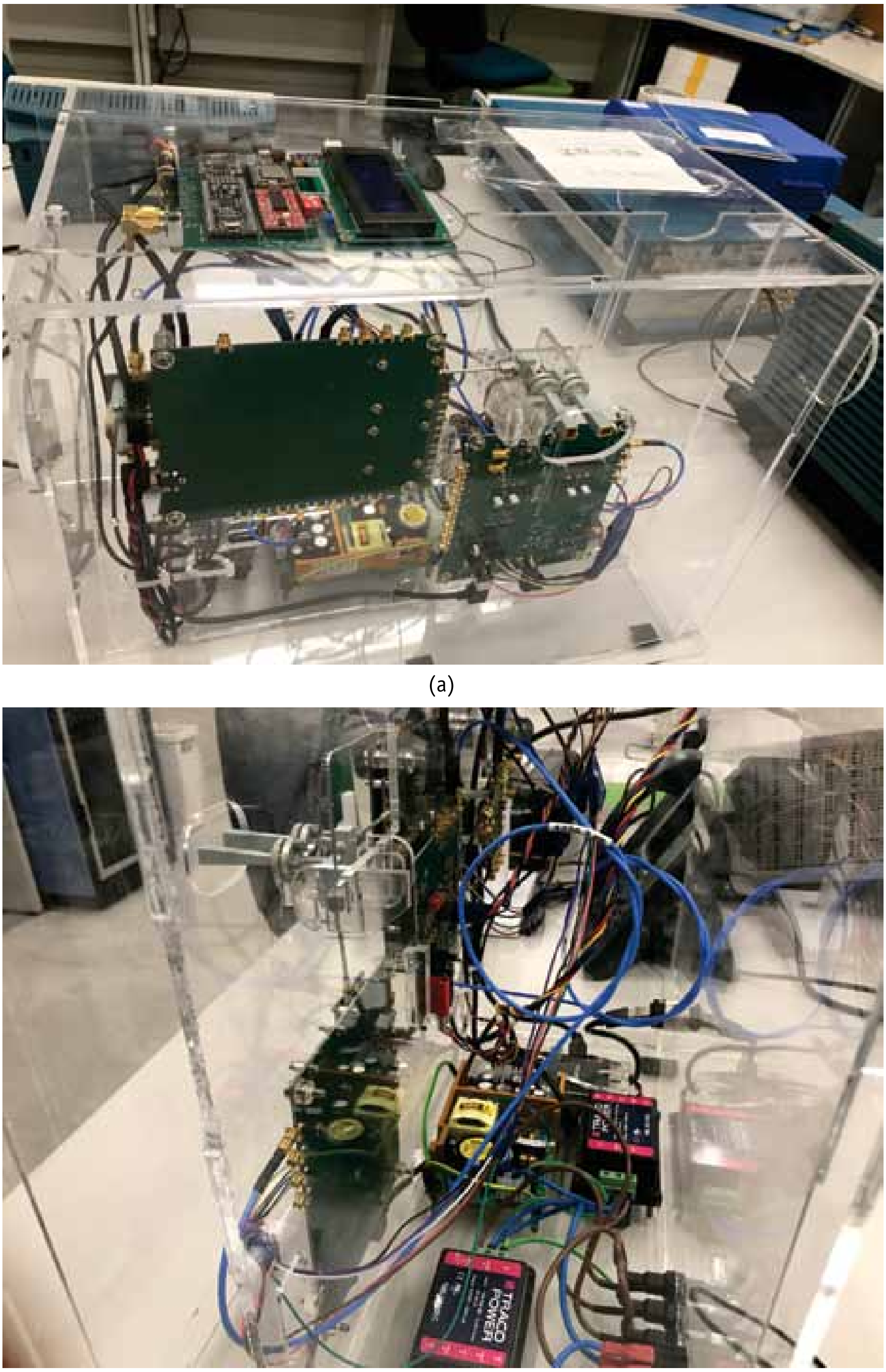
As the reader system is based on the SAR concept, a linear rail and its controller circuit was inserted for precise movements of the tags in front of the reader antennas. Compact 10-bit DIP switches were used as an auxiliary input when some interaction from the user was needed. A special Wi-Fi module was used to transfer data to the local host or remote (cloud) computer for further processing of the digitized backscattered signals from the tag.

This chipless RFID hardware made it possible for the first time to read chipless tags without any need for lab measurement devices at the higher 60-GHz spectrum. This is the first developed 60-GHz chipless RFID tag system, according to the open literature and to the best knowledge of the authors. Figure 3.29 shows the final reader from different angles.

3.8 Chipless Reader Questions and Answers

3.1 Why use a reader at 60 GHz and not lower frequency bands?

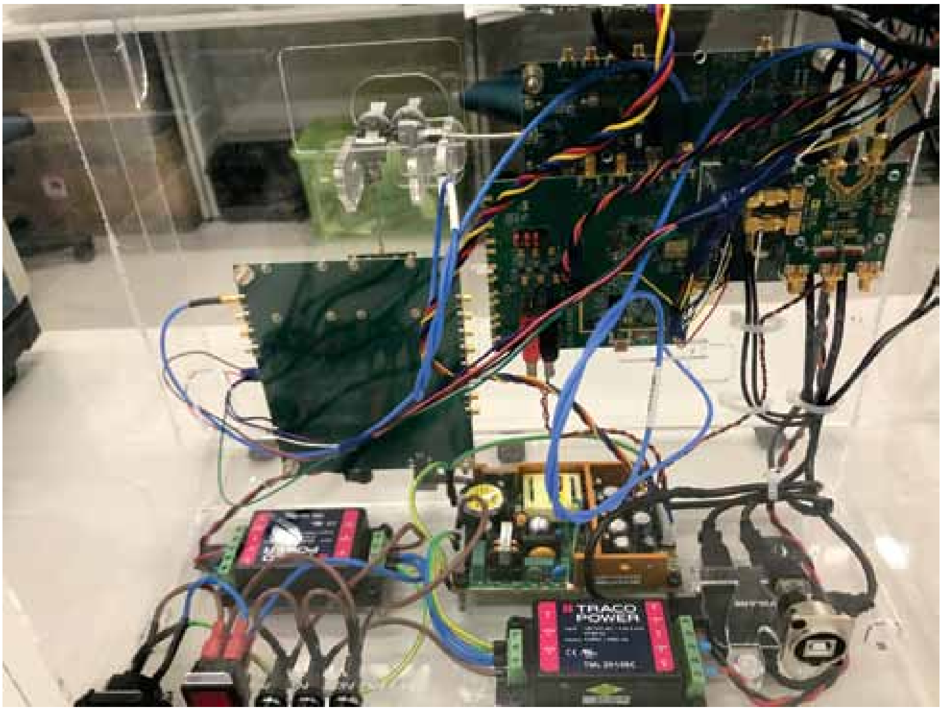
The next available spectrum in the ISM band for radio location applications like RFID after 24 GHz is 61 GHz [41]. Also, tags, readers, and antennas will be smaller in 60 GHz as the frequency goes higher. The next available ISM band after 60 GHz for RFID applications is 244–246 GHz, which is just out of reach for the current technology.



(a)

(b)

Figure 3.29 The developed 60-GHz RFID reader from different angles.



(c)

Figure 3.29 (continued)

3.2 Why not use the MIMO reader developed here?

The simple answer is the market availability at the start time of this project. Currently there are many MIMO based industrial mm-wave radar sensors to choose from. The modular design in this chapter can be used as a sample/guideline for frequency and image-based tag readers no matter what RF boards are used.

3.3 Why did we use a VM in this chapter?

The Tx and Rx boards of the HMC6350 use the same port number. A VM is used within the local PC to be able to capture one of these boards with a different port number. The VM is not used after the initial testing phase.

3.4 Why do we need to perform a chipless reader calibration?

It is important to do calibration for your developed reader, especially if the results come from more than one device;

for example, if different tags are read by the VNA or by different readers. A calibration process ensures that the results will be similar no matter what reader is used. The reader device resolution might be higher or lower than the VNA in use, but it is important that the measured points for the reader have the same magnitude/phase as the VNA. The VNA itself needs yearly calibration by accredited companies. These accredited companies must be certified themselves by a national authority, such as the National Association of Testing Authorities (NATA) in Australia.

3.5 What are the recommendations for a linear rail for a chipless tag reader?

The best linear rail should have as many nonmetal parts as possible, as any metal can interfere with the chipless tag detection. In addition, the stepper motor in a linear rail is a good source of audible and electrical noise. Using the MIMO reader will eliminate the need for a linear rail, but it will add to the computing costs.

3.6 Are there any technical considerations for the developed circuit?

As shown in Figure 3.29, three separate power supplies are used to empower digital/analog boards Tx and Rx. This is to minimize coupling noise between the circuits.

3.7 Any recommendations for the updated boards?

With the new technology, we recommend using a RPi pico board with Wi-Fi as a replacement for the local computer and Wi-Fi device instead of the Arduino ESP8266 in Figure 3.6. The small version of Linux (Fuzix) on RPi can help to run a web-client for REST API communications to a cloud computer in a platform-as-a-service (PaaS) configuration, or to access Google Drive files and Google Colab for software-as-a-service (SaaS) configuration (more details in Chapter 5).

3.8 Why is a PSoC5 used instead of an Arduino-based microcontroller unit (MCU)?

The PSoC5 has many components and more input/output pins than any Arduino-based MCU. An Arduino is an option if multiplexing is used to increase its port numbers and components.

3.9 Any useful tips for PSoC5 application development?

As a general engineering rule, no circuit will work right after the design phase. To make it work, each circuit needs extensive debugging. One very useful tip is to use the PSoC5 printout method using UART component and terminal emulators (like Termite). We recommend watching tutorials from our former lab colleague, J. Lee, on his YouTube channel [42, 43].

References

[1] Preradovic, S., I. Balbin, N. C. Karmakar, and G. F. Swiegers, "Multiresonator-Based Chipless RFID System for Low-Cost Item Tracking," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 57, No. 5, 2009, pp. 1411–1419.

[2] Koswatta, R. V., "Readers for Frequency Signature-Based Chipless RFID Tags," PhD dissertation, Monash University, 2013.

[3] Karmakar, N. C., P. Kalansuriya, R. E. Azim, and R. Koswatta, *Chipless Radio Frequency Identification Reader Signal Processing,* John Wiley & Sons, 2016, doi:10.1002/9781119215783.

[4] Forouzandeh, M., and N. C. Karmakar, "Chipless RFID Tags and Sensors: A Review on Time-Domain Techniques," *Wireless Power Transfer,* Vol. 2, No. Special Issue 02, 2015, pp. 62–77, doi:10.1017/wpt.2015.10.

[5] Sroka, J., "Oscilloscope Influence on the Calibration Uncertainty of the Pulse Rise Time of ESD Simulators," in *2003 IEEE International Symposium on Electromagnetic Compatibility, 2003. EMC '03.*, Vol. 1, May 2003, pp. 378–381, doi:10.1109/ICSMC2.2003.1428270.

[6] Paret, D., *RFID at Ultra and Super High Frequencies*, John Wiley & Sons, 2009, doi:10.1002/9780470682135.

[7] Bibile, M. A., and N. A. Karmakar, "Moving Chipless RFID Tag Detection Using Adaptive Wavelet Based Detection Algorithm," *IEEE Transactions on Antennas and Propagation,* Vol. 66, No. 6, 2018, 2752–2760, doi:10.1109/TAP.2018.2819725.

[8] Arjomandi, L. M., and N. C. Karmakar, *Improved Chipless RFID System and Method*, Australian Patent 2017902112, 2017.

[9] Forouzandeh, M., and N. Karmakar, "Self-Interference Cancelation in Frequency-Domain Chipless RFID Readers," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 67, No. 5, 2019, pp. 1994–2009.

[10] Yogev, A., and S. Dukler, *Radio Frequency Data Carrier and Method and System for Reading Data Stored in the Data Carrier,* US Patent 6,997,388, February 2006.

[11] InkSure Technologies, *Progress Report on Chipless RFID Applications SARcode Development by InkSure Technologies,* https://www.slideserve.com/addisonglenn/progress-report-on-chipless-rfid-applications-sarcodedevelopment-by-inksure-technologies.

[12] "RFID Tattoos to Make a Mark on Cattle Tagging," *RFID Journal,* 2019, https://www.rfidjournal.com/articles/view?6916.

[13] Vubiq, *Vubiq Networks' RFID Hyperimaging Patent Portfolio Executive Summary,* https://www.vubiqnetworks.com/wp-content/uploads/VubiqRFID-Patents-Exec-Summary-02JUL19.pdf.

[14] A.C.M.Authority,AustralianRadiofrequencySpectrumAllocatonsChart,2008, https://www.acma.gov.au/theacma/australian-radiofrequencyspectrum-plan-spectrum-planning-acma.

[15] U.S. Frequency Allocations, https://www.ntia.doc.gov/files/ntia/publications/january_2016_spectrum_wall_chart.pdf.

[16] Federal Communications Commission, Office of Engineering and Technology, "Guidelines for Determining the Effective Radiated Power (ERP) and Equivalent Isotropically Radiated Power (EIRP) of a RF Transmitting System," 2010, https://apps.fcc.gov/oetcf/kdb/forms/FTSSearchResultPage.cfm? id=47469&switch=P.

[17] Chapman, R. B. C., *Radiocommunications (Low Interference Potential Devices) Class Licence 2015*, September 2015, https://www.legislation.gov.au/Details/F2022C00281.

[18] Alwis, G. D., and M. Delahoy, *GHz Band Millimetre Wave Technology,* Australian Communications Authority (ACMA), December 2004.

[19] A-Info, *LB-15-10 50.0–75.0 GHz Standard Gain Horn Antenna,* http://www.ainfoinc.com.cn/en/pro_pdf/new_products/antenna/Standard%20Gain%20Horn%20Antenna/tr_LB-15-10.pdf.

[20] Balanis, C. A., *Antenna Theory: Analysis and Design*, Third Edition, Hoboken, NJ: John Wiley, 2005.

[21] Griffin, J. D., and G. D. Durgin, "Complete Link Budgets for Backscatter Radio and RFID Systems," *IEEE Antennas and Propagation Magazine*, Vol. 51, 2009, doi:10.1109/map.2009.5162013.

[22] Maini, A. K., *Handbook of Defence Electronics and Optronics: Fundamentals, Technologies and Systems,* John Wiley & Sons, 2018.

[23] Analog Devices, *60 GHz Millimeterwave Transmitter, 57 GHz to 64 GHz*, 2017, https://www.analog.com/media/en/technical-documentation/datasheets/HMC6300.pdf.

[24] Analog Devices, *60 GHz Millimeterwave Receiver, 57 GHz to 64 GHz*, 2017, https://www.analog.com/media/en/technical-documentation/datasheets/HMC6301.pdf.

[25] Analog Devices, *EK1HMC6350 User Guide*, 2017, https://www.analog.com/media/en/technical-documentation/user-guides/EK1HMC6350UG-1031.pdf.

[26] O. Corporation, *Oracle VM VirtualBox User Manual,* https://download.virtualbox.org/virtualbox/5.2.4/UserManual.pdf.

[27] Cypress Semiconductor Corporation, *CY8CKIT-059 PSoC 5LP Prototyping Kit Guide*, 2018, https://www.cypress.com/file/157971/download.

[28] Analog Devices, *Analog Devices AD8302, LF–2.7 GHz RF/IF Gain and Phase Detector,* https://www.analog.com/media/en/technical-documentation/datasheets/ad8302.pdf.

[29] Arjomandi, L. M., G. Khadka, Z. Xiong, and N. C. Karmakar, "Document Verification: A Cloud-Based Computing Pattern Recognition Approach to Chipless RFID," *IEEE Access*, Vol. 6, 2018, pp. 78007–78015.

[30] Cypress Semiconductor Corporation, *Delta Sigma Analog to Digital Converter (ADC$_D$elSig)*, 2020, https://www.cypress.com/documentation/componentdatasheets/delta-sigma-analog-digital-converter-adcdelsig.

[31] Allegro Microsystems Inc., *A3967 Micostepping Driver with Translator*, 2017, https://www.sparkfun.com/datasheets/Robotics/A3967.pdf.

[32] Analog Devices, *ADF4350 Wideband Synthesizer with Integrated VCO*, 2017, https://www.analog.com/media/en/technical-documentation/datasheets/ADF4350.pdf.

[33] *Cypress, Analog Multiplexer (AMux),* https://www.cypress.com/documentation/component-datasheets/analog-multiplexer-amux.

[34] Gau, J.- R., and W. D. Burnside, "A Full RCS Calibration Technique Using a Dihedral Corner Reflector," in *IEEE Antennas and Propagation Society International Symposium 1992 Digest*, 1992, pp. 557–560.

[35] Chen, T.- J., T.- H. Chu, and F.- C. Chen, "A New Calibration Algorithm of Wide-Band Polarimetric Measurement System," *IEEE Transactions on Antennas and Propagation*, Vol. 39, No. 8, 1991, pp. 1188–1192.

[36] Garthwaite, M. C., M. Thankappan, M. L. Williams, S. Nancarrow, A. Hislop, and J. Dawson, "Corner Reflectors for the Australian Geophysical Observing System and Support for Calibration of Satellite-Borne Synthetic Aperture Radars," in *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, IEEE, 2013, pp. 266–269.

[37] Karmakar, N. C., M. Zomorrodi, and C. Divarathne, *Advanced Chipless RFID: MIMO-Based Imaging at 60 GHz–ML Detection*, Hoboken, NJ: John Wiley & Sons, 2016.

[38] Attaran, A., R. Rashidzadeh, and R. Muscedere, "Rotman Lens Combined with Wide Bandwidth Antenna Array for 60 GHz RFID Applications," *International Journal of Microwave and Wireless Technologies*, Vol. 9, No. 1, 2017, pp. 219–225.

[39] Necibi, O., A. Ferchichi, T.- P. Vuong, and A. Gharsallah, "A Discussion of a 60GHz Meander Slot Antenna for an RFID TAG with Lumped Element," *International Journal of Wireless and Microwave Technologies (IJWMT)*, Vol. 4, No. 2, 2014, pp. 1–11.

[40] Bisognin, A., J. Thielleux, W. Wei, et al., "Inkjet Coplanar Square Monopole on Flexible Substrate for 60-GHz Applications," *IEEE Antennas and Wireless Propagation Letters*, Vol. 13, 2014, pp. 435–438.

[41] Wikipedia, "ISM Radio Band," August 2022, https://en.wikipedia.org/wiki/ ISM_radio_band.

[42] Lee, J., *PSoC Tutorial–Printing to Termite with UART*, July 2017, https://www.youtube.com/watch?v=lU3RGuzuqdE&ab_channel=JonathanLee.

[43] Lee, J., *PSoC Tutorial–Printing Floats to Termite Solution*, July 2017, https://www.youtube.com/watch?v=jXTAfFSo9v8&ab_channel=JonathanLee.

4

Tag Decoding

4.1 Introduction

A chipless tag is just a piece of metal with no intelligence, so it is up to the reader to be smart enough to detect the tag and decode the tag's encoded data. After describing tag design in Chapter 2 and reader hardware in Chapter 3, this chapter deals with these smart processing methods.

Concerns arise when dealing with chipless tag decoding. First, as the distance or angle of reading from the reader toward the tag changes, the frequency response changes both in the amplitude and in phase. This is because the contact area for an interrogated signal from the transmitter changes, so the surface current will change and thus the resonant frequency and amplitude will vary. Second, moving the tag's angle and distance toward the reader changes the background objects slightly, thus changing the total response of the tag. Third, chipless tags are always subject to some manufacturing tolerances. Although two tags may seem similar, their edge sharpness may vary because of changed resolution or the metal thickness may be different because of using different inks or even using different substrates. This chapter tries to resolve these nonlinear responses of the chipless tag to some extent by making tag SLAR images, and decoding using machine learning techniques.

This chapter also tries to answer one of the main problems in a chipless RFID, which is misdecoding or a reading error. Sometimes the output of the decoder software is the wrong tagID. As the chipless tags have no handshaking means, this is quite a dangerous situation. Using novel SLAR imaging, a trial around this is created. This image-based tag detection method is used in parallel with the AI methods to provide a more reliable system, which will be discussed in detail in Chapter 5.

The position of this chapter in this work is shown in Figure 4.1 by the ML block.

The chapter structure is as follows: After this introduction, in Section 4.2 the concept of a feedforward network and several pattern recognizers are presented. In Section 4.3, methods of data collection in simulations and experiments are discussed. The ML problem is defined with the proposed solution, and some aspects of the performance, such as decoding success rate and orientation sensitivity analysis, are discussed in Section 4.4. Various methods of pattern recognition are presented and compared in Section 4.5. Because of the abovementioned misdetection problem, an imaging method based on SLAR is proposed in Section 4.6. The section includes mathematical calculations to create the final image using one- or two-port (monostatic or biostatic) reader systems along with the computational costs. The chapter finishes by summing up the methods discussed. This chapter and the next one comprise the brain part of the proposed chipless RFID reader. Figure 4.2 illustrates the material covered by this chapter.

This book is based on ML, which requires a large amount of storage, high demand for processing, and data sharing between different readers. Cloud computing is used here to facilitate these requirements, as will be explained further in Chapter 5.

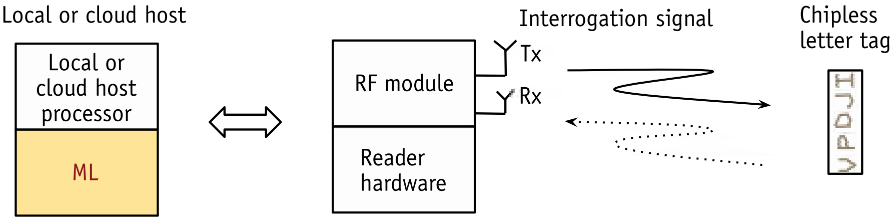


Figure 4.1 Illustration of the machine learning section within the RFID system.

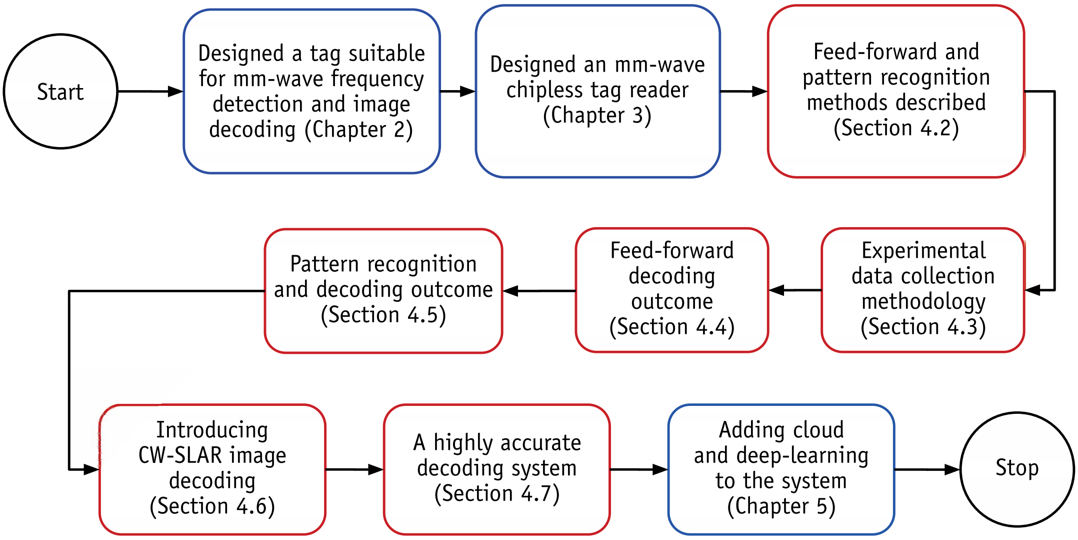

Start
Designed a tag suitable for mm-wave frequency detection and image decoding (Chapter 2)
Designed an mm-wave chipless tag reader (Chapter 3)
Feed-forward and pattern recognition methods described (Section 4.2)
Experimental data collection methodology (Section 4.3)
Feed-forward decoding outcome (Section 4.4)
Pattern recognition and decoding outcome (Section 4.5)
Introducing CW-SLAR image decoding (Section 4.6)
A highly accurate decoding system (Section 4.7)
Adding cloud and deep-learning to the system (Chapter 5)
Stop

Figure 4.2 An overview of the different sections in this chapter. Topics in the red blocks will be covered here.

4.2 Machine Learning and Pattern Recognition

With recent advances in AI, machine learning can now be used as a valuable decoding tool in RFID and IoT. Conventionally, a computer is programmed to do a particular task. In contrast, with machine learning the computer learns how to do a difficult task by observing data. The way the computer breaks down the problem into smaller blocks and solves them is generally uninterpretable.

Every machine learning problem can be described in a few steps [1]. Figure 4.3 shows the steps in a typical machine learning method. After defining the machine learning problem, a dataset is produced. In the simulations reported in this book, a CST program has been used to produce this data, and in the experiments, a VNA and the tag reader developed in Chapter 3 has been used for creating the dataset. The dataset generated in the simulation or experimental methods is firstly shuffled and then used to train the model until an acceptable mean square error (MSE) is reached in the training. The last step is to use this trained network to make predictions, which is the chipless tag decoding in this work.

In every neural network, there is a web-connected set of nodes called neurons and links between them. A neural network's main function is to receive inputs, do complex calculations, and put it

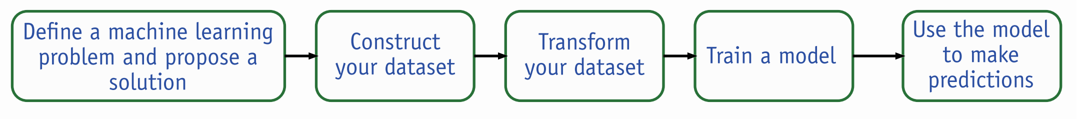

Define a machine learning problem and propose a solution
Construct your dataset
Transform your dataset
Train a model
Use the model to make predictions

Figure 4.3 Machine learning steps.

to output as a solved problem. They can be used for many applications. For the classification problem, for example, temperature, height, and heartbeat rate can be the inputs, and the output level shows how likely a patient is sick.

Every network has input, output, and hidden layers. Each layer gets the input from the previous layer, processes it with its activation function, and passes the result to the next level for further processing. The way this type of neural network propagates data from input layers to output is called feedforward propagation.

Each edge (link) has a weight and each node has a bias. This means that the combination for each activation is also unique. The process of increasing the neural network (NN) output accuracy to the actual desired output from a provided set of input-output data is called training. Training is done with a comparison of current NN output to the actual desired output. The aim is to minimize the cost function of difference of these two. Figure 4.4 illustrates the biological neuron model of a nerve cell, with simple mapping of inputs and outputs (cell's synapses). Millions of these nerve cells connect to each other to transfer an impulsive message from the body parts to the brain.

4.2.1 Tag Decoding Using Feedforward Networks and Backpropagation

Backpropagation (also known as the generalized delta rule) is one of the best-known methods for training an artificial neural network (ANN) [3]. The most common NN is composed of multiple layers with a connection between the nodes (neurons) in neighboring layers only, with no interconnection between nodes in the same layer. Information flows in one direction, from the input layer to the output layer. This is also called a feedforward network. Initially, a feedforward network is used in the book to check the feasibility of using AI in chipless RFID decoding.

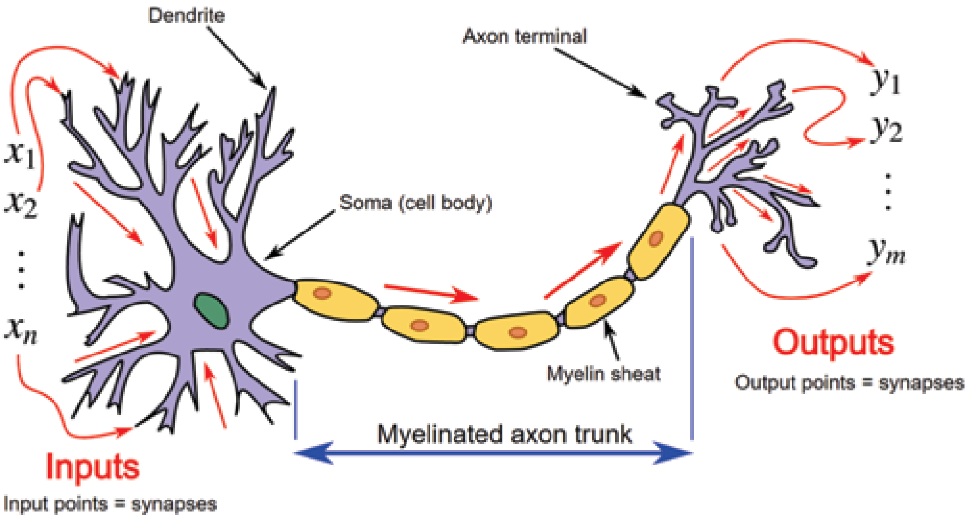


Figure 4.4 A biological neuron model illustrated beautifully, showing input points, cell body, and the outputs [2]. Some of the inputs or outputs connect with weaker or stronger connections to the other neurons to form a neural network that is capable of learning and predictions.

4.2.2 Feedforward Concept

A feedforward network is composed of a few layers. The computation in each layer (except the input layer) is based on a weighted summation of all connections from the previous layer by the node function plus a bias. This computation, also called collected function, can be of the form

$$net_{pi} = \sum_i w_{ij} a_{pj} + bias_i \tag{4.1}$$

where w_{ij} is the weight from current node i to the previous node j, a_{pj} is the activation function of the node j for the input pattern p, and $bias_i$ is the bias at the node i. An activation function can be a nonlinear one, normally a sigmoid in the following form:

$$a_{pi} = \frac{1}{1+e^{-net_{pi}}} \tag{4.2}$$

For training, this type of network is supervised. Supervised learning uses collected input/output sets from the previous experiments and calculates the weights and biases to obtain the same results. Some sets of input-outputs are applied and the w_{ij} connections are adjusted to minimize the error function given by

$$E = \sum_{p} \sum_{i} \left(t_{pi} - O_{pi}\right)^{2} \tag{4.3}$$

where t_{pi} and O_{pi} are the desired and actual values of the pattern p in the output node i. In this way, the ANN learns the input-output relations. Using the gradient descent method, the iterative weight up ε date rule will become

$$\Delta w_{ij}\left(n+1\right) = \varepsilon \gamma_{pi} a_{pi} + \mu \Delta w_{ij}\left(n\right) \tag{4.4}$$

where ε is the learning rate parameter, μ is the momentum constant, and γ_{pi} is the node i error for the pattern p, which can be written in the form

$$\gamma_{pi} = a_{pi}\left(1 - a_{pi}\right) \sum_{k} \gamma_{pk} w_{ki} \tag{4.5}$$

where the summation collects the errors from the previous layer and the other terms are derivative of the activation function [4].

Backpropagation in neural network training is used with the gradient descent optimization method. It calculates the gradient of a loss function compared to all available weights in the network. When this gradient is used in an optimization problem for updating the neuron weights, the loss function will be minimized. The cost value, which is the difference between the current NN output and the desired output, is lowered by changing weights and biases until a minimum is reached [3].

Verifying through a few networks, Neilson [5] showed that neurons in the first layers of a network learn at a much slower rate than deeper layers. With lower learning speeds in earlier hidden layers, this phenomenon means that the first hidden layers simply throw away useful input information, making it harder for later layers to construct the desired output. The learning gradient is higher in the last layers and goes down in the first layers. The gradient at each point is the multiplication of gradients at former layers (from output to input, as this is backpropagation).

In addition, it takes much more time to train the first layers compared to the last layers. Earlier layers are in charge of detecting simple patterns and building blocks, but if they are not trained correctly, the whole result would be wrong. The smaller learning gradient toward the first hidden layers is called the vanishing gradient problem. The instability in the learning gradient problem is unfortunately unavoidable in feedforward networks. The learning gradient sometimes vanishes and sometimes explodes in earlier layers. The vanishing (or exploding) gradient is the cause of very long training time and very low overall accuracy. Before 2006, there was no way to train deeper networks due to this vanishing or exploding gradient problem.

It can be proven that finite superpositions of a fixed, single-variable function that is discriminatory can uniformly approximate any continuous function of any real variables [6]. Normally sigmoid functions are used for this discriminatory rule. The choice of the activation function is not by itself an influencing factor for universal approximation. The feedforward architecture is the one that can potentially approximate any function [7]. However, just because of the abovementioned proven universality theory, it should not be assumed that the feedforward architecture can essentially be used in all sorts of problems.

Feedforward networks can be used for some applications if they are not stuck with a vanishing gradient. Catal et al. [8] used multilayer perceptrons (MLPs) in an ensemble of other networks for classification of human body activities. Wang et al. [9] used an improved particle swarm optimization to enhance the connection weights on the feedforward network for a location-based service application. Selvan et al. [10] used feedforward networks for designing a spiral strip monopole antenna fed by a coplanar waveguide (CPW) for RFID applications. Kuo et al. [11] used a combination of an optimization artificial immune network (Opt-aiNET) and artificial immune system (AIS) to train the connecting weights of a feedforward neural network for their RFID positioning method. A recent Pediatric Risk of Mortality Prediction Tool (PROMPT), for real-time prediction of all-cause mortality, is mentioned in [12].

In the next few sections, some important shallow networks are discussed and are used as the classifiers in this book. Shallow networks usually have only one hidden layer compared to deep learners that might have several hidden layers of different types.

4.2.3 Support Vector Machines

SVMs are supervised learning models. In the known feedforward and perceptron algorithms, the aim is to get a local or global minimum of misclassification errors, whereas in SVM the objective is to maximize the hyperplane margins between different classes. Figure 4.5 illustrates the SVM for a two-dimensional dataset.

The problem here is to find the right weight w and offset b to maximize the distance of the hyperlines in Figure 4.5.

The set of training set

$$(\mathrm{y}_1, \mathbf{x_1}), \ldots, (y_n, \mathbf{x_n}) \;\; y_i \in -1, 1 \tag{4.6}$$

are linearly separable if there are a vector $\mathbf{w}$ and scalar b where inequality

$$y_i(\mathbf{w}.\mathbf{x_i} - b) \geq 1 \;\; y_i \in \{-1, 1\}, \; i = 1, \ldots, n \tag{4.7}$$

can be satisfied. As shown in Figure 4.5, the term $2/||w||$ (the separation space) should become maximum or $||w||$ should be minimum. By solving this constrained minimization problem of (4.7), the optimal values for w and b can be found [13, 14]. In practice, however, it is computationally easier to minimize the reciprocal term $||w||^2$ using dual quadratic programming [15].

SVMs are already used for personal healthcare applications to process multichannel data [16]. In [17] SVM is used to distinguish the walking pattern of sensors on the arms and legs. Jo et al. [18] suggested an intelligent approach prediction method of tag detection rate using support vector machines. In this book, SVM is one of many classifiers used to distinguish between tag responses.

4.2.4 KNN as a Lazy Learner

K-nearest neighbors are called lazy classifiers because they do not learn from data. They just memorize the training dataset and use metrics to decide which category the new data belongs to. In this case, a KNN is a nonparametric or instant-based learning model. KNN's algorithm is summarized as the following:

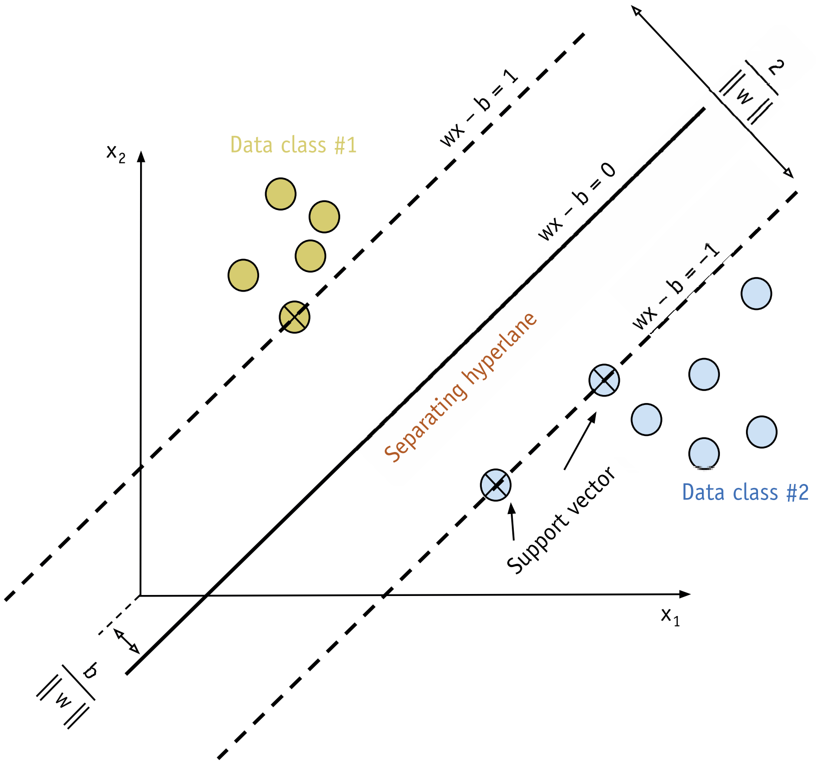


Figure 4.5 SVM illustration for two data categories. The support vectors (shown with cross signs) are those data points that are closest to the separating hyperplane.

- Choose the number of k (normally an odd number) and a distance metric;
- Use the metric and find the k-nearest neighbors;
- Assign the label to the majority vote class.

There are many metrics for the distance calculations, such as Minkowski distance [19], which is expressed as

$$D(X,Y)=\left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{1/p} \tag{4.8}$$

The Minkowski distance of (4.8) will be the Euclidean distance if $p = 2$, the city-block (Manhattan) distance if $p = 1$, and the Chebyshev distance if $p = \infty$. The cosine distance is another successful metric, and the definition of its metric for the distance between vectors X and Y is

$$\begin{aligned}&\text{Cosine Distance} = 1 - \text{Cosine Similarity}\\ &= 1 - \frac{\vec{X}\cdot\vec{Y}}{\|X\|\|Y\|} = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}\end{aligned} \tag{4.9}$$

Some of the weighted cosine and hamming distances for a better recognition can be also found in [20]. Figure 4.6 illustrates a KNN network decision for new data with two known categories. Data can be multidimensional.

Because of its light processing needs, KNN is of high interest. Xu et al. used KNN for an indoor RFID positioning algorithm [21]. Kyuwon et al. [22] used an adaptive KNN in their RFID location system. Jordy et al. [23] used KNN for location estimation of the Electronic Product Code (EPC) Gen2 tags with the help of the trilateration technique, which is knowing the location of three measurement antennas. Because of its simplicity and low training time, KNN is one of the best decoders in this book, as will be discussed later.

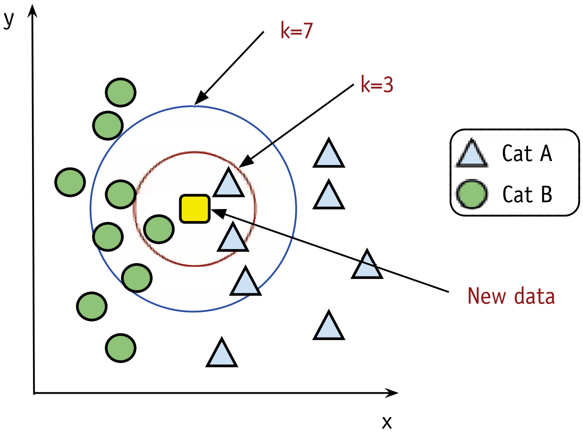


Figure 4.6 An illustration of KNN to categorize new data (indicated in yellow). If k = 3, a new point will be in Cat A, and if k = 7, it will be in Cat B, so a mindful range and distance metric is needed for correct categorization based on the dataset.

4.2.5 Decision Trees Ensembles

Decision trees have been around for a long time and have been used in probability and business decisions [24, 25]. The ensemble method combines several decision trees to produce a better predictive performance rather than utilizing a single decision tree. The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner.

If the goal is to reduce the variance of the error in a decision tree, a bootstrap aggregation tree ensemble or bagged trees ensemble is used. As illustrated in Figure 4.7, a collection of data (A) is used to train the decision trees, and another subset (B) is used for

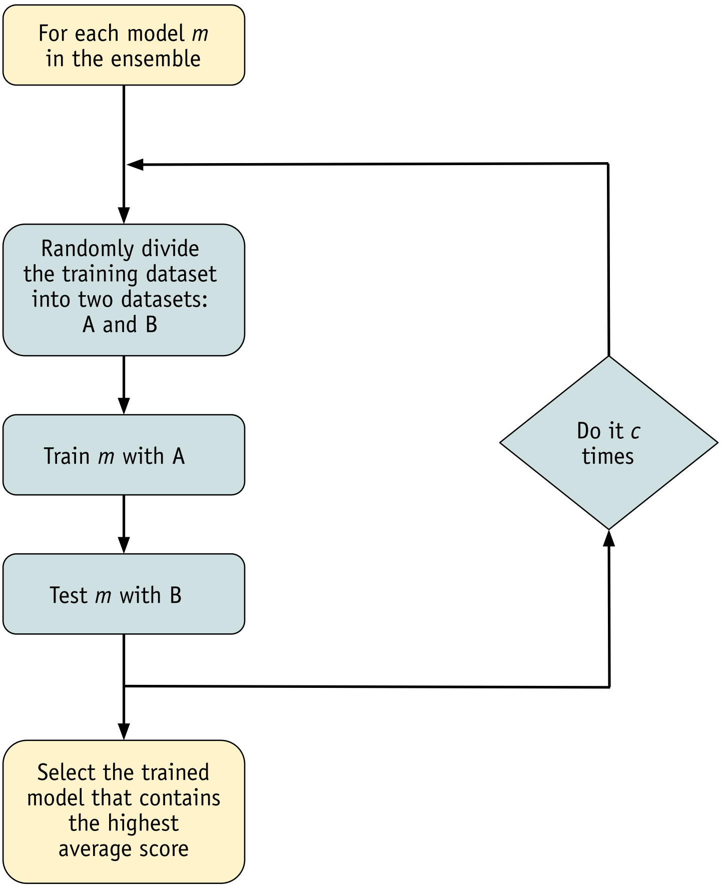

For each model *m* in the ensemble
Randomly divide the training dataset into two datasets: A and B
Train *m* with A
Test *m* with B
Do it *c* times
Select the trained model that contains the highest average score

Figure 4.7 Bucket ensemble model, in which a model selection algorithm is used to choose the best model for each problem [26].

testing. The training is done for these random subsets for *c* times. The final result is an ensemble of the different models. The average of all the predictions with the highest score will be used, which is more robust than a single decision tree.

A bootstrap ensemble is another ensemble technique in which the mean of the error is intended to be reduced in the distance cost function. In this technique, learners learn sequentially with smaller sets of data and create the models. The best subsets of models with minimum errors are used as the estimators.

Here are a few examples of the background literature for using ensemble networks. Catal et al. [8] used an ensemble of classifiers (including MLP and logistic regression) for a better classification of accelerometer-based human body activity recognition. Yang et al. [27] used an ensemble of online sequential extreme learning machines (OS-ELM) for RFID tag positioning. Decision tree ensembles will be used among other pattern recognition methods to decode a tag's data in this chapter.

4.2.6 Deep Learning Methods and Frameworks

As deep learning is used in this work for recognizing 2-D tag images, a brief background is presented. Shallow networks, or networks with one or a few hidden layers, were first used around the 1980s. Both feedforward networks and pattern recognizers are shallow networks. There was a delay in teaching more complex networks (with a few hidden layers) until 2006 [5, 28] when deep learning techniques were developed. Afterward, a new era of machine learning came to be in almost all big data areas, such as computer vision [29], speech recognition [30], and autonomous surveillance [31], to name a few.

Deep learning is the application of artificial neural networks that contain more than one hidden layer with various functions for each layer normally. The key point on what empowers a deep net to recognize complicated patterns is its ability to break complex patterns into simpler ones. For face recognition in an image, for example, a deep network will first try to detect edges and simple shapes. Then it begins to learn about more complex objects like eyes and noses. The last stage in learning is to decide which shapes should be put together to define a human face. The downside in deep net use is their demand for higher computation speed [32]. It takes much longer to train a deep network compared to a shallow one. A fast central processing unit (CPU)/graphics processing unit (GPU)s can do the training in a matter of days or weeks.

A deep learning framework is a combination of interface and libraries that makes the development of deep learning models easier and quicker without getting into the details of the underlying algorithms. Machine learning could not get that advanced if there were no frameworks. Frameworks create the automation for many aspects of machine learning, and they make it possible for developers to reuse the code. The most popular high-level frameworks are Caffe, Theano, and TensorFlow.

Caffe was created in the Berkeley Vision and Learning Center [33]. Caffe's models are defined in a configuration file instead of being coded directly. Caffe developers claim to have the fastest convolutional neural networks (CNNs) implementation available [34]. Theano was initially designed for fast, stable, symbolic operations, including symbolic differentiation [35]. Theano's power is generating dynamically optimized C-code, which can be transparently executed on a GPU up to 140 times faster than an equivalent CPU implementation. In Theano it is possible to efficiently define, optimize, and evaluate mathematical expressions involving multidimensional arrays. Finally, TensorFlow, the most recent of the three, was designed by Google and is currently experiencing the fastest growth in usage [36]. As of this writing, TensorFlow has slightly slower performance than the other two on benchmarks, but is easier to deploy on multiple GPUs.

There are a number of frameworks built on top of these platforms, such as Lasagne and Keras, that allow further abstraction and thus ease and speed of development. Keras is a good choice because it uses either TensorFlow and/or Theano as a backend processor and it provides a simpler model for deep learning development [37].

As a background review for using deep learning techniques on RFID domains, Li et al. [38] used CNN for moving activity recognition on a passive RFID system. Umbricht et al. [39] used deep learning with wearable sensor signals for the quantified evaluation and continuous monitoring of behavioral problems. T. Wang et al. [40] used a stacked denoising autoencoder (SDA) for learning robust features from RFID sensor signals. C. Wang et al. [41] used a deep belief network (DBN) along with multilayer backpropagation feedforward in an order completion time (OCT) prediction. General guidelines for choosing the right deep learning technique are summarized in Table 4.1.

4.2.7 Machine Learning in Chipless RFID and the Gaps

In the last few sections, different architectures of NNs have been discussed. These include shallow networks such as feedforward neural networks, SVM, KNN, decision tree ensembles, and deep learners, such as CNN and DBN. However, all of these methods are not being used in chipless RFID applications. In this book, first a set of known architectures will be used to decode the data stored on a chipless RFID frequency response at 60 GHz in the following sections of this chapter, and the deep learning methods will be further discussed in Chapter 5. Before going into the training, our data collection method is presented in the next section.

4.3 Data Collection Methodology

Figure 4.8 shows the overall data collection method in these simulations. This figure is used for the experiment and simulations, as explained in the steps of the flowchart:

Table 4.1
Deep-Learning Method Application Comparison

Deep Learning Name	Learning Method	Characterization	Example
Restricted Boltzmann machine (RBM) Autoencoders network	Unsupervised	Feature extraction, pattern recognition	Cancer classification [42], finding unknown patterns [43]
Recursive neural tensor network (RNTN)	Supervised	Text processing, parsing name, handwriting recognition, object recognition	Compositionality in language [44], sentiment analysis [45]
Recurrent nets	Supervised/ unsupervised (semisupervised)	Character level operations, speech recognition, memorizing	Apple Siri [46], time series analysis [47]
DBN CNN	Supervised	Image recognition	Mood fatigue detection [48], Facebook image recognition [49], behavioral activity monitoring [39]

A deep learning system based on CNN and Keras frameworks will be developed in this book to decode 2-D image-based chipless tag data with high accuracy in Chapter 5.

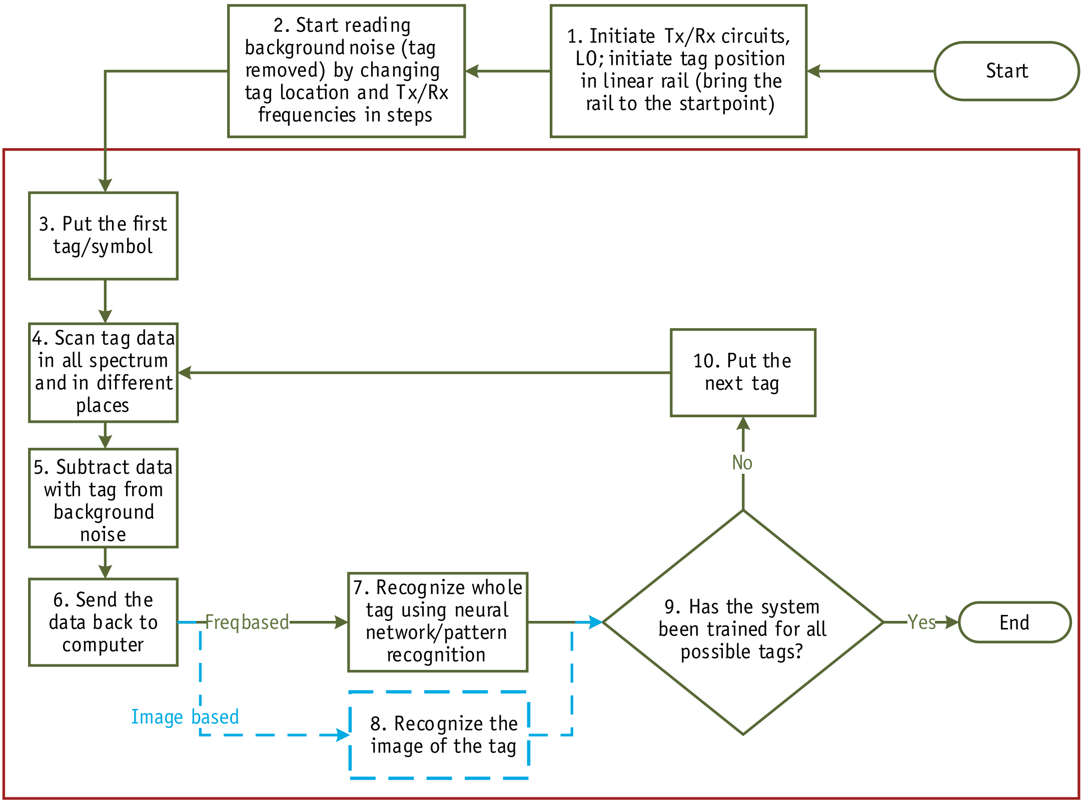

Start
1. Initiate Tx/Rx circuits, LO; initiate tag position in linear rail (bring the rail to the startpoint)
2. Start reading background noise (tag removed) by changing tag location and Tx/Rx frequencies in steps
3. Put the first tag/symbol
4. Scan tag data in all spectrum and in different places
5. Subtract data with tag from background noise
6. Send the data back to computer
Freq based
Image based
7. Recognize whole tag using neural network/pattern recognition
8. Recognize the image of the tag
9. Has the system been trained for all possible tags?
Yes
End
No
10. Put the next tag

Figure 4.8 Flow diagram of the data collection method for this chapter. Data collection in simulations steps are in the red-outlined block only, and the experimental lab measurements includes all steps [50]. (Image courtesy of IEEE.)

1. Initialize RF, LO, and put the linear rail to the first position.
2. Collect the background noise (structural mode data) in the whole frequency range. Move the linear rail to all other positions and collect the backscatter data for each position in all frequency steps.
3. Put the first tag in front of the reader.
4. Scan the tag in the whole spectrum. Move the rail in defined steps and rescan the tag.
5. Subtract step 4 collected backscattering data from step 2. There is no background noise in the simulations.
6. Send the data in step 5 to the computer for further processing.

7. Recognize the tag using feedforward or pattern recognition methods. This is a vector-based method. In these AI methods, frequency-space-based backscattering data is provided as inputs to the supervised network, and tagID is the output.
8. Recognize the tag using vector-based image and pattern recognizers. As will be explained in Section 4.6, a vector of 2×10 is made initially as a tag image, and a pattern recognizer is used to detect the tagID.
9. If all the available tags are scanned, stop the process.
10. Continue steps 4 to 9 for all of the remaining tags.

The chipless tags are simulated in the CST and MATLAB in steps 3 to 10 in Figure 4.8. For the experiments, all steps in that figure are followed. Despite the decoding method used, in this book CST/MATLAB software is used for the simulated data collection and the VNA and the developed reader presented in Chapter 3 for the experimental datasets. In this section, these data collection methods are illustrated.

4.3.1 Data Collection in the Simulations

Figure 4.9 demonstrates the simulation in the CST microwave studio suite. A tag made of five letters is centered in front of the cross-polarized reader antennas, which are connected to the transmitter and the receiver of the system. The size of the antennas is the same as the one used in our experiments [51].

The technique for tag detection should match the tag used and its specifications. In this chapter a hybrid method of the frequency and position has been followed, as shown in Figure 4.8. The method used is similar to the frequency-domain tags if just pattern recognition is used, and is similar to the SAR if image-construction is used.

Figure 4.10 illustrates one of the tag's responses in the linear rail. The scanning is done within the azimuth resolution, which is 6 mm according to the maximum resolution possible by the SAR method. This maximum resolution of SAR is D/2, where D = 1.2 cm is the antenna aperture, referring to (2.18). The strongest signal happens once a tag is directly facing the Tx/Rx antenna in position

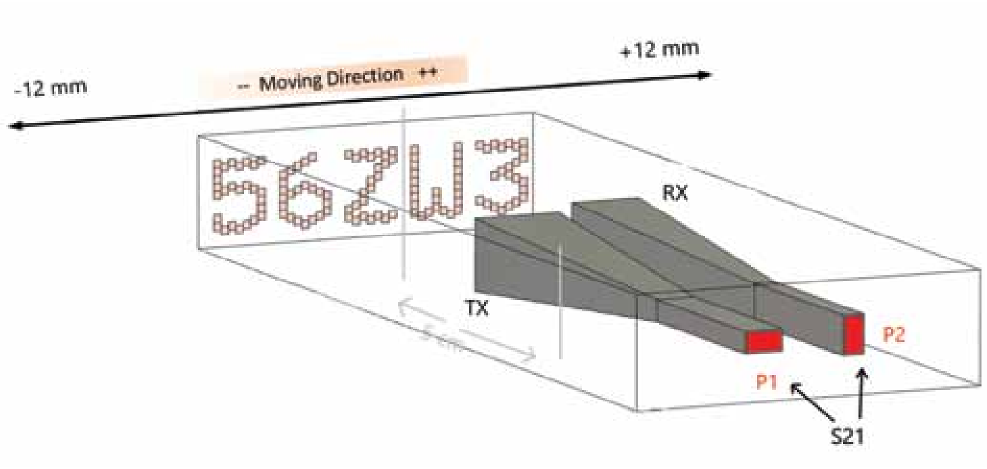


Figure 4.9 Moving tags in front of reader antennas in the simulations. Reader antenna distance to tag is 5 cm, and there is 1-cm separation between cross-polarized Tx and Rx antennas. The horn antennas are from [51].

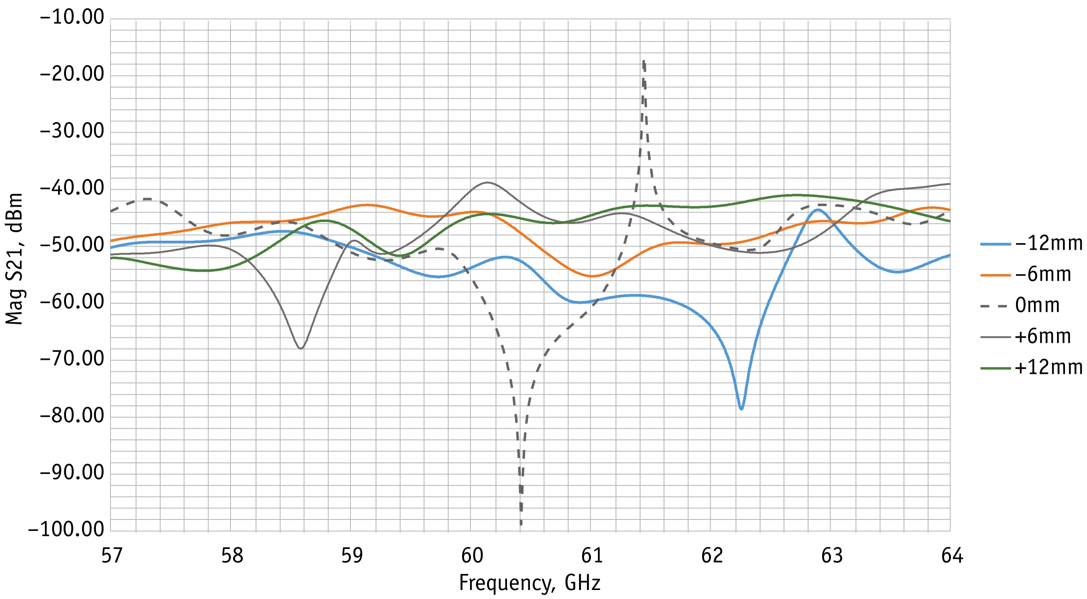


Figure 4.10 Simulation results of the tag movements in front of the reader. A few tag responses within the azimuth resolution spaces with $\delta x = 6$ mm are shown [50]. (Image courtesy of IEEE.)

0, which is illustrated by the dashed line in Figure 4.10. Moving the tag to the right or left by the D/2 creates weaker signals. These resultant scattering waveforms, along with the reader to the tag location, are the inputs to the neural network and pattern recognizers discussed in the next sections.

4.3.2 Data Collection in the Experiments

In the experiments, there are two measurement cases: using two-port VNA or using two-port developed modular reader, which is fully described in Chapter 3. In both cases, background noise data (without the tag) is collected first and then subtracted from the measurement in the presence of the tag.

Figure 4.11 shows the data collection method in the experiments. For every position of tag/antenna, the 7-GHz bandwidth of 57–64 GHz is split into five steps of 1.5 GHz each. This frequency range splitting is because of the restrictions in the Rx/Tx boards,

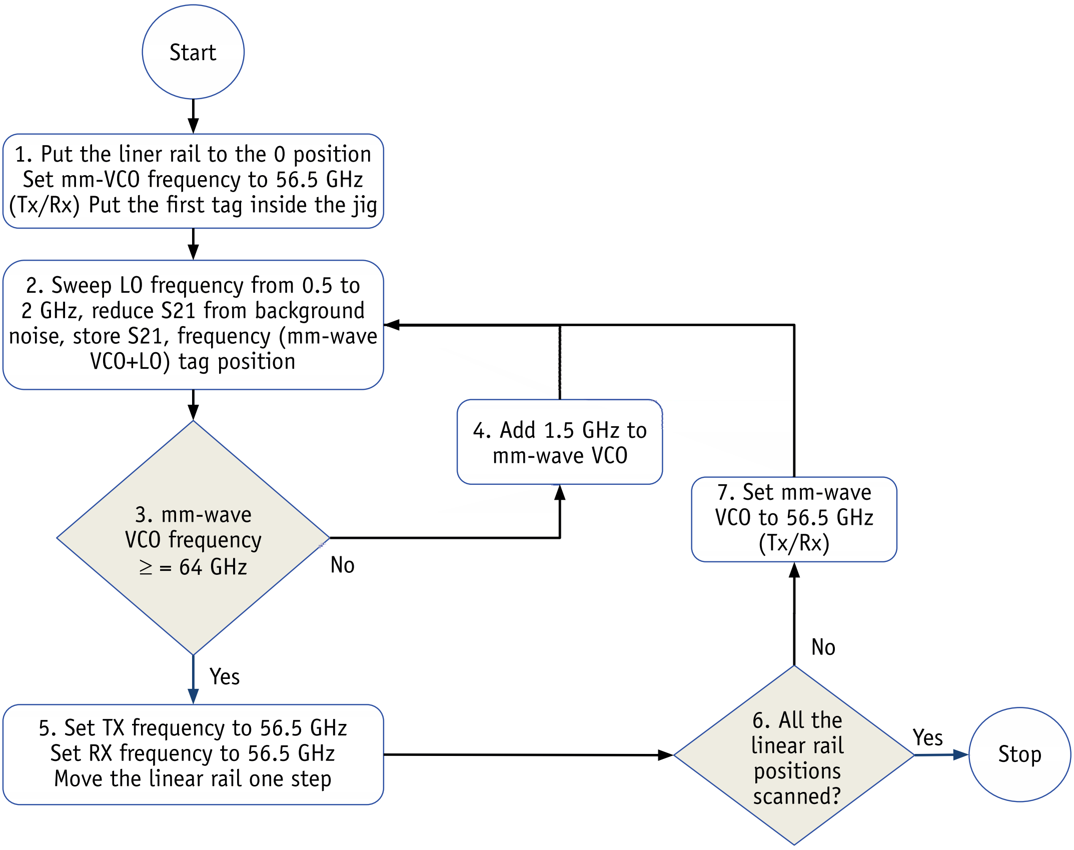

Start
1. Put the liner rail to the 0 position
Set mm-VCO frequency to 56.5 GHz
(Tx/Rx) Put the first tag inside the jig
2. Sweep LO frequency from 0.5 to
2 GHz, reduce S21 from background
noise, store S21, frequency (mm-wave
VCO+LO) tag position
3. mm-wave
VCO frequency
≥ = 64 GHz
No
4. Add 1.5 GHz to
mm-wave VCO
Yes
5. Set TX frequency to 56.5 GHz
Set RX frequency to 56.5 GHz
Move the linear rail one step
6. All the
linear rail
positions
scanned?
No
7. Set mm-wave
VCO to 56.5 GHz
(Tx/Rx)
Yes
Stop

Figure 4.11 Tag scanning method in the experiments. This process is done for every available tag.

as described in Chapter 3. First, the background noise data is collected in all range positions and in all frequency steps. For every tag, the backscattered data from the tag is stored in a vector that contains tagID, backscattered data (subtracted from background noise), and frequency that is the sum of mm-wave VCO frequency and LO (according to Figure 3.6). The process continues for all the SLAR scanning locations (6-mm distance separations) and for all the available combinations of the tags.

Once all data is collected it is time for the decoding phase. In the next few sections, the two different decoding procedures will be discussed, as per Figure 4.8 blocks 7 and 8. In the pure decoding using AI of block 7 of that figure, feedforward and pattern recognition methods are discussed, and in the image-based method of block 8, a new SLAR decoding method will be illustrated in detail.

4.4 Using Feedforward Networks

Figure 4.12 shows the feedforward network used for initial realization of the system. The inputs are the tag location (as in Figure 4.9), frequency (in GHz), magnitude, and phase of the S_{21} parameter (as in Figure 4.10). The output is set to be the tagID.

In this fully supervised feedforward network, the expectation is that the trained network is able to recognize the tagID based on the four inputs given. Data vectors collected from Section 4.3 are

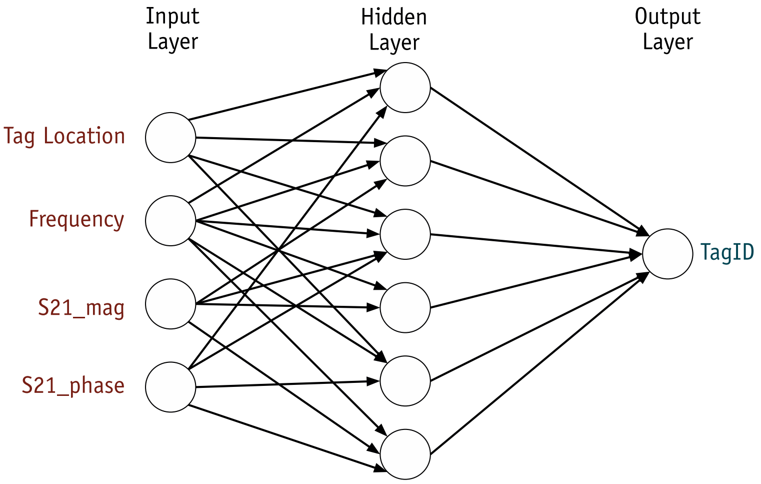


Figure 4.12 Graph of the feedforward network used, showing inputs and the tagID as output.

shuffled first, normalized, and then divided into three parts: training, validation, and test data. The validation and test data are not used during the training session. A few trials with different hidden layer neurons are done and the results are explained in the following section.

4.4.1 Feedforward Network Results

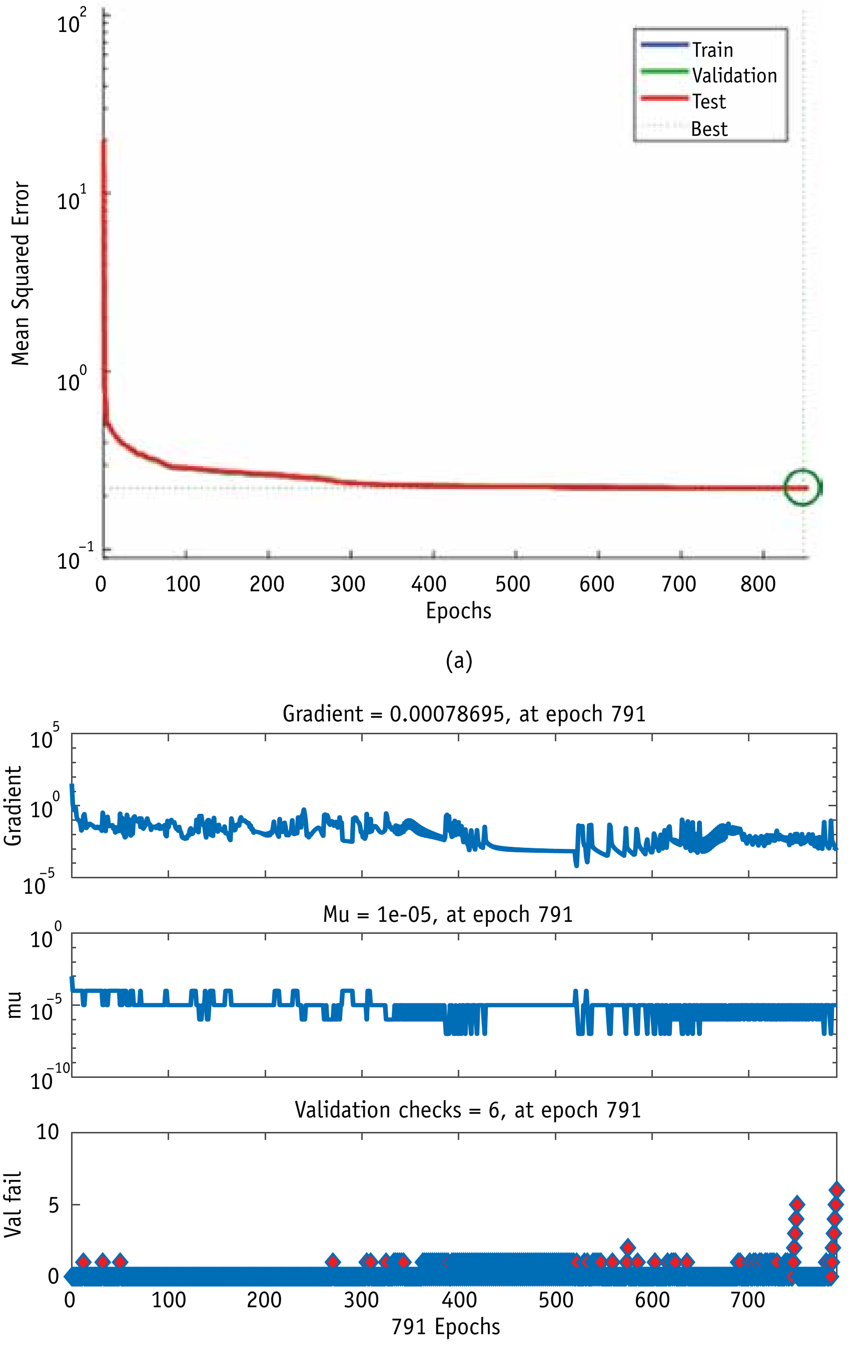
Figure 4.13 shows the convergence graphs for one of the trials done. In this trial, the number of neurons in the hidden layer is chosen to be 100. In Figure 4.13(a), the MSE graph for the feedforward network is shown. Best validation performance is 0.22146 at epoch 848, and in the Figure 4.13(b) gradient, μ and validation failure graphs are illustrated. For an ideally successful training, the MSE is in a global minimum, the gradient and mu[1] have decreasing trends, and the validation failure number does not exceed a certain defined error (here it is 6). If validation failure goes beyond that predefined error number, this is a sign that the network is getting overtrained and training should be stopped.

Figure 4.14 demonstrates the error histogram for a bin of 20 groups. Zero error is shown with a line in the middle. The bigger the bins closer to the zero error are, the better the overall performance is.

The higher the number of hidden neurons, the more complicated problems can be solved. In this initial research, networks with 10, 25, 50, 100, 200, and 500 hidden neurons were tested. With higher hidden neurons, namely 250 to 500, the network had the ability to decode misaligned or rotated tags, as shown in Figure 4.15.

Higher neuron networks come with an obvious cost: training time. As illustrated in Figure 4.16, training a network with 50 neurons takes just a few minutes on a core i7-4790 CPU 3.6-GHz, 16-GB RAM machine (with no GPU), whereas for 500 neurons, training takes more than 6 hours, which is an exponentially growing trend (the red line in Figure 4.16). Training time is measured with MATLAB tic/toc commands.

[1] Mu, or μ, is momentum constant which is included in weight update (4.4). Its purpose is to avoid the problem of the local minimum, in the case that the network training getting stuck into a local minimum and convergence does not happen. The value of mu can be between 0 and 1, although normally it is set from 0.8 to 1.



(a)

(b)

Figure 4.13 Feedforward network training graphs. Data is shuffled initially, which is why the graphs in (a) are matched for different sets.

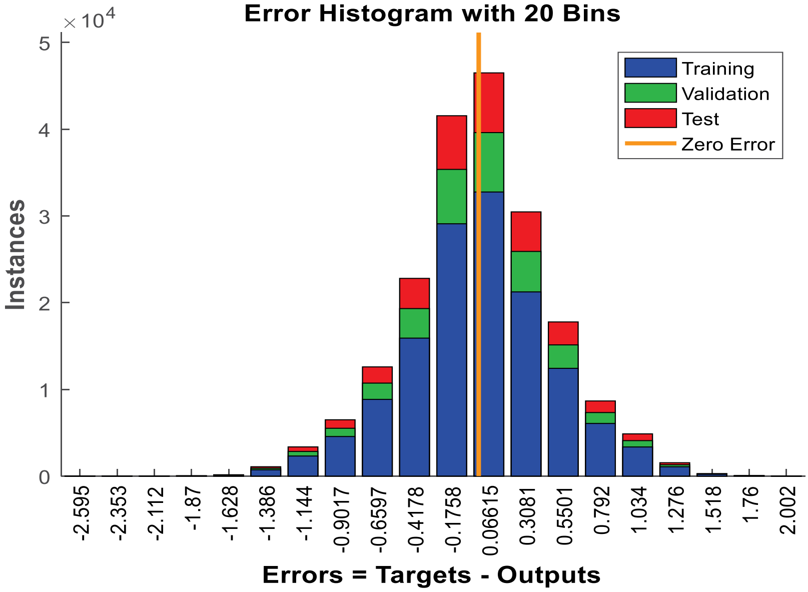


Figure 4.14 Error histogram for a network of 100 hidden neurons.

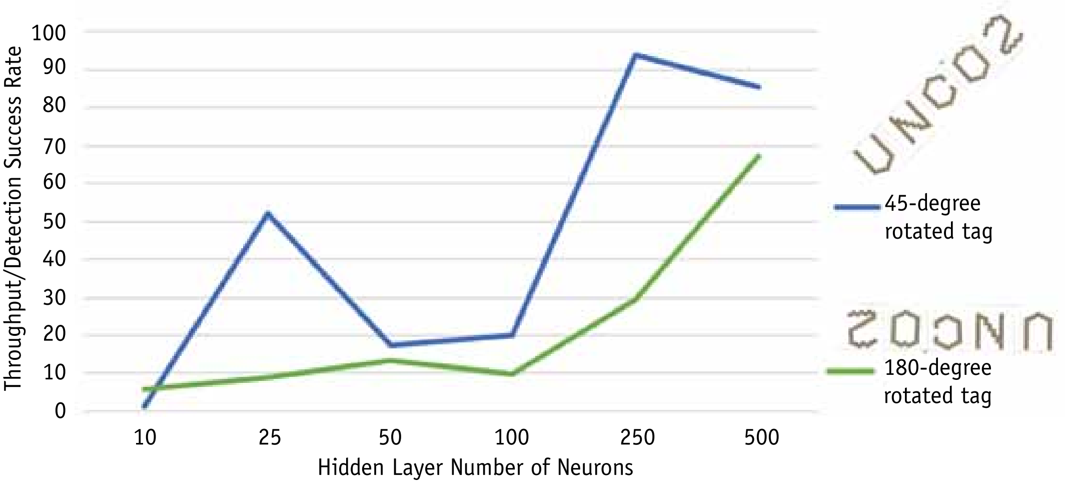


Figure 4.15 Feedforward network output graphs for a different number of hidden layer neurons: detection success rate for misaligned tags.

Although feedforward networks are powerful enough in solving simple problems, their vanishing gradient problem as described in Section 4.2.2 is sometimes challenging once the number of samples increases. The network training should also be

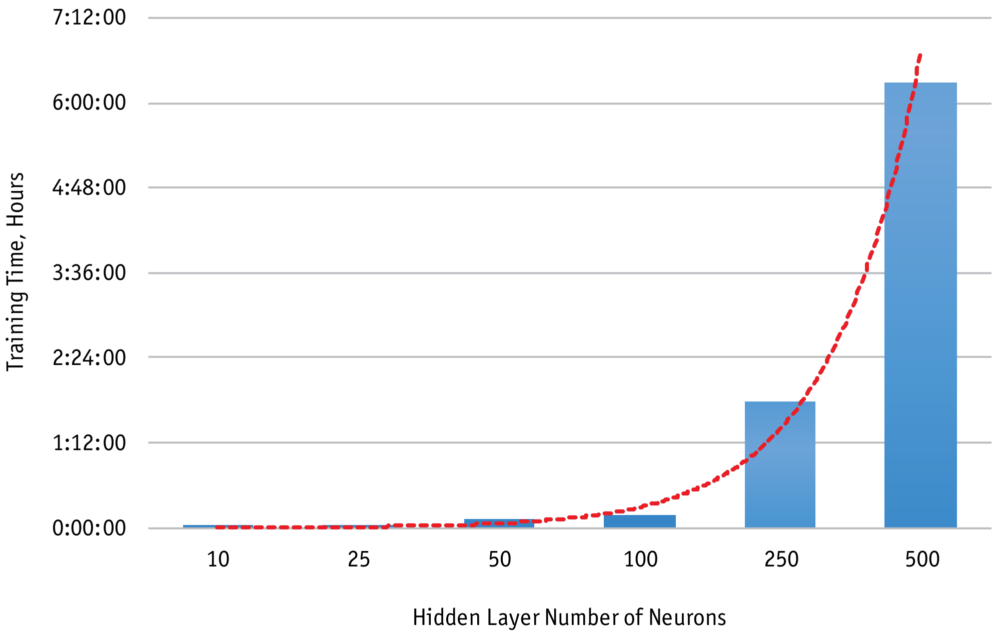


Figure 4.16 Feedforward network output graphs for a different number of hidden layer neurons: training time.

tried a few times to overcome the problem of sticking in the local minimum.

For the interested reader, the dataset outcome of this simulation is available in [52].

4.5 Using Pattern Recognition Methods

Pattern recognition is one of the most important parts of AI. It focuses on the description, measurement, and classification of patterns included in various data. Like any machine learning, a typical pattern recognition system involves the following steps: defining the problem, creating the dataset, and transforming the data (preprocessing), feature extraction, and classifier design (training a model) and postprocessing. Pattern recognition ability to differentiate between many samples decreases if the number of samples goes high and their responses become closer to each other. Pattern recognizers are normally composed of shallow networks with a few layers. The main steps in any classification algorithm are illustrated in Figure 4.17 [53]:

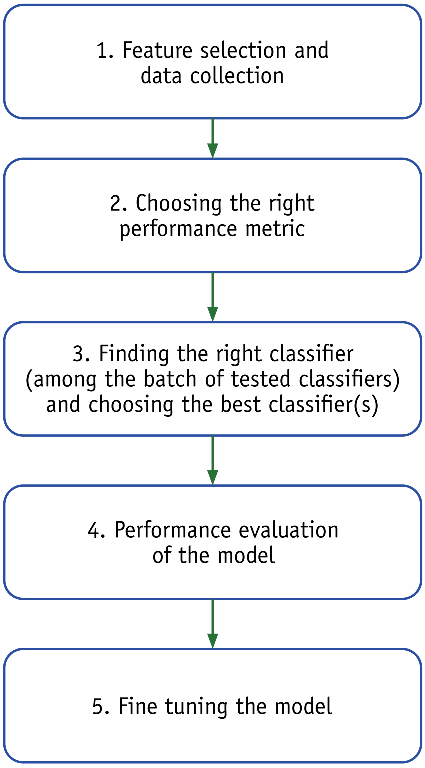

1. Feature selection and data collection
2. Choosing the right performance metric
3. Finding the right classifier (among the batch of tested classifiers) and choosing the best classifier(s)
4. Performance evaluation of the model
5. Fine tuning the model

Figure 4.17 Flowchart of the main steps in classification problems.

Feature selection and data collection. Out of the available data, some features are more important than others. In our study, for example, knowing the tag location might give more information to the classifiers than the tag's phase at a particular frequency. There are a few techniques for feature selection. One of the most known ones is principal component analysis (PCA), in which the data is reduced into fewer dimensions by projecting it to onto a unique orthogonal basis. Another technique is nonnegative matrix factorization used with physical-based data that does not have negative values. Proper feature selection has the following effects [54, 55]:

- Improving the accuracy of the algorithm;
- Reducing overfitting, as less redundant data is available;
- Boosting the performance on high-dimensional data;

- Reducing training time, because of less noisy or redundant data.

In this book PCA is used, and its results will be discussed further.

Choosing the right performance metric. Performance metrics in this book are chosen to be training accuracy, area under curve (AUC), and training time.

Finding the right classifier (among the batch of tested classifiers) and choosing the best structure. Many classifiers were used and their initial performance comparison was done based on the metrics in step 2.

- *Performance evaluation of the model.* Further evaluation of the chosen classifiers of step 3.
- *Fine tuning the model.* After converging to the best pattern recognizer, its parameters are further tuned to improve the final performance metrics.

In this section, 54 classifiers (22 ensembles, 12 SVMs, 7 KNNs, 12 trees, 1 linear discriminate) will be compared for their performance and the metrics will be discussed.

For the interested reader, to replicate the results the dataset outcome of this experiment is available in [56].

4.5.1 Pattern Recognizer Results

The main pattern recognizers used in this book were discussed in Section 4.2. This section explains the results from many pattern recognition methods. A dataset of 27 different tags in three different positions in front of the antenna (81 data states) was collected. The dataset then was applied to the pattern recognition techniques.

Figure 4.18 shows one of the results. The total dataset is divided into the three batches: training, test, and verification. The test and verification dataset batches are not used during the training session. The term “accuracy” is the percentage of correct matches for verification data (verification is normally 10% of total data). The AUC is the percentage of the total data that can be classified correctly, which is a metric of the coverage.

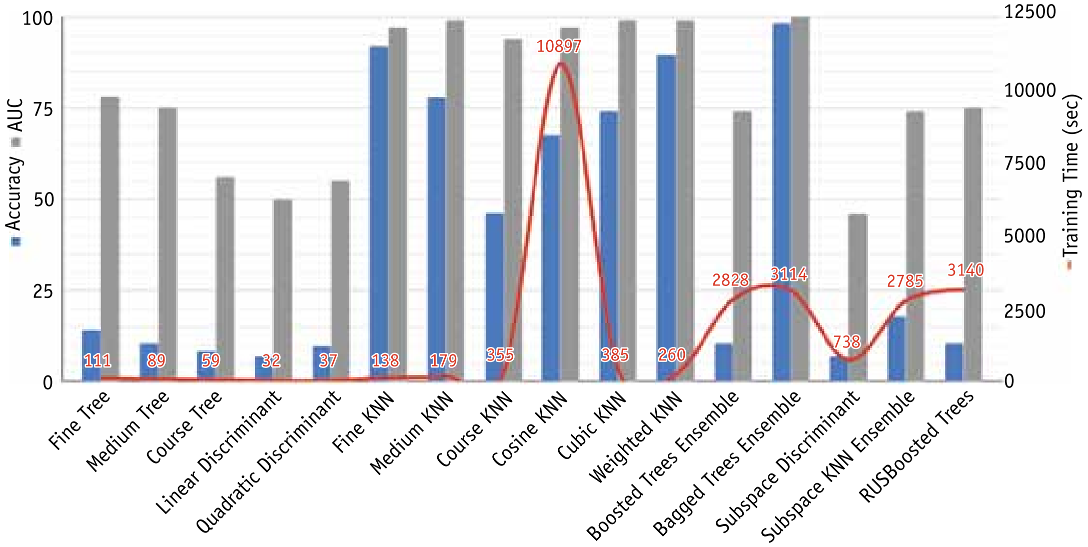


Figure 4.18 Comparing accuracy, AUC (left vertical axis), and training time (right vertical axis) for some of the better-performing pattern recognizers for our dataset [57]. (Image courtesy of IEEE.) Positions in front of the antenna (81 data states) were collected. The dataset was then applied to the pattern recognition techniques.

Besides accuracy and coverage, one of the most important aspects of the network is its training time. Every time new data is added to the training set, the whole network should be retrained to classify that new data. Although it might not happen right away, the scheduled retraining should not take longer that an accepted time.

Figure 4.18 illustrates the concept of the training time for the same set of pattern recognizers used.

Although "bagged trees ensemble" performs better than "fine KNN" in terms of accuracy and AUC in Figure 4.18, its training time is almost 20 times slower than the "fine KNN" network (3,114 sec compared to 138 sec). So, obviously the KNN network is the optimum one comparing different factors. Fine KNN uses 1 to 3 neighbors to find the classification for each point during predicting.

The faster training time of the KNN networks is not surprising though. As said in Section 4.2.4, KNNs are called the lazy learners, as they actually do not learn, but they associate every new sample to the nearest defined class.

To choose the right pattern recognizers in practice however, many factors should be considered other than the accuracy and the training time. Figure 4.19 is based on the same dataset of Figure 4.18 but with an emphasis on two other parameters, subspace dimensions for the KNNs and the maximum number of splits for the decision trees. This figure further classifies the best optimizers from the previous figures. The last four columns to the right in Figure 4.19 refer to the KNNs with different subspace dimensions, and the other 10 columns to the left are for the decision trees.

A few interesting observations are noted from this chart:

- The accuracy and the AUC (the red and yellow lines in the left vertical axis) almost follow each other. This simply means that for a network with a higher accuracy normally AUC is also high, and vice versa.

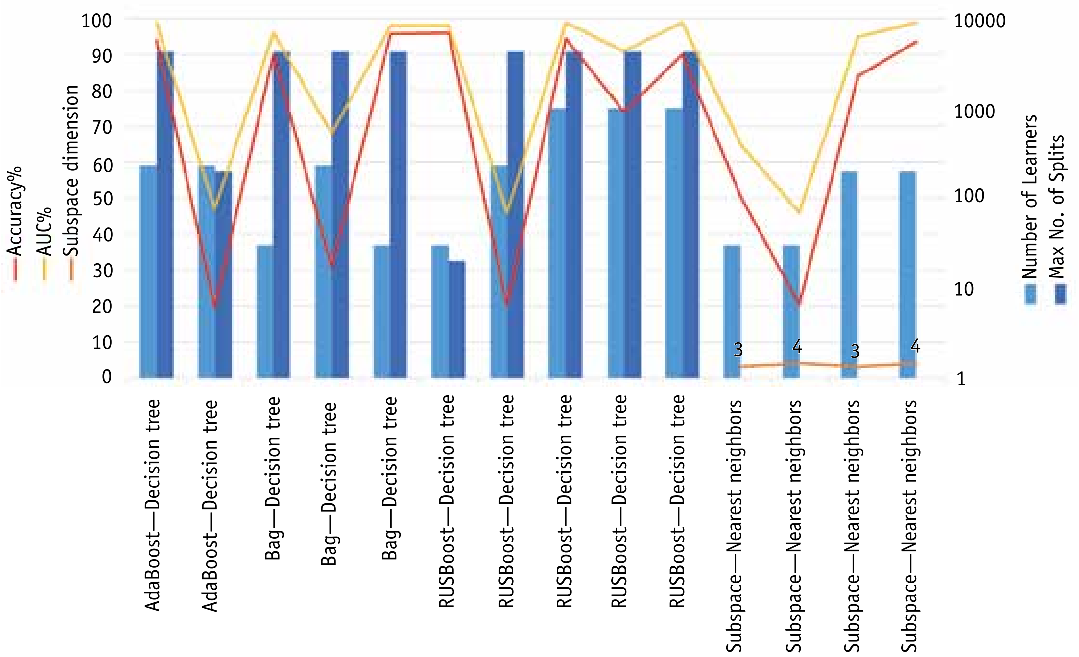
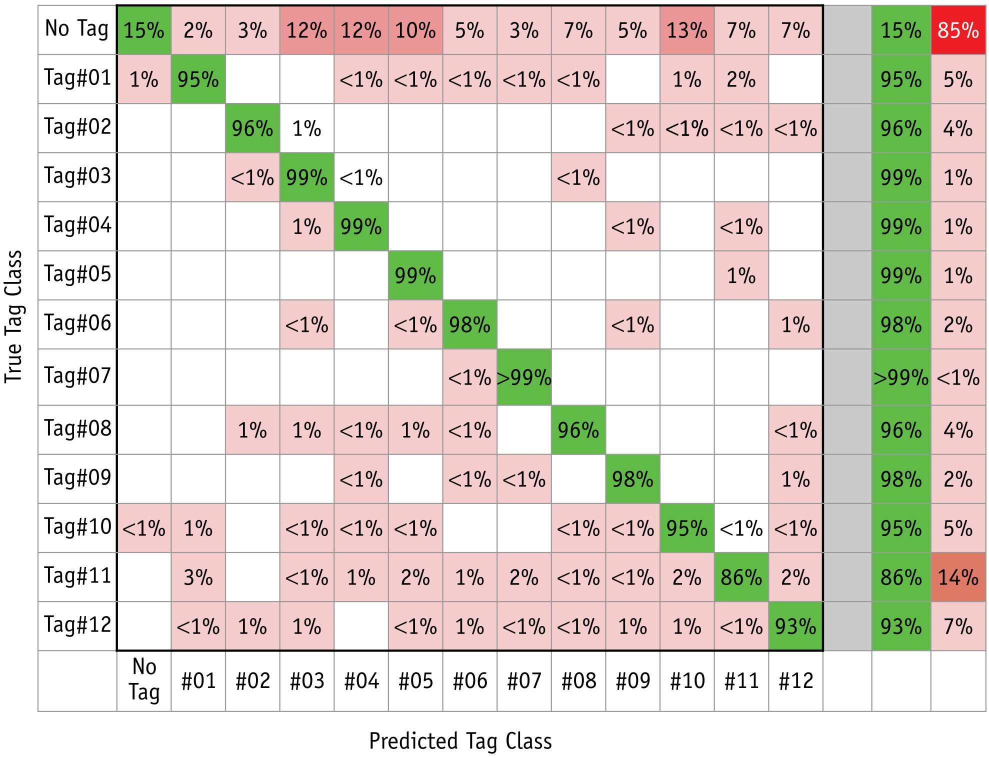


Figure 4.19 Comparing best-performing pattern recognizers: decision trees and KNN parameters. Accuracy, AUC, and "Subspace dimension" are in the left vertical axis, and "Number of learners" and "Max no. of splits" are shown in the right vertical axis. "Subspace dimension" is a property of the KNNs and "Max no. of splits" is a property of the decision trees only [57]. (Image courtesy of IEEE.)

- Both in decision tress and KNNs, the higher the number of learners (light blue bars in the secondary vertical axis), the higher the accuracy and coverage.
- The number of splits for the decision trees should be used with the right algorithm (boost or bag) with the proper number of learners to be effective. A higher number of splits in decision trees does not necessarily end up with a higher accuracy.
- For the KNNs (the last four columns in the graph), increasing the number of learners shows increased decoding accuracy, but the higher subspace dimension (3 to 4) is only effective once the number of learners is big enough. Otherwise increasing the subspace dimension can result in a lower accuracy.
- Enabling PCA normally degraded the performance metrics, such as accuracy and AUC. The PCA feature was tested on ensemble networks, decision trees, and KNN. This might happen because of low dimensions of the inputs.

Another detailed metric for the performance of the classifiers is their confusion matrices. Figure 4.20 illustrates the confusion matrix for one of the KNN trials. A fine KNN classifier has been trained with a dataset of 12 chosen tags in three different positions in front of the antenna. A confusion matrix for more than 12 tags is difficult to display. For those 12 tags, the “true tag class” is the actual tag under the measure on the vertical axis, and the “predicated tag class” is the predication results on the horizontal axis. The diagonal (green) shows the decoding success rate for each tag. That number is the chance that particular tag is in front of the reader and the reader detects it correctly. For each tag, there is a chance that the system detects them wrong (misdetection or failure rate). Those “green” success rates are also reflected in the right side of the table for each tag, with the pink failure rate (1 = success rate) next to it. The best decoding rate for this experiment is 99% (for Tag#03 and Tag#04), and the worst one is 86% (for Tag#11). Tags with better decoding rates have more distinguishable frequency responses and a better alphanumeric combination in the tag.



True Tag Class

No Tag	15%	2%	3%	12%	12%	10%	5%	3%	7%	5%	13%	7%	7%		15%	85%
Tag#01	1%	95%			<1%	<1%	<1%	<1%	<1%		1%	2%			95%	5%
Tag#02			96%	1%						<1%	<1%	<1%	<1%		96%	4%
Tag#03			<1%	99%	<1%				<1%						99%	1%
Tag#04				1%	99%					<1%		<1%			99%	1%
Tag#05						99%						1%			99%	1%
Tag#06				<1%		<1%	98%			<1%			1%		98%	2%
Tag#07							<1%	>99%							>99%	<1%
Tag#08			1%	1%	<1%	1%	<1%		96%				<1%		96%	4%
Tag#09					<1%		<1%	<1%		98%			1%		98%	2%
Tag#10	<1%	1%		<1%	<1%	<1%			<1%	<1%	95%	<1%	<1%		95%	5%
Tag#11		3%		<1%	1%	2%	1%	2%	<1%	<1%	2%	86%	2%		86%	14%
Tag#12		<1%	1%	1%		<1%	1%	<1%	<1%	1%	1%	<1%	93%		93%	7%
	No Tag	#01	#02	#03	#04	#05	#06	#07	#08	#09	#10	#11	#12			

Predicted Tag Class

Figure 4.20 Fine KNN confusion matrix for a dataset of 12 tags. Although the misdetection (right-most column) rate is low, it is still a real concern with chipless RFID tags [57]. (Image courtesy of IEEE.)

One interesting but problematic observation is that a decoder system merely based on frequency data decoding has a chance of misdetection. Even for the best detectable tags in Figure 4.20, there is a 1% chance that the tag is detected wrongly. Another example is the “no tag” column. Even if there is no tag in the reader, meaning that the background noise is being measured, there is only a 15% chance that system correctly detects the situation. Otherwise the “no tag” state might be reported as one of the known classes. This situation can be avoided by some further actions, such as using two or three decoding methods at the same time and comparing their results.

As a brief conclusion, the different AI methods examined are very good at distinguishing different signatures in the frequency response of a chipless tag in low-cost substrates. These AI-based methods are not based on the Q-factor of the tags, which is a great advantage to using low-cost substrates.

Although different AI methods covered so far (including feedforward networks and pattern recognizers) can have tag success rates as high as 98.1%, as in Figure 4.18 for chipless tags, their failure rate should be addressed as it can be a big concern for applications that require higher reliable decoding. These applications include chipless RFID in financial institutes, such as banknotes, and in grocery chains. This misdetection problem will be expanded more in Section 4.6 and Chapter 5.

4.6 Using CW-SLAR Imaging

One of the initial ideas behind tag scanning in this book was to make an image of the tag using SAR techniques. This process is indicated with light blue lines in Figure 4.8. Conventional SAR generally uses pulse radars and phased arrays to shape the pulse [58, 59]. The image construction can be created with very different techniques. Images are based on the fact that SAR determines the 3-D reflectivity from measured data. Basically, it is a sort of spectrum estimation, because for a particular pixel of the resultant image, the complex data SAR from measurements is a sampled version of the Fourier transform of reflectivity in the elevation direction. Because the Fourier transform is not balanced, the spectral estimation techniques are used to improve the resolution and reduce speckle compared to the results of conventional Fourier transform SAR imaging techniques [60]. Here are the steps in use for image construction [61]:

- The raw data is acquired and then is segmented into subapertures. It contains a 1-D signal.
- The range and azimuth of the received signal is created by a location dependent matching filter. Grid-division is now done as per azimuth (the azimuth at that particular time).
- Calculations for the slant range (the distance between the antenna's phase center and the middle point on the beam footprint) are done for every azimuth time using coordinate transformations.

- All the subapertures of the image are superimposed to make the final image.

Figure 4.21 illustrates this concept. Before going into details, it is worth mentioning that the approach for data collection in this study is similar to the stop-go in geosynchronous satellite synthetic aperture radar (GEO-SAR), meaning that there are no moving objects in the scene. In GEO-SAR, the assumption is that transmitting and receiving the pulses in the satellite are at a stationary point, which means that all movements between and within the signal pulse are ignored for the target and the satellite [58].

Assuming the reader has a slant angle of θ toward the middle of the beamwidth footprint of ABCD and the reader is scanning the tag with the speed of v, Figure 4.21 shows a simple illustration of the scene for SLAR technique development. The SLAR formulas from (4.11) to (4.18) are adopted or modified from [61]. The reader moves with speed v and perpendicular distance to the scanning tag is h. $p(x,y)$ is the arbitrary point in the beam footprint, in which

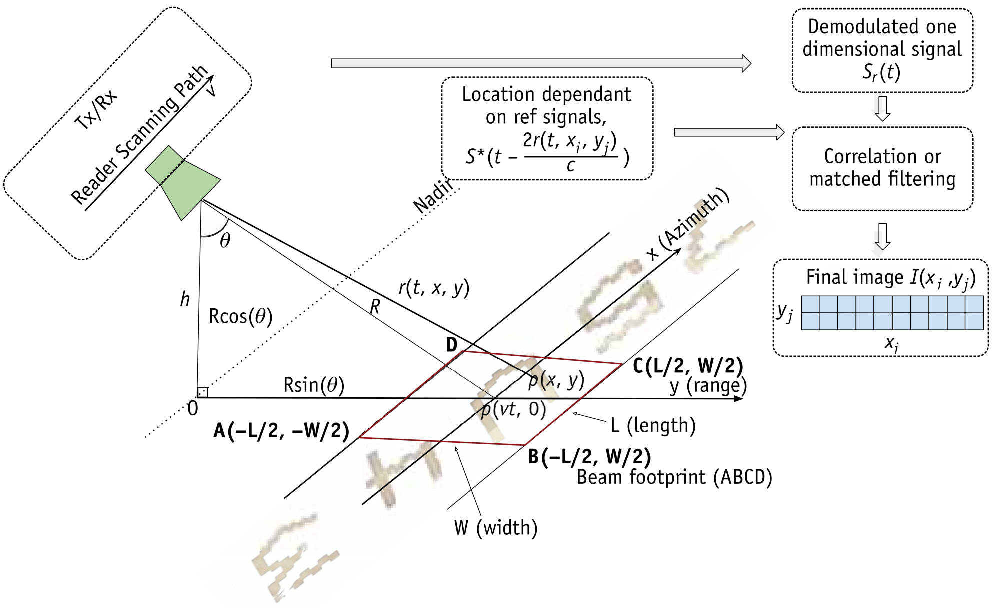


Figure 4.21 SLAR geometry concept in reading letter-based tags [57, 61]. (Image courtesy of IEEE.)

p(*vt*,0) is the center. If L_1 is the distance from point O to *p*(*x*,*y*), the L_1 equation is

$$L_1(t,x,y)=\sqrt{(R\sin\theta+y)^2+(x-vt)^2} \tag{4.10}$$

in which *θ* is the beam incident angle. If the *R* is the distance from the antenna to the center of the beam footprint, *r*(*t*,*x*,*y*), the slant range from the antenna to the observation point of *p*(*x*,*y*) is expressed as

$$\begin{aligned} r(t,x,y)&=\sqrt{(L_1)^2+(R\cos\theta)^2} \\ &=\sqrt{(R\sin\theta+y)^2+(x-vt)^2+(R\cos\theta)^2} \end{aligned} \tag{4.11}$$

Following a stop-and-go procedure, in which scanning is done in the steps once the reader has stopped moving, *v* will be zero. If *s*(*t*) is the continuous transmitted signal, the received signal $s_r(t)$ will be a superposition of a large number of backscatter signals from the beam footprint [58]; that is:

$$s_r(t)=\int_{-\frac{L}{2}}^{\frac{L}{2}}\int_{-\frac{W}{2}}^{+\frac{W}{2}}\sigma(x,y)s\left(t-\frac{2r(t,x,y)}{c}\right)\times e^{-j\frac{4\pi}{\lambda}r(t,x,y)}dydx \tag{4.12}$$

In (4.12), *σ*(*x*,*y*) is the RCS of each (*x*,*y*) point in the scanning footprint. The output of the matched filter; that is, the constructed image in any pixel of (x_i,y_j), is determined as

$$I(x_i,y_j)=\int_0^T s_r(t)s^*\left(t-\frac{2r(t,x_i,y_j)}{c}\right)\times e^{j\frac{4\pi}{\lambda}r(t,x_i,y_j)}dt \tag{4.13}$$

where *T* is the scanning time for this particular footprint in Figure 4.21 (*T* is around 600 ms for 20,001 points sampling the VNA used). Substituting (4.12) into (4.13) we obtain

$$I(x_i,y_j)=\int_{x_i-L}^{x_i+L}\int_{-W/2}^{W/2}\int_0^T s\left(t-\frac{2r(t,x,y)}{c}\right)s^*\left(t-\frac{2r(t,x_i,y_j)}{c}\right) \tag{4.14}$$
$$e^{-j\frac{4\pi}{\lambda}r(t,x,y)}e^{j\frac{4\pi}{\lambda}r(t,x_i,y_j)}\sigma(x,y)dtdydx$$

Equation (4.14) can be rewritten as the following:

$$I(x_i,y_j)=\int_{x_i-L}^{x_i+L}\int_{-W/2}^{W/2}A_{x,y}(x_i,y_j)\sigma(x,y)dydx \tag{4.15}$$

in which $A_{x,y}(x_i,y_j)$ is ambiguity function [61, 62] and expressed as

$$\begin{aligned}A_{x,y}(x_i,y_j)=\int_0^T s\left(t-\frac{2r(t,x,y)}{c}\right)\\ s^*\left(t-\frac{2r(t,x_i,y_j)}{c}\right)e^{-j\frac{4\pi}{\lambda}r(t,x,y)}e^{j\frac{4\pi}{\lambda}r(t,x_i,y_j)}dt\end{aligned} \tag{4.16}$$

The ambiguity function in (4.15) can be considered as a weighting function in each point for the RCS, $\sigma(x,y)$. The range resolution is the first null of $A_{0,0}(0,y) = 0$ and azimuth resolution will come as the first null of $A_{0,0}(x,0) = 0$. Following the simplification assumptions made in [61], range resolution is expressed as

$$\delta_y=\frac{c}{2B\sin\theta} \tag{4.17}$$

and the azimuth resolution is:

$$\delta_x=\frac{\lambda R}{2L}=\frac{L_a}{2} \tag{4.18}$$

in which B is the bandwidth and L_a is the antenna aperture.

Equations (4.17) and (4.18) prove that the same range and azimuth resolutions are achieved with SLAR as the FMCW and pulse-based SARs. The interesting thing about this SLAR technique is that it converts the time (range)-space (azimuth) domain making of the traditional SAR into the 1-D frequency scanning. The other advantage is using a CW signal as the transmitter, which provides a simpler circuitry both for the Tx and Rx. Disadvantages are that this approach is not suitable for nonflat terrain scanning. Any height will result in increased ambitions error. The second problem is moving object detection. As this approach is based on the CW, it is assumed that there is no change in the scene during the scanning time [63], so any availability of any moving objections at that time will end up to the increased final image distortion.

Due to the limitations of the hardware design of Chapter 3, CW-SLAR is the only practical option for this book. The desired image resolution here is 30 mil (0.762 mm) based on the tag dimensions of Chapter 2. Based on (4.17) and (4.18), assuming a $\theta = 45°$, the range and the azimuth resolutions for the antenna in this study [51] are as the following:

$$\delta_y = 3\times10^8 \Big/ \left(2\times7\times10^9\,\frac{1}{2}\right) = 4.3\text{ mm} \tag{4.19}$$

and

$$\delta_x = \frac{1.2\text{ cm}}{2} = 6\text{ mm} \tag{4.20}$$

Equations (4.19) and (4.20) simply mean that a 1 × 1 cm^2 letter tag can only be represented by a shape of 2 × 2 pixels. This is obviously not enough to have a high-resolution image resolution for the tag, but the formed image can be used beside other methods to increase the tag detection success rate.

Using a stop-and-go scanning method, each tag is scanned in a 10 row by 2 column matrix, as shown in Figure 4.22. To create the image of the tag, two types of lab measurements are done: (1) using a one-port VNA, and (2) using the two-port reader developed in Chapter 3. These two approaches are presented in the following sections.

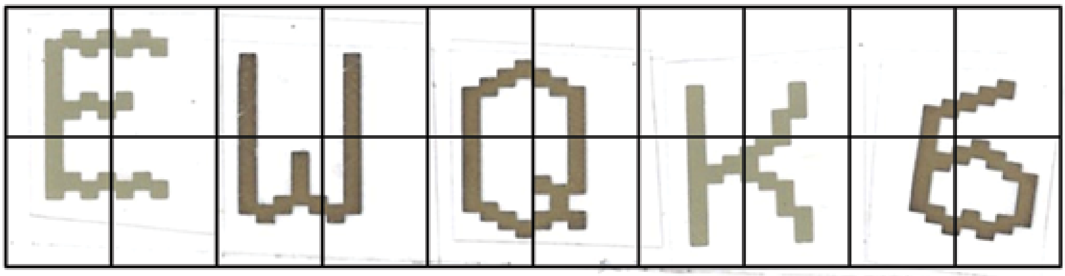


Figure 4.22 Continuous-wave side-looking airborne radar (CW-SLAR) scanning areas for a 5 × 1 cm^2 tag with 10 × 2 scanning regions [57]. (Image courtesy of IEEE.)

4.6.1 One-Port VNA

Figure 4.23 shows tag scanning using a one-port VNA, and Figure 4.24 shows the scanning. In this case, where the scanning of the tag area is done by just one port, (4.13) can be used for VNA measurements, as

$$S_{11}(f) = \frac{S_r(f)}{S(f)} = \frac{\mathcal{F}(s_r(t))}{\mathcal{F}(s(t))} \tag{4.21}$$

where $S_r(f)$ and $S(f)$ are the Fourier transforms of $s_t(t)$ and $s(t)$, respectively. Replacing (4.21) into (4.13) will result in

$$I(x_i, y_j) = \int_0^T \mathcal{F}^{-1}(S_{11}(f)S(f))s^*\left(t - \frac{2r(t, x_i y_j)}{c}\right) \times e^{j\frac{4\pi}{\lambda}r(t, x_i, y_j)}dt \tag{4.22}$$

As $r(t,x_i,y_j)/c$ (c is the speed of the light) in the RFID systems, (4.22) can be rewritten as

$$I(x_i, y_j) = \int_0^T \mathcal{F}^{-1}(S_{11}(f)S(f))s^*(t) \times e^{j\frac{4\pi}{\lambda}r(t, x_i, y_j)}dt \tag{4.23}$$

Assuming $s(t)$ is a series of linear frequency ramps, in discrete time domain it can be expressed as

$$s(t) = \sum_{n=0}^{N-1} e^{j2\pi f_n t}\delta(t-n) \tag{4.24}$$

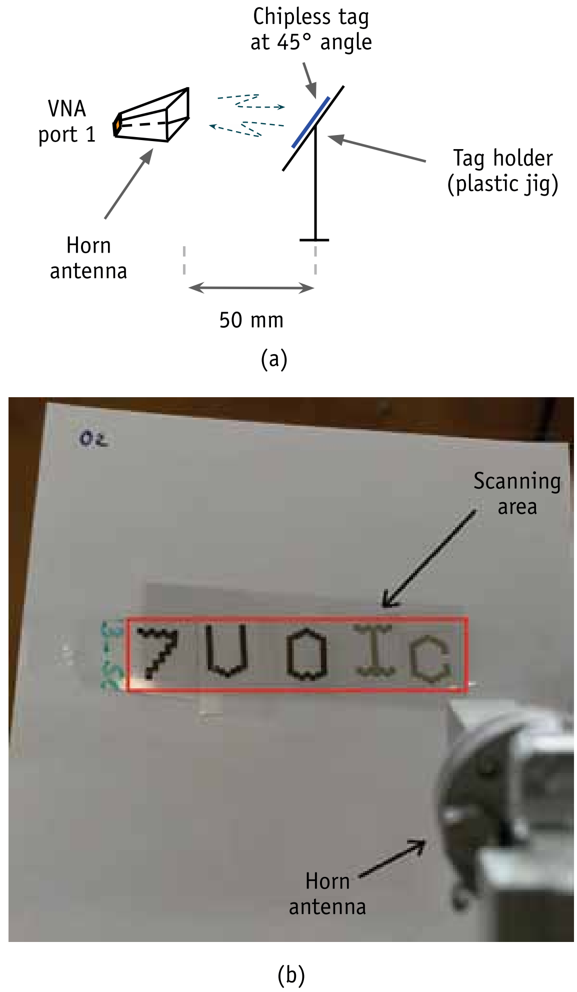


Figure 4.23 SLAR measurement using a one-port VNA. The tag is angled at 45° in front of the antenna in a 50-mm distance [57]. (Image courtesy of IEEE.)

where N is the number of frequency steps in the sweeping time T, hence the $S(f)$ becomes

$$S(t) = \sum_{n=0}^{N-1} 2\pi\delta\left(f - f_n\right) \tag{4.25}$$

where f_n represents each of the frequency steps. Comparing (4.23) and (4.25), the term $S_{11}(f)S(f)$ simply becomes

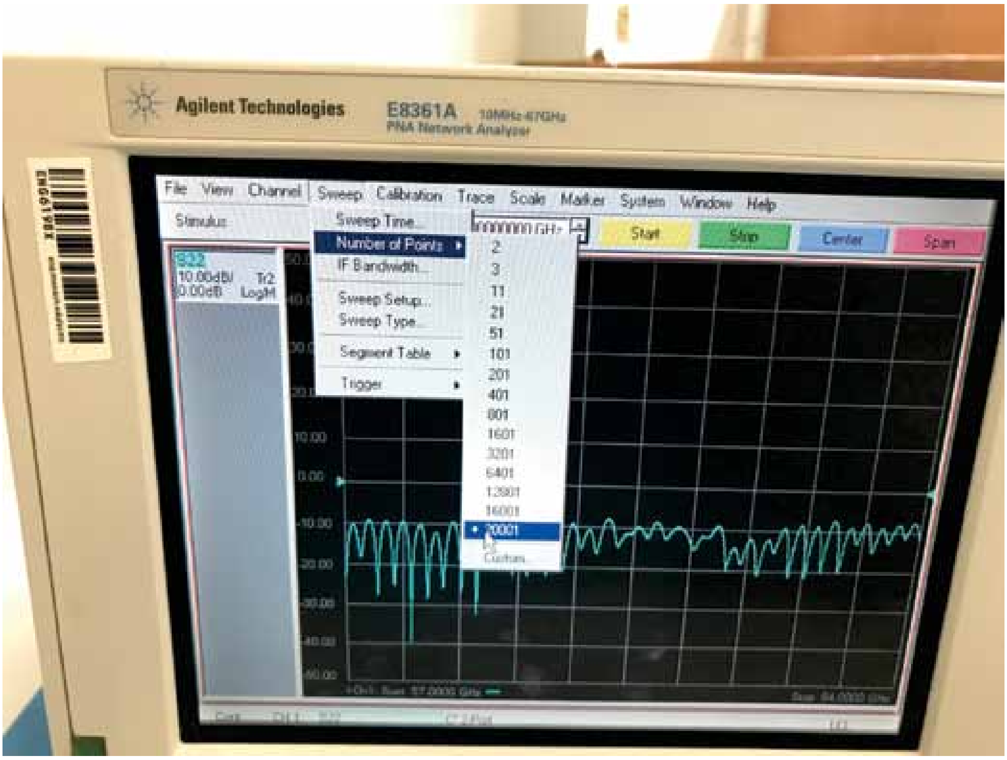


Figure 4.24 A sample VNA outcome of the one-port tag scanning showing the number of sampling points in the sweeping bandwidth.

$$S_{11}S\left(f\right)=S_{11}\sum_{n=0}^{N-1}2\pi\delta\left(f-f_{n}\right)=2\pi S_{11}\quad n=0,\ldots,N-1 \tag{4.26}$$

which is S_{11} sampled over N points, so it is S_{11}. Considering the resolution, for every image section in Figure 4.22, one can consider $x_i = y_j = 0$, as every region can produce one data pixel only. Following the stop-and-go procedure, $r(t,0,0) = R$. Substituting R, (4.24) and (4.25) into the (4.23) provides

$$I\left(0,0\right)\approx 2\pi\int_{0}^{T}\mathcal{F}^{-1}\left(S_{11}\left(f\right)\right)\sum_{n=0}^{N-1}e^{-j2\pi f_{n}t}e^{j\frac{4\pi}{\lambda_{n}}R}\delta\left(t-n\right)dt \tag{4.27}$$

where λ_n is the wavelength corresponding to f_n. Equation (4.27) simply says that for every pixel in the final image, do an inverse Fourier transform on S_{11}, multiply it with a series of $e^{-j2\pi f_n t}$ for the resolution points (N), and integrate over the acquisition time (T). $I(x_i,y_j)$ is changed to $I(0,0)$ as the whole beam footprint is allocated to one point in the center of footprint. As the term $\delta(t - n)$ makes the whole phrase in front of the integral in the (4.27) quantized, one can rewrite the equation as

$$I(0,0) \approx 2\pi \sum_{T} \sum_{n=0}^{N-1} \mathscr{F}^{-1}\left(S_{11}(f)\right) e^{-j2\pi f_n t} e^{j\frac{4\pi}{\lambda_n}R} \delta(t-n) \tag{4.28}$$

4.6.2 Two-Port Reader

In the actual developed reader in Chapter 3, the received signal is $S_r(f)$ instead of the S-parameters of the VNA. The calculations for image construction are quite similar to those in the previous section. Equation (4.23) becomes

$$I\left(x_i, y_j\right) = \int_0^T \mathscr{F}^{-1}\left(S_r(f)\right) s^*(t) \times e^{j\frac{4\pi}{\lambda}r\left(t, x_i, y_j\right)} dt \tag{4.29}$$

Following the same simplification, the final equation of (4.28) becomes

$$I(0,0) \approx 2\pi \sum_{T} \sum_{n=0}^{N-1} \mathscr{F}^{-1}\left(S_r(f)\right) e^{-j2\pi f_n t} e^{j2\pi f_n t} e^{j\frac{4\pi}{\lambda n}R} \delta(t-n) \tag{4.30}$$

which is similar to (4.28).

4.6.3 Computational Costs

One important aspect of image-based tags is their processing power. In this section an approximation is provided to create each pixel in the scene.

To make the calculation faster using the fast Fourier transform (FFT), one can put the number of points in each beamwidth scanning to 2^k, where k is an integer in which 2^k is closest to the number of points in the original signal. From [64], the least number of calculations in $N = 2^k$ FFT is $\frac{39}{4}N \log_2(N)$. The sigma summation in (4.28) has N complex-number calculations, and the integral also has N complex sums, which creates an $N \times N$ matrix with the computation cost of N^2. Having 20 scanning areas per tag (as in Figure 4.22), the total calculation cost is approximately

$$\text{Total calculations} \approx \frac{39}{4} N \log_2(N) * \left(N^2\right) * 20 = 195N^3 \log_2(N) \tag{4.31}$$

The scanning resolution for this book is 20,001 points over the 7-GHz bandwidth of 57–64 GHz using available VNA. Resampling 20,001 points to $2^{14} = 16{,}384$, the approximate calculations per each final pixel is 8.6×10^{14}. This number for a resolution of 100 MHz in the bandwidth (64 points) is 51 million calculations. A trade-off between calculation time and resolution is needed if there is a computation limitation.

4.6.4 Tag Imaging and Experimental Results

Figure 4.25 illustrates the steps needed to create the image of the tag in CW-SLAR technique. The image is made using the 1-D backscatter signal, which includes time delays and Doppler frequency shift [61]. To create the image, the following steps are done:

- The tag is put in the distance of 5 cm with $\theta = 45°$ (beam incident angle in Figure 4.21);
- A range of frequencies from 57 to 64 GHz (BW = 7 GHz) is sent to the tag and backscatter is recorded with 20,001 points (350-KHz resolution);
- The tag is moved in the range direction according to resolution of $\delta_x = 5$ mm;
- The tag is moved in the azimuth direction by $\delta_y = 5$ mm and the same process as above is done;
- The scanning in range and azimuth continues until all parts of the tag are scanned;
- Using (4.28) with T equal to the beam footprint scanning time (600 ms for 20,001 points sampling), calculate $I(x_i = 0, y_j = 0)$ for each footprint area (Figure 4.22);
- Rescale $|I(0,0)|$ to 0 to 255, create an image of 500 × 100 pixel size and put the $|I(0,0)|$ in each 50 × 50 pixels.

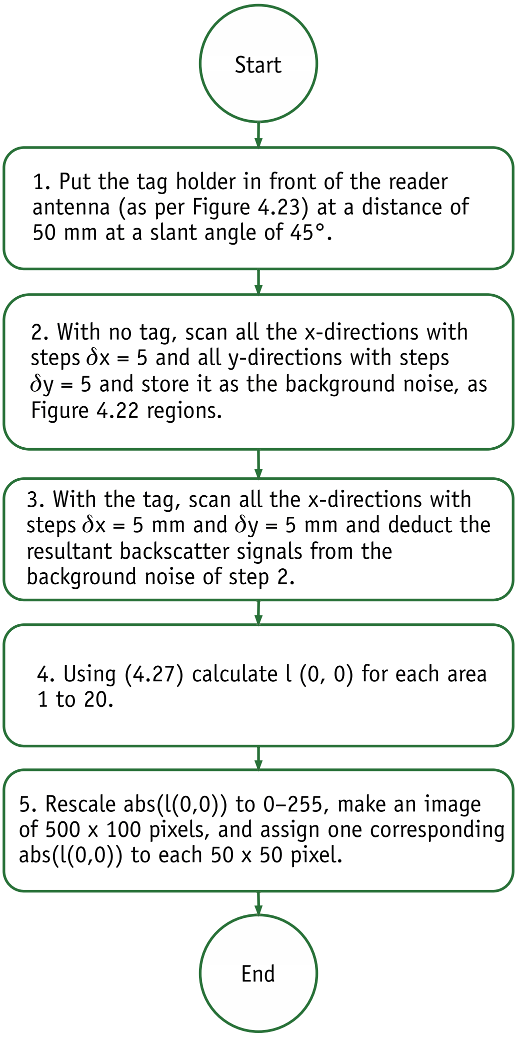
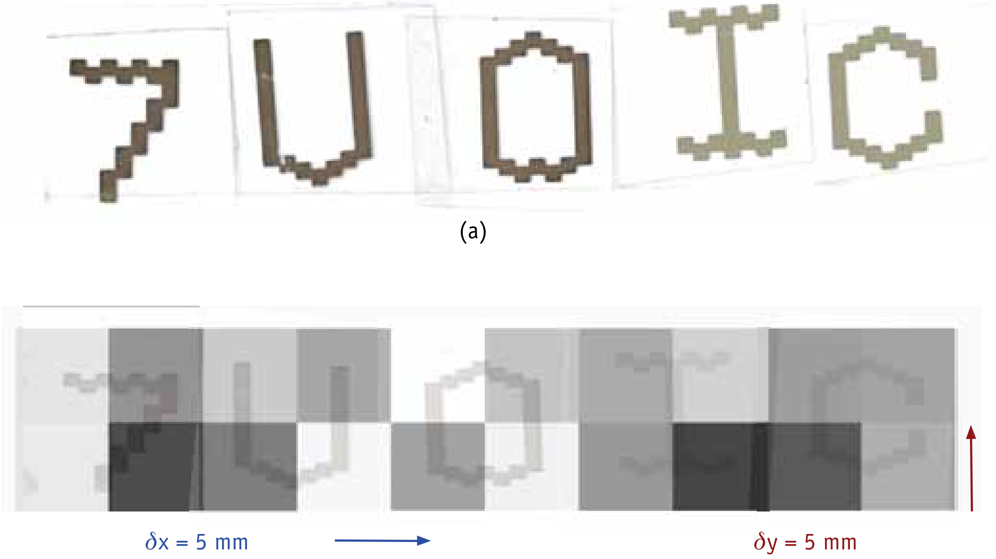

Start
1. Put the tag holder in front of the reader antenna (as per Figure 4.23) at a distance of 50 mm at a slant angle of 45°.
2. With no tag, scan all the x-directions with steps δx = 5 and all y-directions with steps δy = 5 and store it as the background noise, as Figure 4.22 regions.
3. With the tag, scan all the x-directions with steps δx = 5 mm and δy = 5 mm and deduct the resultant backscatter signals from the background noise of step 2.
4. Using (4.27) calculate l (0, 0) for each area 1 to 20.
5. Rescale abs(l(0,0)) to 0–255, make an image of 500 x 100 pixels, and assign one corresponding abs(l(0,0)) to each 50 x 50 pixel.
End

Figure 4.25 Flowchart of creating an image of the tag using the CW-SLAR technique.

Figure 4.26 shows the detected tag using the CW-SLAR technique explained previously. There are a few considerations for this image. First, this is the maximum theoretical resolution available for the tag, as per (4.17) and (4.18). Although the resultant image resolution of 4.26 is low for human sight, there are 255 levels of gray scale available in each block, and each alphanumerical character in the tag is represented by four blocks. This makes distinguishing between different letters easy for a machine. It is worth mentioning that because of measurement difficulties, $\delta_x = \delta_y = 5$



(a)

(b)

Figure 4.26 A plastic tag image reconstruction using the SLAR technique: (a) the original screen-printed plastic tag, and (b) the tag's image created using the CW-SLAR technique. Image size is 500 × 100 pixels. Each letter is represented by four areas. Letters are shown on the tag's background with very low transparency [57]. (Image courtesy of IEEE.)

mm was used instead of actual 6 and 4.3 mm for the azimuth and range calculated resolutions.

Since the tag is now captured as a 2-D image, it is possible to use more sophisticated deep learning methods for decoding. We used Google Cloud for storing the data and machine learning; a more robust way to combine frequency and image data will be discussed in Chapter 5.

4.7 A Reliable Tag Decoder Architecture

As said, one of the main disadvantages of the chipless systems is the lack of verification in the final decoded tag. If both methods of the frequency pattern recognition and the SLAR imaging are combined, a higher reliable system compared to a single method can be created. The overall flow graph for utilizing two parallel systems is illustrated in Figure 4.27. The main steps in the system are:

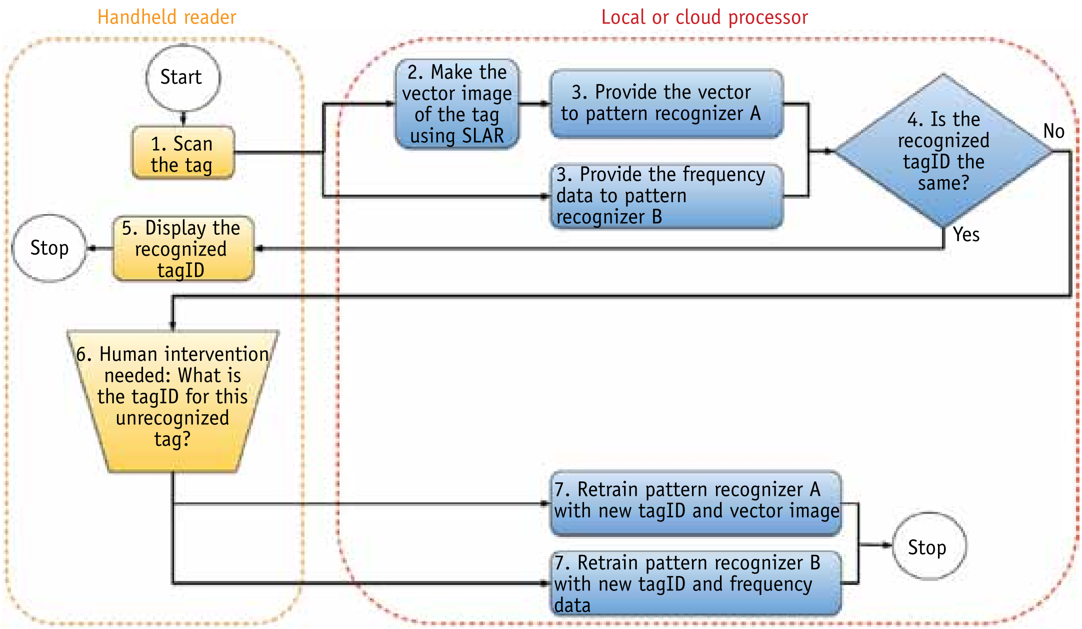

Handheld reader
Local or cloud processor
Start
1. Scan the tag
2. Make the vector image of the tag using SLAR
3. Provide the vector to pattern recognizer A
3. Provide the frequency data to pattern recognizer B
4. Is the recognized tagID the same?
No
Yes
Stop
5. Display the recognized tagID
6. Human intervention needed: What is the tagID for this unrecognized tag?
7. Retrain pattern recognizer A with new tagID and vector image
7. Retrain pattern recognizer B with new tagID and frequency data
Stop

Figure 4.27 Flow graph of the parallel decoder system. Two pattern recognition machines are used at the same time for decoding [57]. (Image courtesy of IEEE.)

- Scan the tag. Using the method described in Section 4.3.2, the tag data is scanned in the whole frequency spectrum and in the required positions. This data is sent to the local or cloud computer, using ESP8266 in the mainboard, as shown in Figures 3.19 and 3.22.
- Create the vector image of the tag. Based on Section 4.6.4, a low-resolution image of the tag is produced. Each tag is basically a vector of 20-pixel values, as per the flow graph in Figure 4.25.
- Provide the data to the pattern recognizer A and B networks. The produced vector image is the input to the pattern recognizer A, and at the same time the frequency-location data is presented to the pattern recognizer B. These two networks are different.
- The recognized tagID from both networks are compared in this stage. If the outcome of the two systems is the same, it means the systems are both trained for this tag, and also the chance of misdetection (i.e., recognizing the wrong tagID) is

very low. The two methods together provide parallel verification of the tagID.

- If the outcome of the previous stage for both recognizers is the same tagID, the tagID is sent back to the handheld reader and the tagID is displayed.
- Human intervention is needed if the recognized tagID in stage 4 is not the same as from the previous stage. The user should confirm the tagID via direct observation.
- Both the pattern recognizers A and B are retrained with the available data from parts 1 and 2. These networks will be ready to correctly recognize the tag next time.

Providing more image samples is quite time consuming as each tag needs to be read 20 times. Human intervention is also needed for the first training of each tag, as well as if the parallel systems give different results. The double verification method proved to be quite robust in the lab experiments. There was no misdetection problem for the available samples observed.

4.8 Conclusions

This chapter presented two methods of decoding for chipless RFID. It was shown that shadow networks such as feedforward and pattern recognizers could effectively be used for high-accuracy decoding tasks. Choosing an appropriate classification algorithm for a particular problem was shown to be a matter of trial and error as each algorithm had its own advantages and disadvantages and was based on particular assumptions and datasets. As Wolpert's proved with his no free lunch (NFL) theorems for optimization, there was actually no single optimization method that performed best around all problems [65].

As a matter of practice, several classifiers were tested and the best-performing ones were chosen for this particular classification problem with the given dataset. The final performance of the classifier depended heavily on underlying training data. To choose the right classifier, these parameters were considered: speed of training, accuracy of the prediction, and AUC. For some models such as KNN, interpretability—meaning how easily the reason behind algorithm is understandable—was easier to explain.

Although pattern recognition and feedforward networks showed very high decoding rates, their false rate was quite problematic. To cover this intrinsic problem of chipless RFID, a SLAR method for making a low-resolution image of the tag was presented. This practical way of using this imaged-based SLAR technique was developed and tested based on the VNA and the reader developed previously. The tags were scanned in a 2-D matrix, and the image was formed based on the CW backscatter received signals and their positions. The actual image resolution was 2 × 10 pixels, but it is shown as 100 × 500 pixels to be visible and comparable to the tag's dimensions. The resulting image resolution was low, but every letter was associated with four different pixels/areas, and each pixel/area had 256 gray-scale levels. This made the image of each letter distinguishable. The resultant image from SLAR was used as a vector to train a parallel decoding system in conjunction with past AI decoding methods to create a more highly reliable chipless decoder. The proposed double verification method was quite robust in the lab experiments.

There was no misdetection problem for the available samples observed with our limited number of tags.

4.9 Chipless Tag Decoding Questions and Answers

4.1. Can I replicate the results of machine learning in this chapter myself?

You are highly encouraged to replicate the experiments done in this chapter.

- For the feedforward network in Section 4.4, collected data is from the Computer Simulation Technology (CST) Studio Suite, in which a few tags are put in front of a cross-polarized Tx/Rx reader and the tags are moved in a linear path perpendicular to the propagation wave direction, as per Figure 4.9. The dataset can be found in [52]. We used MATLAB's Neural Network Fitting app for the feedforward network training.

- For the pattern recognition results (Section 4.5.1) the dataset is from [56] and MATLAB's Classification Learner is used. The output parameters such as training time, AUC, and accuracy alongside input settings are collected manually and the graphs are created within Google Sheets and Microsoft Excel.

4.2 How do I make 2-D SLAR images of the tag?

It depends on the aperture of the antenna in your experiments. In our case, with an antenna aperture of $D = 10$ mm, the maximum resolution is $D/2 = 5$ mm, and scanning for each 5×5 mm^2 of the tag's area should be done at 45°, as per Figures 4.22 and 4.23. Using (4.27), one $I(0,0)$ illumination point will be calculated for each scanning area. So for a tag of 1×5 cm^2, 20 scans and 20 illuminations will be made. Those point illuminations will be converted to image areas as shown in Figure 4.26(b).

4.3 Will it be time consuming to make an image?

Yes, in our case, 20 individual scans are needed for a 1×5 cm^2 tag, and there is a bit of processing time to display the image.

4.4 What is the limit of the number of different tags in the ML or pattern recognition methods?

Pattern recognition methods are based on the different *distinguishable* frequency signatures of the tag space. The more tags, the higher the chance that the pattern recognition outcome is a misdetection. In our experiments, we avoided highly closed pattern tags, such as "ASD0K" and "ASDOK" (letter "O" and number zero). We have not gone beyond our sample space of 27 tags. In an average grocery store, the total different items can be above 15,000, so the tag space should be chosen carefully.

References

[1] Google, Google AI: Machine Learning Products, https://cloud. google.com/ml-engine/docs/pricing.

[2] Vu-Quoc, P. L., *Neuron and Myelinated Axon, with Signal Flow from Inputs at Dendrites to Outputs at Axon Terminals*, 2018, https://en.wikipedia.org/wiki/Biological_neuron_model#/media/File:Neuron3.png.

[3] Chauvin, Y., and D. E. Rumelhart, *Backpropagation: Theory, Architectures, and Applications,* Philadelphia: Psychology Press, 2013.

[4] Heermann, P. D., and N. Khazenie, "Classification of Multispectral Remote Sensing Data Using a Backpropagation Neural Network," *IEEE Transactions on Geoscience and Remote Sensing,* Vol. 30, No. 1, 1992, pp. 81–88, doi:10.1109/36.124218.

[5] Nielsen, M., *Neural Networks and Deep Learning,* Determination Press, 2015, http://neuralnetworksanddeeplearning.com.

[6] Cybenko, G., "Approximations by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, Vol. 2, 1989, pp. 183–192.

[7] Leshno, M., V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function," *Neural Networks*, Vol. 6, No. 6, 1993, pp. 861–867.

[8] Catal, C., S. Tufekci, E. Pirmit, and G. Kocabag, "On the Use of Ensemble of Classifiers for Accelerometer-Based Activity Recognition," *Applied Soft Computing*, Vol. 37, 2015, pp. 1018–1022.

[9] Wang, C., Z. Shi, and F. Wu, "An Improved Particle Swarm Optimization-Based Feed-Forward Neural Network Combined with RFID Sensors to Indoor Localization," *Information*, Vol. 8, No. 1, 2017, p. 9.

[10] Selvan, P. T., and S. Raghavan, "Neural Network Model for Design of Compact CPW-Fed Monopole Antenna for 5.8 GHz RFID Application," in *2010 Second International Conference on Computing, Communication and Networking Technologies*, IEEE, 2010, pp. 1–5.

[11] Kuo, R., and J. Chang, "Intelligent RFID Positioning System through Immune-Based Feed-Forward Neural Network," *Journal of Intelligent Manufacturing*, Vol. 26, No. 4, 2015, pp. 755–767.

[12] Kim, S. Y., S. Kim, J. Cho, et al., "A Deep Learning Model for Real-Time Mortality Prediction in Critically Ill Children," *Critical Care*, Vol. 23, No. 1, 2019, p. 279.

[13] Cortes, C., and V. Vapnik, "Support-Vector Networks," *Machine Learning*, Vol. 20, No. 3, 1995, pp. 273–297.

[14] Vapnik, V., *Statistical Learning Theory,* New York: Wiley, 1998.

[15] MathWorks, Support Vector Machines for Binary Classification, https://au.mathworks.com/help/stats/support-vector-machines-for-binaryclassification.html.

[16] Amendola, S., R. Lodato, S. Manzari, C. Occhiuzzi, and G. Marrocco, "RFID Technology for IoT-Based Personal Healthcare in Smart Spaces," *IEEE Internet of Things Journal*, Vol. 1, No. 2, 2014, pp. 144–152.

[17] Yamano, K., K. Tanaka, M. Hirayama, E. Kondo, Y. Kimuro, and M. Matsumoto, "Self-Localization of Mobile Robots with RFID System by Using Support Vector Machine," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, IEEE, Vol. 4, 2004, pp. 3756–3761.

[18] Jo, M., H. Y. Youn, S.-H. Cha, and H. Choo, "Mobile RFID Tag Detection Influence Factors and Prediction of Tag Detectability," *IEEE Sensors Journal*, Vol. 9, No. 2, 2009, pp. 112–119.

[19] Zhang, S., "Nearest Neighbor Selection for Iteratively kNN Imputation," *Journal of Systems and Software*, Vol. 85, No. 11, 2012, pp. 2541–2552.

[20] Li, B., and L. Han, "Distance Weighted Cosine Similarity Measure for Text Classification," in *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, 2013, pp. 611–618.

[21] Xu, H., Y. Ding, P. Li, R. Wang, and Y. Li, "An RFID Indoor Positioning Algorithm Based on Bayesian Probability and K-Nearest Neighbor," *Sensors*, Vol. 17, No. 8, 2017, p. 1806.

[22] Han, K., and S. H. Cho, "Advanced LANDMARC with adaptive k-Nearest Algorithm for RFID Location System," in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, IEEE, 2010, pp. 595–598.

[23] Huiting, J., H. Flisijn, A. B. Kokkeler, and G. J. Smit, "Exploiting Phase Measurements of EPC Gen2 RFID tags," in *2013 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, IEEE, 2013, pp. 1–6.

[24] Caragea, D., A. Silvescu, and V. Honavar, "A Framework for Learning from Distributed Data Using Sufficient Statistics and Its Application to Learning Decision Trees," *International Journal of Hybrid Intelligent Systems*, Vol. 1, No. 1-2, 2004, pp. 80–89.

[25] Magerman, D. M., "Statistical Decision-Tree Models for Parsing," in *Proceedings of the 33rd annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 1995, pp. 276–283.

[26] Wikipedia, "Ensemble Learning," July 2022, https://en.wikipedia.org/wiki/Ensemble_learning.

[27] Yang, Z., P. Zhang, and L. Chen, "RFID-Enabled Indoor Positioning Method for a Real-Time Manufacturing Execution System Using OS-ELM," *Neurocomputing*, Vol. 174, 2016, pp. 121–133.

[28] Deng, L., "A Tutorial Survey of Architectures, Algorithms, and Applications for Deep Learning," *APSIPA Transactions on Signal and Information Processing*, Vol. 3, 2014, pp. 1–29, doi:10.1017/atsip.2013.9.

[29] Krizhevsky, A., I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[30] Deng, L., G. Hinton, and B. Kingsbury, "New Types of Deep Neural Network Learning for Speech Recognition and Related Applications: An Overview," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 8599–8603.

[31] Unlu, E., E. Zenou, N. Riviere, and P.- E. Dupouy, "An Autonomous Drone Surveillance and Tracking Architecture," *Electronic Imaging*, Vol. 2019, No. 15, 2019, pp. 35–1.

[32] Akhilesh, K., "Smart Technologies—Scope and Applications," in *Smart Technologies*, Springer, 2020, pp. 1–16.

[33] Why Caffe? https://caffe.berkeleyvision.org.

[34] Jia, Y., E. Shelhamer, J. Donahue, et al., "Caffe: Convolutional Architecture for Fast Feature Embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[35] Team, T. T. D., R. Al-Rfou, G. Alain, et al., "Theano: A Python Framework for Fast Computation of Mathematical Expressions," *arXiv preprint arXiv:1605.02688*, 2016.

[36] Abadi, M., P. Barham, J. Chen, et al., "TensorFlow: A System for Large-Scale Machine Learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, USENIX Association, 2016, pp. 265–283, https://www.usenix.org/conference/osdi16/technicalsessions/presentation/abadi.

[37] Keras: Deep Learning for Humans, 2022, https://keras.io.

[38] Li, X., Y. Zhang, I. Marsic, A. Sarcevic, and R. S. Burd, "Deep Learning for RFID-Based Activity Recognition," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, ACM, 2016, pp. 164–175.

[39] Umbricht, D., W.-Y. Cheng, F. Lipsmeier, A. Bamdadian, P. Tamburri, and M. Lindemann, "Deep Learning-Based Human Activity Recognition for Continuous Activity and Gesture Monitoring for Schizophrenia Patients with Negative Symptoms," *Frontiers in Psychiatry*, Vol. 11, 2020, p. 574375.

[40] Wang, T., Y. He, B. Li, and T. Shi, "Transformer Fault Diagnosis Using Self-Powered RFID Sensor and Deep Learning Approach," *IEEE Sensors Journal*, Vol. 18, No. 15, 2018, pp. 6399–6411.

[41] Wang, C., and P. Jiang, "Deep Neural Networks Based Order Completion Time Prediction by Using Real-Time Job Shop RFID Data," *Journal of Intelligent Manufacturing*, Vol. 30, No. 3, 2019, pp. 1303–1318.

[42] Yuan, Y., Y. Shi, C. Li, et al., "Deepgene: An Advanced Cancer Type Classifier Based on Deep Learning and Somatic Point Mutations," *BMC Bioinformatics*, Vol. 17, No. 17, 2016, p. 476.

[43] Nie, L., A. Kumar, and S. Zhan, "Periocular Recognition Using Unsupervised Convolutional RBM Feature Learning," in *2014 22nd International Conference on Pattern Recognition*, IEEE, 2014, pp. 399–404.

[44] Irsoy, O., and C. Cardie, "Deep Recursive Neural Networks for Compositionality in Language," in *Advances in Neural Information Processing Systems*, 2014, pp. 2096–2104.

[45] Zhang, L., S. Wang, and B. Liu, "Deep Learning for Sentiment Analysis: A Survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 8, No. 4, 2018, e1253.

[46] O'Boyle, B., *What Is Siri? Apple's Personal Voice Assistant Explained*, 2015.

[47] Che, Z., S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," *Scientific Reports*, Vol. 8, No. 1, 2018, pp. 1–12.

[48] Shi, X., Y. Hao, D. Zeng, et al., "Cloud-Assisted Mood Fatigue Detection System," *Mobile Networks and Applications*, Vol. 21, No. 5, 2016, pp. 744–752, doi:10.1007/ s11036-016-0757-x.

[49] Hazelwood, K., S. Bird, D. Brooks, et al., "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, IEEE, 2018, pp. 620–629.

[50] Arjomandi, L. M., and N. C. Karmakar, "An Enhanced Chipless RFID System in 60 GHz Using Pattern Recognition Techniques," in *2018 48th European Microwave Conference (EuMC)*, 2018, pp. 973–976, doi:10.23919/ EuMC.2018.8541656.

[51] A-Info, *LB-15-10 50.0-75.0 GHz Standard Gain Horn Antenna*.

[52] Arjomandi, L. M., "Chipless Tag Movement in Front of the Cross Polarized Reader in a Linear Rail, Simulation Data For 6 Tags in mm-Wave Spectrum," July 2022, bridges.monash.edu, doi:10.26180/20390031.v1.

[53] Mohammed, M., M. B. Khan, and E. B. M. Bashier, *Machine Learning: Algorithms and Applications*, Taylor & Francis, 2016.

[54] Shaikh, R., *Feature Selection Techniques in Machine Learning with Python*, 2018, https://towardsdatascience.com/feature-selection-techniquesin-machine-learning-with-python-f24e7da3f36e.

[55] MathWorks, *Feature Selection, Identify Influential Features to Improve Model Performance,* https://au.mathworks.com/discovery/feature-selection.html.

[56] Arjomandi, L., *S11 Parameters for 27 Tags at 3 Tag-to-Reader Positions (Left-Middle-Right)*, 2021, https://dx.doi.org/10.21227/ me2h-t256.

[57] Arjomandi, L. M., G. Khadka, and N. C. Karmakar, "mm-Wave Chipless RFID Decoding: Introducing Image-Based Deep Learning Techniques," *IEEE Transactions on Antennas and Propagation*, Vol. 70, No. 5, 2021, pp. 3700–3709.

[58] Tian, Y., S. Guo, and W. Yu, "Stop-and-Go Error Analysis of geosynchronous SAR," *Science China Information Sciences,* Vol. 59, No. 6, 2016, p. 062306, doi:10.1007/s11432-015-5388-5.

[59] Yang, J., *Study on Ground Moving Target Indication and Imaging Technique of Airborne SAR*, Springer Singapore, 2017, ISBN: 9789811030758.

[60] Zhu, X., "Spectral Estimation for Synthetic Aperture Radar Tomography," master thesis, Technische Universität München, Earth Oriented Space Science and Technology–ESPACE, 2008.

[61] Nan, Y., X. Huang, and Y. J. Guo, "Generalized Continuous Wave Synthetic Aperture Radar for High Resolution and Wide Swath Remote Sensing," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 56, No. 12, 2018, pp. 7217–7229, doi:10.1109/TGRS.2018.2849382.

[62] Jankiraman, M., *Design of Multi-Frequency CW Radars,* SciTech Publishing, 2007, Vol. 2, DOI:10.1049/SBRA004E_ch3.

[63] Meta, A., and P. Hoogeboom, "Time Analysis and Processing of FM-CW SAR Signals," in *Proc. IRS*, 2003, pp. 263–268.

[64] Lundy, T., and J. Van Buskirk, "A New Matrix Approach to Real FFTs and Convolutions of Length 2k," *Computing*, Vol. 80, No. 1, 2007, pp. 23–45.

[65] Wolpert, D. H., and W. G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 1997, pp. 67–82.

5

Cloud-Based Deep Learning

5.1 Introduction

The RFID reader system developed in this book is based on AI, and it needs to handle and process a huge amount of data when the data volume increases with the number of tags. The cost of maintaining the data, making efficient backups, and maintaining the hardware and security issues can be quite high if it is considered just for one reader. If more than one reader is involved, the collected data and the trained AI recognition network need to be shared between devices, enabling them to recognize the tags scanned from other devices without doing the time-consuming scanning process over and over again. This is where a *shared* backend processing unit for all the readers is desirable. This shared processing unit can be simply a remote server or a cloud computing service.

This chapter describes how the available state-of-the-art cloud computing facilities are used to realize the deep learning methods using Keras and Tensorflow backend languages [1] for chipless tag decoding. Initially, the total cloud connection and architecture is discussed using infrastructure-as-a-service (IaaS) or SaaS models. Having enough data is essential for training deep learners, so in one section the method for dataset preparation is described, which includes the data augmentation technique used.

The 2-D deep learning method is an enhanced version of the pattern recognizers introduced in Chapter 4. In the dataset preparation part, a novel 2-D chipless RFID data representation and augmentation method is described. This is the first time a deep learning technique is used in the chipless RFID domain.

This chapter is organized as follows (Figure 5.1): After this introduction, important aspects of cloud computing, such as security and costs, are discussed in Section 5.2. In Section 5.3, the details of preparing the hardware model for this project are discussed. Section 5.4 illustrates an effective way of creating augmented images based on available 2-D frequency scans of the tags, followed by the structure of the CNN networks used and their results. Section 5.5 explains how the RFID reader can interact with cloud processing facilities to make a more reliable system. The chapter ends with conclusions.

5.2 Cloud Computing Considerations

The cloud is revolutionizing the information technology (IT) industry in many ways. Cloud computing provides a model for pervasive and on-demand access to some shared group of comput-

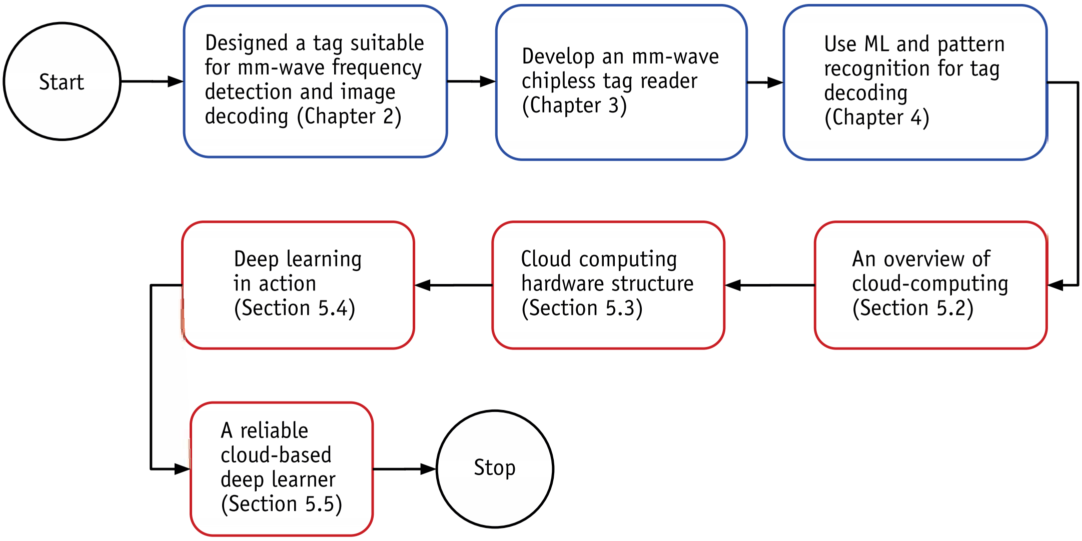

Start
Designed a tag suitable for mm-wave frequency detection and image decoding (Chapter 2)
Develop an mm-wave chipless tag reader (Chapter 3)
Use ML and pattern recognition for tag decoding (Chapter 4)
An overview of cloud-computing (Section 5.2)
Cloud computing hardware structure (Section 5.3)
Deep learning in action (Section 5.4)
A reliable cloud-based deep learner (Section 5.5)
Stop

Figure 5.1 An overview of Chapter 5. The material in the red boxes will be covered in this chapter.

ing resources (such as storage, servers, applications, and services) with quicker release times and less management efforts [2]. Cloud computing provides increased versatility, scalability, data security, and dramatically reduced costs because of cloud hardware sharing [3–5]. In this book, cloud computing data-sharing makes it possible for different readers to have access to a tag's data from other readers. Any improvement in the detection algorithm can be implemented directly on the server, and clients always get the benefit of having the latest data from all readers and improved detection techniques. Cloud computing also removes the need for data synchronization between local hosts, thus dramatically reducing the cost per reader unit.

This book focuses on an AI-based reader, which deals with several million rows of data. Cloud computing has the following advantages for this AI-based reader over a traditional local processor reader:

Disaster recovery and preventing downtimes are easier. If one of the tag readers fails, no data is lost as no data is kept locally in the reader side. Data loss disaster is unlikely to happen in the cloud as data in the cloud is kept in several servers around the world [6].

The cost of cloud usage is shared between all the chipless tag readers using it. Initial setup cost of a cloud server is not high (around 50 cents/hour for Amazon Elastic Computer Cloud (EC2) m4.xlarge [7] or Google large_model ML-Engine [8]). The more readers used, the lower the costs of cloud computing per unit would be. In this book, only the free tiers of the cloud servers are used.

One central storage location. All the scanned tag data is stored in one place, which gives the recognition algorithm a chance to train/recognize with different data coming from different readers.

Higher security can be preserved. Security is a matter of serious concern, especially for big data or if data sensitivity and confidentiality is important. It is much easier to control clients' access rights to the data using a cloud data storage model. Three concerns of security are:

- Internal data theft or altering data can be prevented. A staggeringly high percentage of data thefts occur internally by local employees.

- It is claimed that 94% of businesses noticed an improvement in the security after switching to the cloud. The main key to this added security is the data encryption during transmission over networks (secured protocols) and data encryption in databases [3].
- Unlike local hosts, maintaining security (software, patches, updates) is not a matter of concern in cloud computing, especially in the SaaS model.

Scalability, or the flexibility with the growth of the data/tags/readers is easily preserved. All that is needed is to upgrade the cloud service to a higher computational or storage model.

Insight into the data is possible. Having all the data in one server makes it possible to use integrated cloud analysis, establish tracking mechanism, or build customized reports for different organizations [4].

Sustainability and green environment considerations are preserved. With more demands from local councils and governments for greener and more sustainable workplaces, shared cloud computing is a satisfactory solution. Customers consume 77% fewer servers, 84% less power, and reduce carbon emissions by 88% by using the cloud, so there is no doubt that using the cloud has a positive impact on the environment [5].

Cloud computing in general may be grouped into three categories: (1) SaaS, (2) PaaS, and (3) IaaS [9]. SaaS, which is also referred to as application-as-a-service, is a software licensing and delivery model in which software is shared and licensed on a subscription basis and is hosted centrally (like Google Apps). In PaaS, the whole platform (Windows, Linux, etc.) is shared and enables customers to develop, run, and manage applications without getting involved in the maintenance complexities. IaaS is the most flexible cloud computing model and enables customers to choose servers, processing power, storage, and networking. Although IaaS might be run on virtual environments, customers have much more control over their infrastructure compared to the clients of PaaS or SaaS [9]. A comparison of the three cloud computing services is illustrated in Figure 5.2.

Container-as-a-service technology is another lightware version of cloud computing, especially for IoT applications. A container

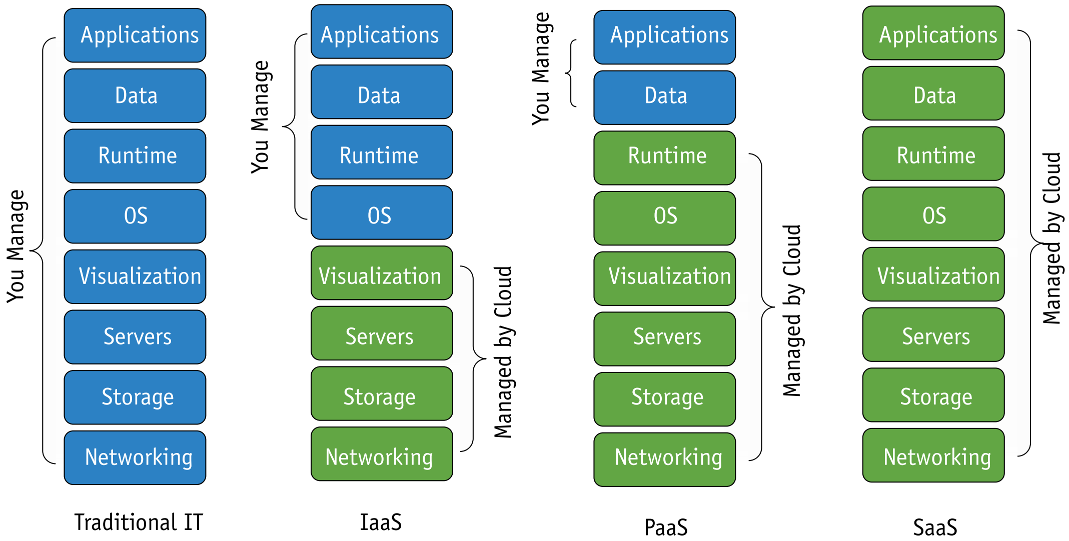


Figure 5.2 Common cloud services in use. In this book, IaaS and SaaS are implemented.

is like a sandbox, a method to package a cloud application so it can be run, with its dependencies, isolated from other processes. Amazon Web Services, Microsoft Azure, and Google Cloud Platform have their own container software names, such as Docker, Apache Mesos, rkt (rocket), and Kubernetes [10, 11]. Container technology takes its name from the shipping industry, in which instead of using a unique way to ship each product, goods are placed into different containers and are delivered to the destination docks. Adoption of PaaS services continues to increase with container-as-a-service now reaching second place behind database-as-a-service [12]. Container challenges are illustrated in Figure 5.3.

5.2.1 Cloud Computing Challenges

Cloud computing has its challenges as well:

Data breaches. Despite its architecture, cloud data is, in general, three times more likely to have data breaches compared to local storage [2]. This is achieved by a man-in-the-cloud attack. Possible solutions to this attack are regular staff training, using data encryption to make stolen data useless to the attacker, enabling multifactor authentication (MFA) as an extra layer of security, and using a

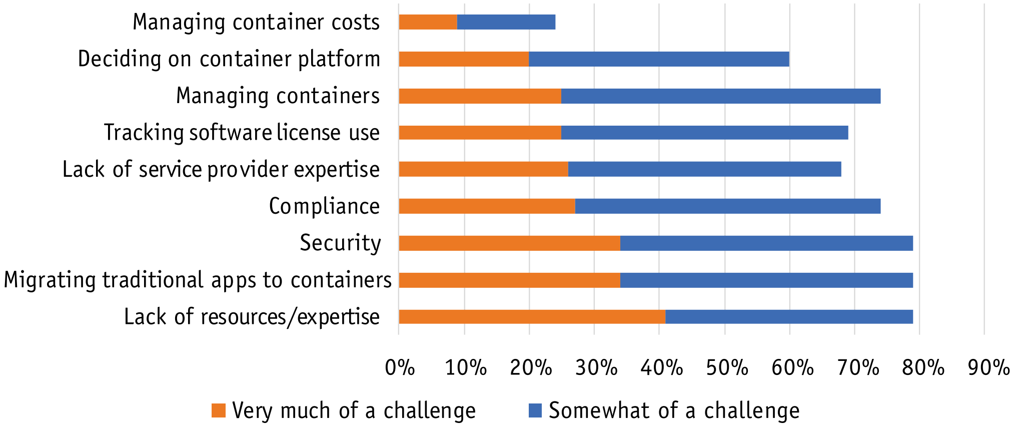


Figure 5.3 Top challenges of container use.

cloud access security broker (CASB) method to intermediate the authentication tokens for apps with the encrypted ones [13].

Hijacking accounts. Hijacking is a common problem once attackers have login information for cloud-based users. Staff training, access rights control, and MFA are the best ways to prevent this security threat.

Malware injection. Malware injections are scripts that can be embedded into cloud services. Once malware injection occurs, user data integrity will be compromised.

Data abuse. Stored data in the cloud can be violated by legal authorities. Terms of agreement should always be considered before deploying any public cloud data for sensitive information.

Denial-of-service attack. Denial-of-service happens once the attacker prevents access to the cloud service but deploys many zombie computers to request access to the destination service at the same time. Denial-of-service attacks are not a breach of the data challenge, but they should be avoided in the early stages [14].

Insider attack. An insider attack is most probably the worst type of attack, in which an authorized employee uses cloud services to exploit and misuse critical information.

Data loss. Although less unlikely for SaaS, there is a chance of data loss in the cloud, especially for IaaS and PaaS. Data loss protection is the shared liability of the cloud user and cloud service provider.

Figure 5.4 illustrates the main contribution factors to cloud-based data loss. Hardware failure is the most important one for data loss in IaaS and PaaS, but normally it is not a concern in SaaS as the main cloud providers have backups around the globe [2].

Figure 5.5 illustrates the top challenges of deploying the cloud, according to feedback from 750 professionals and organizations. Security is still the main concern, followed by cost management of cloud utilization. An IT department may not be the only entity with full control over the provisioning and operations of the infrastructure of the cloud, so governance and control of the cloud is a challenge. Compliance with industry regulations and laws is also challenging once the data is placed in the cloud, especially for public and health records. Some companies use multiple clouds or a combination of private and public clouds, which makes data management difficult. And, finally, managing licenses for different apps in the cloud is a challenge, as cloud resource availability should match the demand in the organization [15].

5.3 Cloud Computing Hardware Architecture

This section explains the hardware implementation part of the cloud-based server used in this book. Figure 5.6 illustrates the proposed cloud computing method using IaaS. After the tags are scanned and processed by the MCU, the data is sent through Wi-Fi or Ethernet interfaces to the cloud. Two cloud services are used in this book, the National eResearch Collaboration Tools and Re-

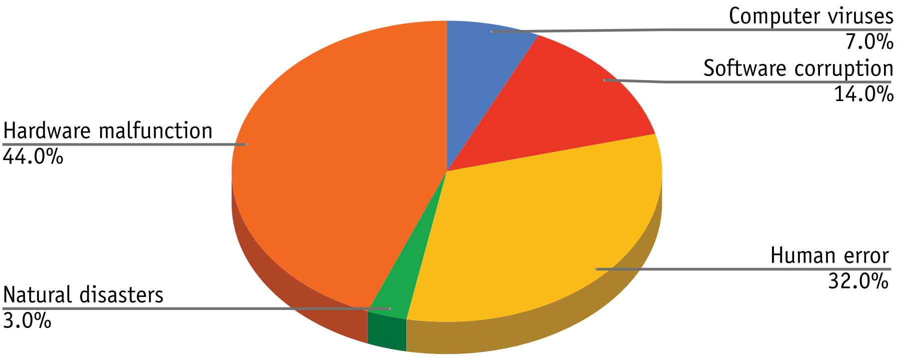


Figure 5.4 Common causes of cloud data loss.

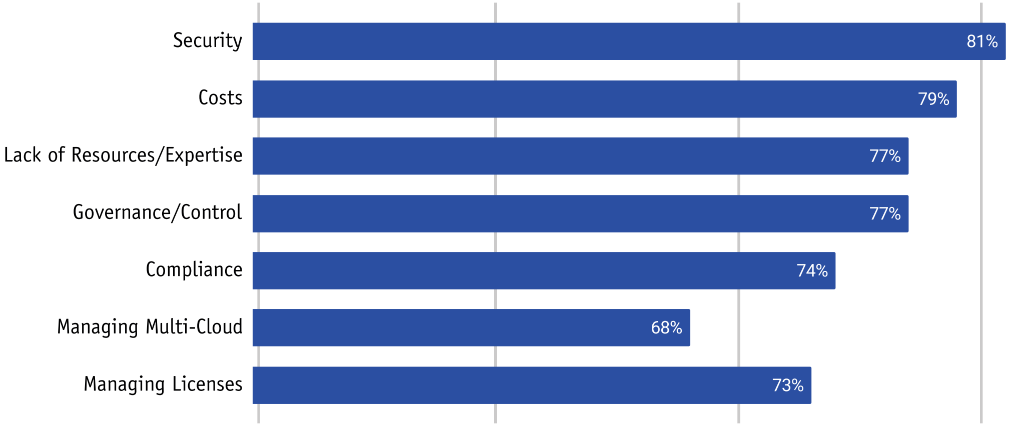


Figure 5.5 Top cloud challenges.

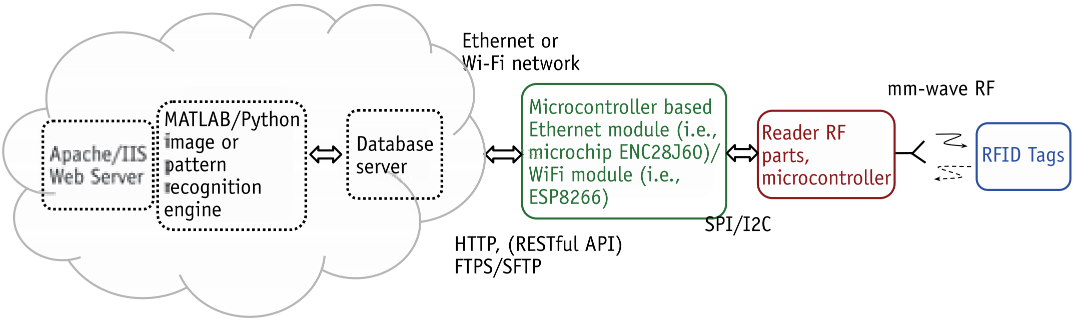

Apache/IIS Web Server
MATLAB/Python image or pattern recognition engine
Database server
Ethernet or Wi-Fi network
Microcontroller based Ethernet module (i.e., microchip ENC28J60)/ WiFi module (i.e., ESP8266)
HTTP, (RESTful API) FTPS/SFTP
SPI/I2C
Reader RF parts, microcontroller
mm-wave RF
RFID Tags

Figure 5.6 Cloud computing architecture based on an IaaS model. Hosting the VM and two-way communications between the reader and the cloud host are the biggest challenges here. Communication between the reader and the cloud needs a corresponding server in the cloud: web-sever, File Transfer Protocol (FTP) server or Secure Shell Protocol (SSH) sever for the HTTPS, and File Transfer Protocol Secure (FTPS) or SSH File Transfer Protocol (SFTP) communications, respectively.

sources (NECTAR) [16] IaaS cloud server and the Google Colab SaaS.

5.3.1 IaaS Model

Figure 5.6 shows this book's first attempt at utilizing the cloud using IaaS. NECTAR has a self-service structure with a scalable computing power to all Australian researchers. NECTAR's structure is quite similar to the structure of the Amazon EC2 [7]. Within that NECTAR virtual machine, an Ubuntu instance is chosen to install MATLAB/Python and the SQL server software. The Microsoft

Structured Query Language (MS-SQL) provides enough database capabilities for storing a lot of tag frequency data as well as categorized permissions for possible different clients.

Chipless tags are scanned by the reader in the mm-wave spectrum, and a microcontroller sends the digitized magnitude, phase, and location of the tag to the Wi-Fi module, ESP8266. The Wi-Fi modules wrap this data into a JSON or Extensible Markup Language (XML) format and send it through Hypertext Transfer Protocol Secure (HTTPS) (PUT/POST in RESTful commands) to the Ubuntu VM provided by NECTAR. The received files are stored in a folder in the cloud host, and a scheduled Data Transformation Services (DTS) transfers this data to the Structured Query Language (SQL). MATLAB/Python use the new data in the SQL to retrain their 1-D (pattern recognition) or 2-D (image-based) learners. Once done, the result (like the decoding outcome, the probability) will be put in a file in the cloud server, and the reader will read the file (GET in RESTful commands). NECTAR can be replaced by an Amazon Web Services (AWS) EC2, a Microsoft Azure VM, or a Google Compute Engine.

Figure 5.6 explains the functionality of the system. This is the first cloud architecture used in the project.

The advantage of this IaaS NECTAR cloud configuration is full access to the Ubuntu VM. Disadvantages are the real complexity of communications between the reader and the server, and maintaining the virtual machine and the scheduled programs within the provided host. NECTAR has a shared processing architecture, and it is best suited for the programs that need a certain processing power in a predefined time. NECTAR suspends the inactive tasks VMs, so better hosting is needed for longer time considerations.

5.3.2 SaaS Model

Because of the complexity of the IaaS, we decided to go with a simpler solution. This SaaS model uses Jupyter Notebooks and Google Colab. The Jupyter Notebook is an interactive environment for programming a few languages, including Python and R [17]. Since its popularity beginning in 2014, the major cloud providers have adopted the Jupyter Notebook or its derivative tools as a front-end interface for their cloud usages. These include Amazon's

SageMaker Notebooks, Google's Colab, and Microsoft's Azure Notebook, to name a few [18].

As said, Google Colab is a cloud-based data science workspace, which has many similarities to the Jupyter Notebook. The backend server for each Google Colab session is a virtual machine running 13 GB of RAM on either a CPU, a GPU, or a Tensor Processing Unit (TPU) processor. The TPU is a Google proprietary ASIC specifically designed for neural network machine learning using Tensorflow language.

In this SaaS implementation, a direct connection to the Google services is needed. This is why a local PC (or a Raspberry Pi) is used in the last stage of the hardware in Figure 5.7. As with IaaS

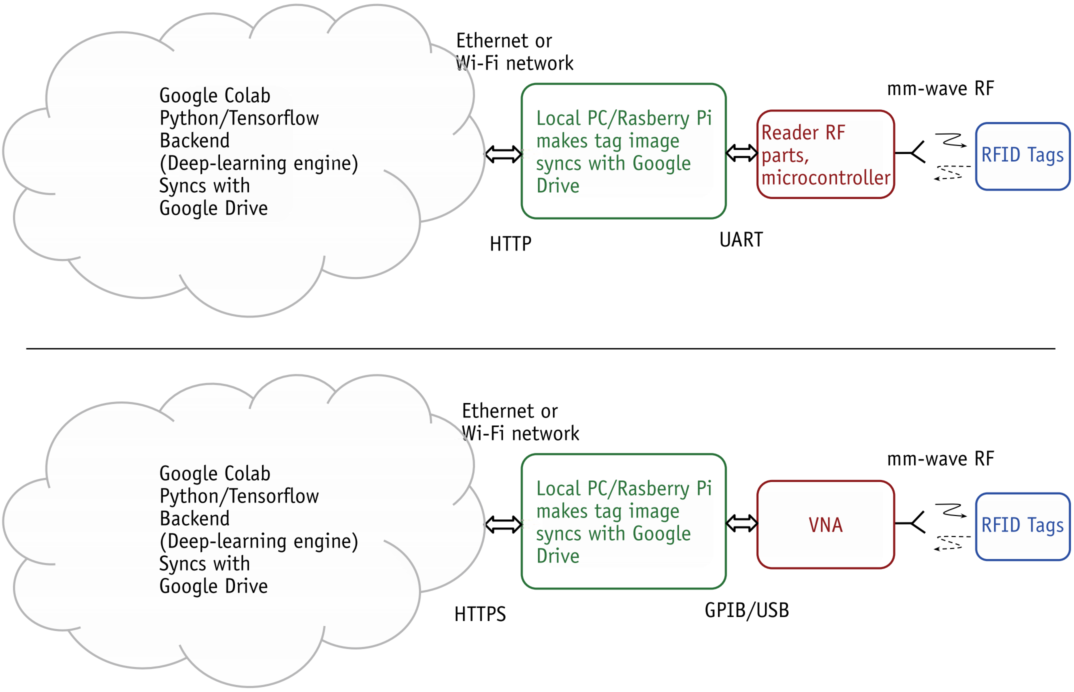


Figure 5.7 Cloud computing architecture based on Google SaaS. Top: the developed reader sends the digitized tag response data to the local PC using the UART protocol, in which the tag's SLAR image is created and the data is then synced with Google Drive. The data is processed by Google Colab using the Keras/Tensorflow programming languages, and the results are synced with Google Drive so that the local PC may view them. Bottom: the reader hardware can be replaced by a VNA that sends the data using its Universal Serial Bus (USB) or General Purpose Interface Bus (GPIB) port. In this SaaS configuration, a secured two-way connection between the local host and the cloud processor is managed by Google Drive.

implementation, in SaaS, chipless tags are scanned by the reader, and a microcontroller sends the digitized magnitude, phase, and location of the tag to the local PC (or Raspberry Pi). Separate Wi-Fi modules are not needed in this configuration as they are included in the local host hardware. The local host makes 2-D images based on the 1-D frequency data and puts them in a shared Google Drive folder. The files in the shared Google Drive are synced with Google Cloud using Google file stream. The Python/Tensorflow in the Google Colab has access to this cloud-based shared folder, and the deep learning training can be done once it is needed, as will be explained in Section 5.4.

There are two advantages to using Google Colab. First, its file system, which is called Google Drive File Stream (Google Sync), can be fully integrated into almost any operating system, and thus any locally generated file can be easily accessible by Google Colab without effort. Second, its basic 12-GB random access memory (RAM)/50-GB hard disk drive (HDD) tier VM is free, which is enough for many applications, including this chipless RFID project. In the next section, the connection between the chipless RFID reader and the cloud processors is explained.

5.4 Deep Learner in Action

In this section, we will go through the details of the CNN network. We start with how to make the image data, how to do data augmentation, how to start with a typical CNN for image recognition, and how to improve it to reach to a certain accepted level of accuracy.

5.4.1 2-D Image Representation of 1-D Frequency Data

The most difficult part of every deep learning experiment is collecting data and making trustworthy datasets. In this work each tag was measured in the frequency range of 57–64 GHz with 20,000 samples for magnitude and phase, in a few different positions in front of the reader. To make a deep learning 2-D representation of the tag, the following steps are required:

1. Rescale the tag's magnitude values to a 0 to 1 range. This is done by normalization of minimum and maximum values

in the magnitude of all measured tags, and not just this particular tag.

2. Resample the tag magnitude samples from 20,000 to 65,536 samples. This is to make a 256 × 256 pixel square-sized image. This step makes it easier to have a squared picture (2^8 pixels in each dimension).
3. Put every 256 sequential frequency data into one column of a 2-D matrix.
4. Convert the resultant 256 × 256 matrix to a gray-scale image in a portable graphics format (PNG) file format.
5. Repeat these steps for the phase of the tag to create a 256 × 256 phase image.

An illustration of the outcome of these calculation procedures is shown in Figure 5.8. The magnitude response in a lower frequency band (less than 60 GHz) is lower in the 1-D figure on the left, which corresponds to the darker left side of the 2-D image on the right.

5.4.2 Data Augmentation

The next step in deep learning preparation is making enough inputs for these data-hungry deep learner networks. As the frequen-

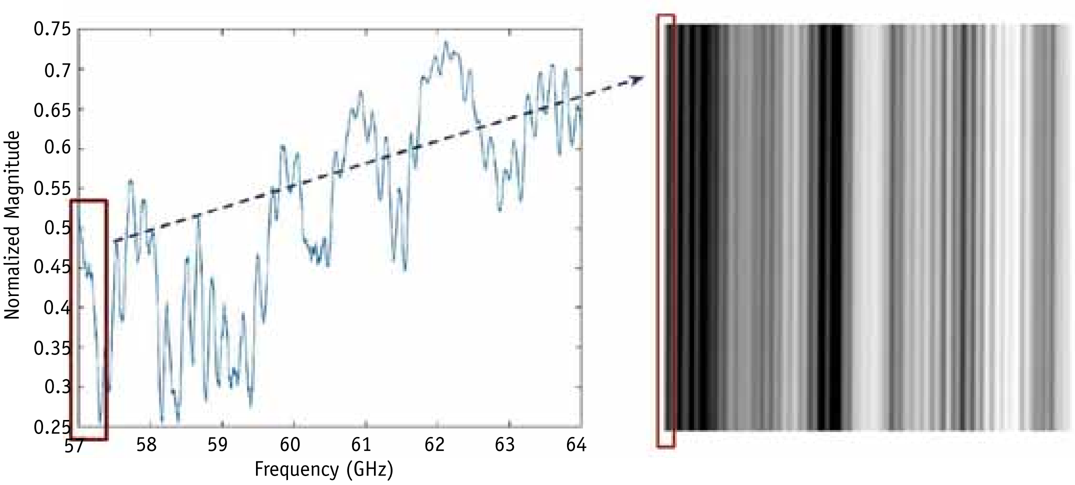


Figure 5.8 Normalized magnitude response of a tag (left) and its 2-D representation (right) [19]. (Image courtesy of IEEE.)

cy data is now presented in 2-D gray-scale pictures (as in Figure 5.8), using image-based data augmentation methods is possible. The goal of data augmentation is to increase the size of the available dataset by creating transforms of each image in the original dataset. Generally, these transforms can be simply done by changing the brightness or reflection, or altering the original image by zooming, rotating, or similar actions.

For chipless RFID 2-D images, this data augmentation comes with its own restrictions. The built-in ImageDataGenerator in Keras is avoided because it will make some transformations to the tag's 2-D data, and these transforms make the underlying data meaningless. For example, changing the tag's brightness will result in misleading 2-D augmented data, as the brightness level corresponds to the magnitude. Therefore, making random brightness levels cannot be utilized in data augmentation.

Figure 5.9 illustrates the random cropping method used in this work. The originally generated files from the magnitude and phase of the backscattered signals are 256 × 256 pixels. A random pixel from the area of the top left 106 × 106 pixels is selected, and this point is the upper left corner of the cropped 150 × 150 pixels' red square shown.

This 150 × 150 pixel wide frame is our way of standardizing the data. Although the pictures are in gray scale, their values are still in the [0, 255] range. In general we seek to make input values to the deep learners small. Here, we normalize values to be in the [0, 1].

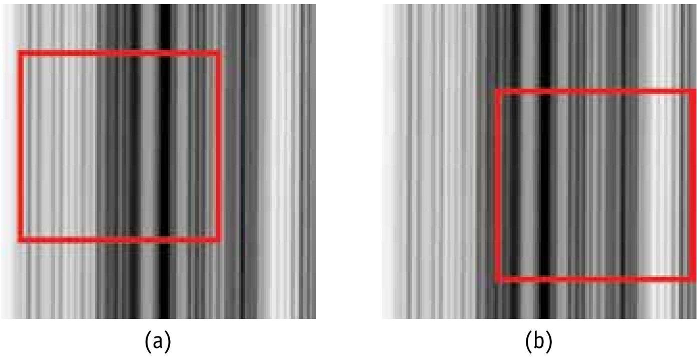


Figure 5.9 Random cropping windows from a 2-D magnitude image for data augmentation [19]. (Image courtesy of IEEE.)

5.4.3 Deep Learner Structure

After deciding on the requirements of the cloud computing, the next step is designing the deep learning structure. As training a deep learner needs lots of computing power, one easy way is to use already trained networks and customize that for the new dataset. For image-based classifiers, several powerful datasets are already available, but they are trained for real images and not for our study case of chipless RFID imaging. In this book, the data is specific to chipless RFID tags, so its training samples are merely coming from the experiments. With more users and more scanned tags, there will be more chances to create a pretrained network in the future. The proposed deep learning structure is shown in Figure 5.10. The descriptions for different layers used in Figure 5.10 are as follows:

- *Image input layer.* The InputLayer is where the image dimensions are defined. It is 150 × 150 × 1 as the pictures are 150 × 150 pixels and gray scale.
- *2-D convolutional layers.* Conv2D is a 2-D convolution with a defined filter size (3 × 3 here), with activation function of a rectified linear unit (ReLU).
- *Pooling layer.* MaxPooling2D reduces the size of the image by assigning the maximum number of each grid for every grid.
- *Flatten layer.* Flatten converts the pooled layer to a single column that is passed to the fully connected layer.
- *Dense layer.* Dense is a densely connected layer. It is used to transfer the number of layers to lower ones. The last dense stage is 27, which is this project's tagID space.

This structure is decided after a few trials and errors for achieving better accuracy in the training sets. The system is tested with different sizes of datasets and augmented data. Figure 5.11 illustrates one of the results for a trial of 50 epochs and 10 steps per epoch. The deep learner accuracy converges to 0.6549 after a few epochs and it does not improve too much afterward. This initial trained network is far from being perfect at this stage, but it will be improved as we go through the chapter. The behavior of any deep learner, including this CNN, is difficult to explain. But as a general rule, if the output accuracy does not improve (or saturates in some

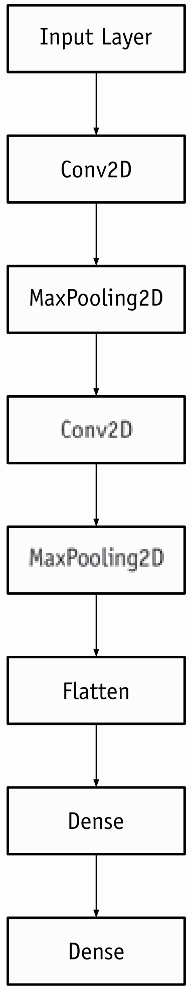


Figure 5.10 Initial structure of the deep learning used in this RFID project.

points), it means the network is unable to learn all the features or the training dataset (or augmentation) has not been chosen properly. As the learning in the CNN is based on the convolutional layers, those layers should be adjusted or added.

5.4.4 Deep Learning Results

Adding extra convolutional and maxpooling layers to the structure proposed in Section 5.10 results in higher accuracy to some degree. This is illustrated in Figure 5.12.

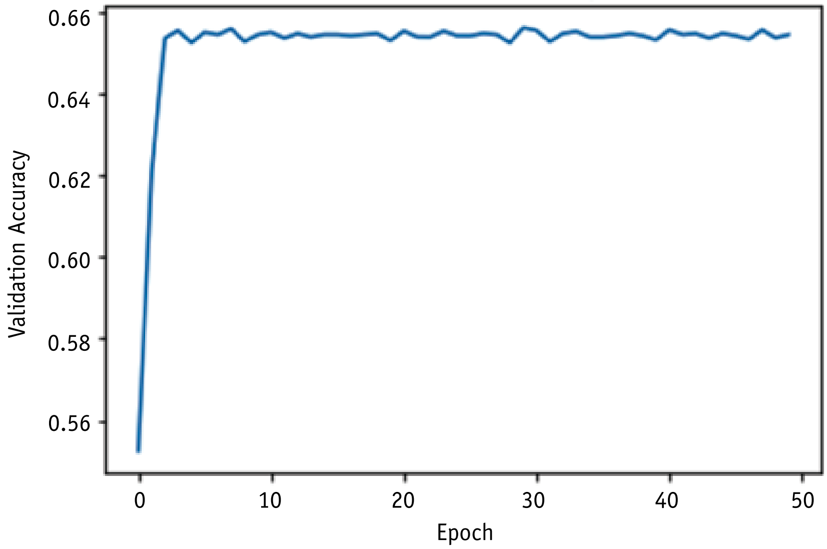


Figure 5.11 Initial validation accuracy vs epoch for the proposed deep learning system of Figure 5.10. Validation accuracy reaches a certain point and does not improve, meaning that the current network cannot learn from all of the available features or the training dataset has not been chosen properly [19]. (Image courtesy of IEEE.)

As per Figure 5.12, two layers of convolution and one layer of maxpooling is added. This is a standard way of adding complexity to a CNN network. Convolution layers with a set of kernels are the learners, and maxpooling layers reduce the dimensions. The initial accuracy does not start from zero, as the existing layers of the network are already trained for the inputs of the first attempt.

After adding these layers, the final graph of validation accuracy is shown in Figure 5.13. This shows a great decoding improvement over the initial network of Figure 5.11. The number of recognized tags is the output of the last dense layer. This means that after inputting the pictures, one binary output of the last layer (a vector of 27 cells) will indicate the actual tagID category detected.

5.5 A Reliable Reader Based on Cloud Deep Learning

The overall flow graph for utilizing deep learning in the cloud is illustrated in Figure 5.14. This is an upgrade to the previous highly reliable tag decoder architecture presented in Section 4.7. Compared to Figure 4.27, Figure 5.14 uses only one learner and all in-

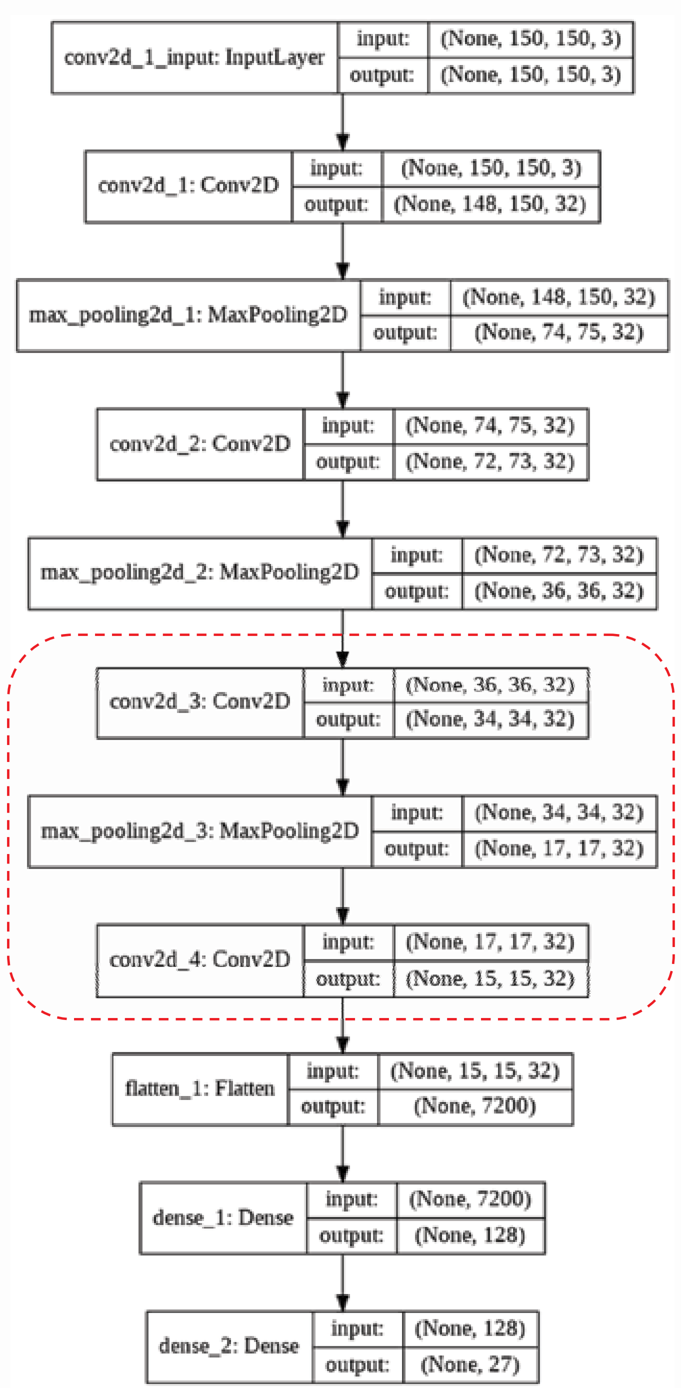

conv2d_1_input: InputLayer
input: (None, 150, 150, 3)
output: (None, 150, 150, 3)
conv2d_1: Conv2D
input: (None, 150, 150, 3)
output: (None, 148, 150, 32)
max_pooling2d_1: MaxPooling2D
input: (None, 148, 150, 32)
output: (None, 74, 75, 32)
conv2d_2: Conv2D
input: (None, 74, 75, 32)
output: (None, 72, 73, 32)
max_pooling2d_2: MaxPooling2D
input: (None, 72, 73, 32)
output: (None, 36, 36, 32)
conv2d_3: Conv2D
input: (None, 36, 36, 32)
output: (None, 34, 34, 32)
max_pooling2d_3: MaxPooling2D
input: (None, 34, 34, 32)
output: (None, 17, 17, 32)
conv2d_4: Conv2D
input: (None, 17, 17, 32)
output: (None, 15, 15, 32)
flatten_1: Flatten
input: (None, 15, 15, 32)
output: (None, 7200)
dense_1: Dense
input: (None, 7200)
output: (None, 128)
dense_2: Dense
input: (None, 128)
output: (None, 27)

Figure 5.12 Details of the final layers used in the CNN. Layers in red are added in the second stage. The last dense layer is basically the same as the number of tags [19]. (Image courtesy of IEEE.)

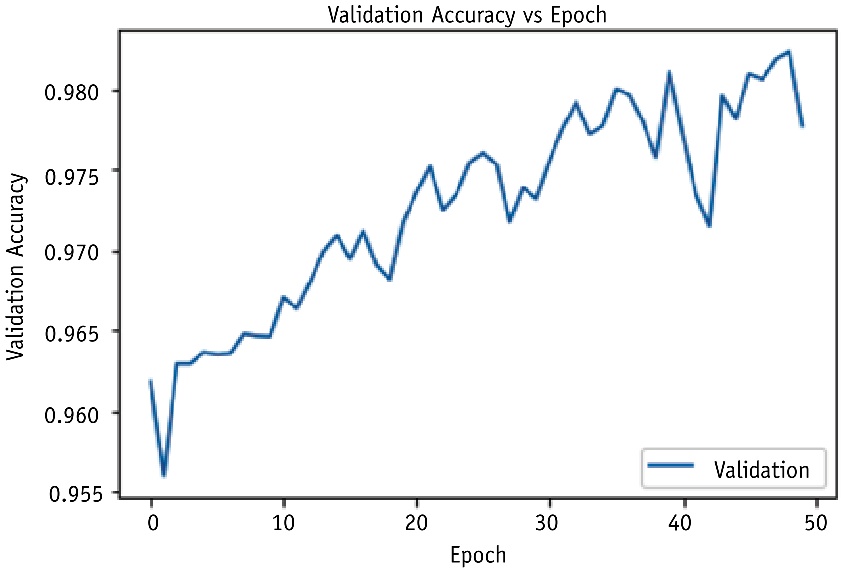


Figure 5.13 Improved validation accuracy vs epochs in the modified deep learning system. Two layers of CONV2D and one MaxPooling2D added to the initial CNN network, resulting in higher accuracy for the validation data [19]. (Image courtesy of IEEE.)

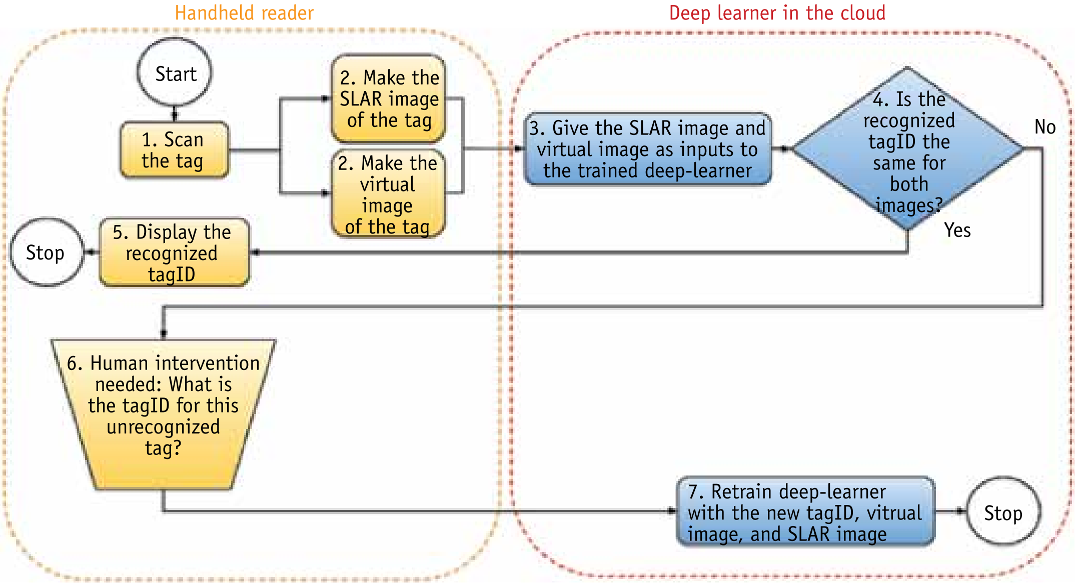

Handheld reader
Deep learner in the cloud
Start
1. Scan the tag
2. Make the SLAR image of the tag
2. Make the virtual image of the tag
3. Give the SLAR image and virtual image as inputs to the trained deep-learner
4. Is the recognized tagID the same for both images?
No
Yes
Stop
5. Display the recognized tagID
6. Human intervention needed: What is the tagID for this unrecognized tag?
7. Retrain deep-learner with the new tagID, vitrual image, and SLAR image
Stop

Figure 5.14 Flow graph of the cloud-based deep learner reader. Only one CNN network is used in the cloud for the SLAR and virtual images, in contrast to Figure 4.27 [19]. (Image courtesy of IEEE.)

puts are in 2-D forms. The main decoding steps in this system are as follows:

1. Scan the tag. Using the method described in Section 4.3.2, the tag data is scanned in the whole frequency spectrum.
2. Make an image of the tag. Here two images of the tag are made, one SLAR image-based in Section 4.6.4, and one virtual 2-D representation of the frequency response that is explained in Section 5.4.1.
3. Provide the two images data to the pretrained deep learner.
4. If the deep learner outcomes for both images of step 2 are the same, it means the systems are both trained for this tag, and also the chance of misdetection (recognizing the wrong tagID) is very low. This is because of the parallel verification mechanism. If the failure rate of each system (SLAR and virtual 2-D) is 1 − 0.975 = 0.025 (as in Figure 5.13), then the final decoding rate of the two parallel decoders of Figure 5.14 is

$$\begin{aligned}\text{decoding rate} &= 1 - \text{ false\_rate}\\ &= 1 - 0.025 \times 0.02 = 0.9993755\end{aligned} \tag{5.1}$$

5. If the outcome of the previous stage for both recognizers is the same tagID, the recognized tagID is sent back to the handheld reader and the tagID is displayed. That is the end of the tag detection process.
6. Therefore, human intervention is needed if the recognized tagID in step 4 is not the same as from the previous step. A user should confirm the tagID via direct observation and provided DIP-Switch (as shown in Figure 3.19).
7. Finally, the deep learner network should be trained with the available SLAR image and virtual image from part 2. This network will be ready to correctly recognize the tag the next time.

The CNN-based deep learning method used in the RFID reader of Figure 5.14 has a few advantages over the previous networks used in Chapter 4 (Figure 4.27) as follows:

- The deep learner method uses 2-D representation of the tag, both in the form of virtual 2-D frequency data and in SLAR imaging. This is a great advantage as augmented data is much easier to create and interpret in 2-D.
- The chipless RFID in Figure 5.14 uses one deep learning network for both backscattering frequency and SLAR cases, whereas in the system in Figure 4.27 it uses two pattern recognizers simultaneously. Maintaining two different networks and making them up to date adds to extra complexity.
- The more significant advantage is using cloud facilities, which is accessible to all chipless readers and is empowered by a freely available deep learning architecture.

The disadvantage of using Google Colab is that there are no APIs for programmatic access to Colab. This means the local reader cannot call Google Colab directly through a code as of now. Therefore, some manual data handling is still in place.

5.6 Conclusions

As an extension to the chipless tag reader hardware presented in Chapter 3 and decoding algorithms of Chapter 4, a deep learning architecture was proposed and tested in two cloud-based environments, IaaS and SaaS, for a larger dataset and mass deployment of the proposed chipless RFID system.

A 2-D representation of the magnitude and phase signals over the frequency spectrum has been proposed. This 2-D image was created in a way that the augmented data based on random cropping had an explainable and meaningful representation. A few structures for the deep learner were tested and the results compared. The trained deep learner showed to have high decoding accuracy, in the order of 0.975%. Reaching a higher accuracy was shown to be possible to some extent by adding extra layers, at the cost of extra training time and processing power.

Based on the lab experiment observations and using 27 plastic tag samples, the combination of the two image-based methods in Figure 5.14 for SLAR and virtual 2-D of tags achieved a detection rate of 0.999%. That might change if the number of tags increases.

5.7 Cloud-Based Deep Learning Questions and Answers

5.1 How can I replicate the results of machine learning in this chapter myself?

Interested readers are highly encouraged to replicate the deep learning training. The Keras code and dataset can be found in in [20] and [21], respectively.

5.2 Why do we use REST APIs?

If we plan to have remote or cloud hosting, we need a mechanism of two-way communication between our reader and the host. The tag's scanned data would need to be sent to the cloud computer using a JSON or XML file format, for example. Using Message Queuing Telemetry Transport (MQTT), which is used in IoT applications, is not that feasible since a lot of tag data needs to be transferred. The RESTful API is an industry accepted way, especially for business-to-business (B2B) communications without any human interactions. Using RESTful API for data transfer in a JSON or XML file needs web-server installation on the cloud side. It is possible to implement a web-client in the ESP8266 shown in Figure 3.6, but it is strongly recommended to use a RPi or more powerful boards with any standard OS instead for making JSON files of scanned tags and sending them through to the cloud server.

5.3 Are there any other ways to make a two-way communication between the reader and the cloud host?

Yes, of course. Using SFTP, or, alternatively, using the FTP or its secured version, FTPS, can be a work-around to avoid the RESTful API method. For any communication protocol you choose, you need to install its server version on the cloud computer, and the reader should support the client-side protocol.

5.4 Do I need to use secure protocols in data transfer?

Although it is not mandatory, securing your server communications is highly recommended once it goes through the open internet. So, in addition to having a username/password (or Internet Protocol (IP) whitelisting if a fixed-IP addressing is possible), you need to install a Secure

Sockets Layer (SSL)/Transport Layer Security (TLS) certificate in the cloud server to prevent impersonations. An alternative way of securing the data can be using a secured Virtual Private Network (VPN) between the reader and the cloud host if an RPi is used in the reader.

References

[1] Keras: Deep Learning for Humans, 2022, https://keras.io.

[2] Mattoo, I. A., "Security Issues and Challenges in Cloud Computing: A Conceptual Analysis and Review," *International Journal of Advanced Research in Computer Science*, Vol. 8, No. 2, 2017.

[3] RapidScale, "Cloud Computing Stats–Security and Recovery," 2015, //www.slideshare.net/rapidscale/cloud-computing-stats-securityand-recovery.

[4] 12 Benefits of Cloud Computing, https://www.salesforce.com/hub/technology/benefits-of-cloud/.

[5] Morgan, J., "5 Reasons Why the Cloud Is Environmentally Friendly," www.missioncloud.com/blog/5-reasons-why-the-cloud-isenvironmentally-friendly.

[6] The Google Chrome Team and C. Niemann, *20 Things I Learned about Browsers and the Web*. Google, 2010, http://www.20thingsilearned.com/enGB/cloud-computing.

[7] Amazon, Amazon EC2 (On-Demand), https://aws.amazon.com/ec2/.

[8] Google, Google AI: Machine Learning Products, https://cloud.google.com/vertex-ai/pricing.

[9] Dabas, C., and J. Gupta, "A Cloud Computing Architecture Framework for Scalable RFID," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 1, 2010.

[10] Hightower, K., B. Burns, and J. Beda, *Kubernetes: Up and Running: Dive into the Future of Infrastructure*, Sebastapol, CA: O'Reilly Media, 2017.

[11] Bernstein, D., "Containers and Cloud: From LXC to Docker to Kubernetes," *IEEE Cloud Computing*, Vol. 1, No. 3, 2014, pp. 81–84.

[12] Weins, K., *Cloud Computing Trends: 2020 State of the Cloud Report*, 2020, https://www.flexera.com/blog/industry-trends/trend-of-cloudcomputing-2020.

[13] "Beware the Man in the Cloud: How to Protect Against a New Breed of Cyberattack," Help Net Security, https://www.helpnetsecurity.com/2019/01/21/mitc-attack/.

[14] Mirkovic, J., S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service: Attack and Defense Mechanisms*, Upper Saddle River, NJ: Prentice Hall PTR, 2004.

[15] Durcevic, S., 10 Cloud Computing Risks Challenges Businesses Are Facing in These Days, 2019, https://www.datapine.com/blog/cloud-computing-risks-and-challenges/.

[16] Nectar, NECTAR CLOUD, Your Gateway to the Global Research Community, https://nectar.org.au/research-cloud/.

[17] Perkel, J. M., "Why Jupyter is Data Scientists' Computational Notebook of Choice," *Nature*, Vol. 563, No. 7732, 2018, pp. 145–147.

[18] Wikipedia, *Project Jupyter*, 2019, https://en.wikipedia.org/wiki/Project_Jupyter.

[19] Arjomandi, L. M., G. Khadka, and N. C. Karmakar, "mm-Wave Chipless RFID Decoding: Introducing Image-Based Deep Learning Techniques," *IEEE Transactions on Antennas and Propagation*, Vol. 70, No. 5, 2021, pp. 3700–3709.

[20] Arjomandi, L. M., "Deep-Learning with Keras Code Listing," December 2020, Monash University, doi:10.26180/ 13309253.v2. [Online]. Available: https://doi.org/10.26180/13309253.v2.

[21] Arjomandi, L. M., Chipless RFID 2D Train and Validation Dataset, Monash University, August 2022, https://doi.org/10.26180/20444058.v2.

6

Conclusions

6.1 Conclusions

This book set out to make a state-of-the-art chipless RFID system in the 60-GHz spectrum. We utilized AI combined with SLAR imaging techniques on the low-cost and low Q-factor plastic substrates in the 60-GHz spectrum. The system was based on three main development parts: designing the tags, designing the reader, and developing appropriate decoding algorithms.

Based on the research presented, it appears that letter-based chipless tags screen printed on cheap plastic substrate are of great interest, as they are cheap and flexible enough to be used as an electromagnetic barcode. The limitation of the tags are (1) they do not have enough backscatter reflections to be detected effectively, (2) they are not symmetric in their design to be orientation-independent, and (3) their frequency response is difficult to be adjusted for peak and null locations.

For enough backscattering, two methods were proposed—a combination of several letters as one tag and screen printing the tags with double printing ink. Combining a few letters as one tag was used to increase the encoding capability. This upfolded the detectability of the tags in the real-world applications. A few trials of different printing techniques and inks were carried out for conductivity and backscatter response. Although the double printing technique increased the ink thickness, it drastically affected the sharpness of the tag edges and resulted in surface roughness.

Although the proposed tags do not have a symmetric design, the rotated tags could still be detectable to some degree because of their other design advantages and the intelligence added to the detection algorithm.

To address the adjustment of the frequency response, a few trials were conducted to tune the size of the tags' cliffs. It was shown that with adequate size of the cliffs in each letter, particular frequency peaks and nulls could be achieved. Although individual letter frequency responses interfered with each other, if the nulls and peaks were separated enough, the overall response of the tag could be adjusted. A mathematical representation of the RCS of the tag was represented and followed by a practical way of calibration using the *S*-parameters.

A modular reader hardware was developed and implemented at 60 GHz for the first time. The design was around Analog Devices HMC6350 Tx/Rx RF modules. The cross-polarized Tx/Rx configuration of this design had the advantage of suppressing copolarized signals from the Tx to the Rx, and also helped with the letter-based tags as the tags were detectable in cross-polarized scenarios. The integration problems including slow DAC and conflicts in the Future Technology Devices International (FTDI) ports were discussed and addressed in the design, using VM and customized SPI signaling for LO and Tx/Rx RF modules.

The Tx/Rx RF boards were integrated into the chipless RFID system using a PSoC5 as the MCU. A customized SPI signaling was developed for direct RF board controlling and a LO circuit was created to cover the low-resolution scanning of the associated RF boards. With the in-house developed LO, a frequency sweep of 500 MHz to 2 GHz in 10 MHz or less steps were possible. A gain/phase comparator circuit was also developed in-house to compare LO sent and received signals, and a high-precision delta-sigma ADC configured to digitize the output of this comparator.

As the reader system is based on the SLAR concept, a linear rail and its controller circuit were implemented for precise movements of the tags in front of the reader antennas. Then 10-bit DIP switches were used as the auxiliary inputs when some interaction from the user was needed. As an interface to the computers, an Arduino Wi-Fi module was added to the PSoC5 to transfer the data to the local or remote (cloud) computer for further processing of the digitized backscattered signals.

Three methods of decoding for chipless RFID tags were presented. It was shown that shadow networks, like feedforward and pattern recognizers, could effectively be used for high-accuracy decoding tasks. Choosing an appropriate classification algorithm for a particular problem was shown to be a matter of trial and error, because each algorithm had its own advantages and disadvantages and the algorithms were based on particular assumptions and datasets.

As a matter of practice, several classifiers were tested and the best-performing ones were chosen for this particular classification problem with the given dataset. To choose the right classifier, these parameters were considered: (1) speed of training, (2) accuracy of the prediction, and (3) the AUC. For some models like KNN, interpretability of the algorithm was easier to explain compared to the others.

Although pattern recognition and feedforward networks showed very high decoding rates, their false detection rate was quite problematic and misleading. To cover this intrinsic problem of a chipless RFID, a CW-SLAR method for making a low-resolution image of the tag was presented. A practical way of implementing this imaged-based technique was developed and tested based on the available two-port reader or the VNA. The tags were scanned in a 2-D matrix and the image was formed based on CW backscatter received signals and their positions. The image resolution was 2×10 pixels. The image resolution produced was low, but every letter was associated with four different pixels/areas, and each pixel/area had 256 levels of gray scale. This made the image of each letter distinguishable. The resultant image from SLAR was used as a vector to train a parallel decoding system in conjunction with former AI decoding methods to form a more highly reliable decoder.

As an enhanced version of decoding, a deep learning architecture was proposed and tested in two cloud-based environments, IaaS and SaaS. A 2-D representation of magnitude and phase signals over frequency was proposed. This 2-D image was created in a way that the augmented data based on random cropping had an explainable and meaningful representation.

A few adjustments were done for the structures of the deep learners and the results were shown. The trained deep learner showed very high accuracy in the order of 96% to 98%. Reaching higher accuracy was shown to be possible to some extent by adding extra layers at the cost of extra training time and processing power.

6.2 Fulfilling Research Goals

This book has addressed the research gaps as follows:

- Using low-cost substrates as the screen-printed mm-wave chipless tags was addressed. A few letters were used as one tag to raise the RCS to the detectable level.
- Low data encoding was addressed in the proposed tags as the tags' encoded data was not based on their Q-factor. Pattern recognizers and image-based techniques were used with a very high decoding success rate as a distinguishable pattern that is necessary from the tag. A high throughput of 6 bits/cm^2 was achieved.
- Although letter-based tags were not symmetric, it was shown that AI methods could decode the rotated letter tags to some extent.
- The wrong decoding of tags addressed using a double verification method. This was done in two different chapters. In Chapter 4, the 1-D backscattered data from tag and vector-based SLAR image was presented into two pattern recognizers to double verify the tagID. In Chapter 5, a 2-D representation of the backscattered data with a 2-D image-based SLAR was presented to a deep learner network and their outputs compared. Both methods were shown to suppress misdetection of the tagIDs effectively.
- A modular design approach was created to implement a nontrivial 60-GHz reader. The reader implementation problems were addressed in detail and some aspects of device characterization were presented.

- To address higher computational and storage demands for the ML and image processing, cloud-based computing was implemented with many advantages, including increased reliability and reduced costs per reader.

6.3 Recommendations for Future Work

Although this book has presented new information and data, it has some shortcomings that can be used as recommendations for future researchers. These recommendations are:

- Moving the tags in a precise way in front of the reader to create images is difficult, and the linear rail stepper motors need too much power, which makes the whole system slow and noisy. The MIMO technique now is feasible with the current solutions in the market [1–3]. It is recommended to make the next generation reader based on MIMO techniques, mainly to eliminate mechanical movements for reading from multiple angles or polarizations.
- Deep learning has great potential to decode chipless tags at different angles or distances, but preparing a reliable data augmentation function is necessary to train such a capable network. More research on these reliable augmentation methods is therefore recommended.
- The proposed double verification method in this book needs human interactions once a problem occurs. To avoid this, an image recognition in visible light can be added to the reader to observe the tag and send the necessary commands to deep learners directly. This will automate the whole process of decoding for the new and misdetected tags and may eliminate human intervention to a great extent.

References

[1] Texas Instruments, IWR6843, Single-Chip 60-GHz to 64-GHz Intelligent mmWave Sensor Integrating Processing Capability, http://www.ti.com/product/IWR6843.

[2] Texas Instruments, AWR1642, Single-Chip 76-GHz to 81-GHz Automotive Radar Sensor Integrating DSP and MCU, http://www.ti.com/product/AWR1642.

[3] Staal Technologies, Omniradar RIC60A, FMCW Radar Single Chip 60 GHz Radar, https://www.staaltechnologies.com/project/ric60a/.

Appendix A
Code Listing

This appendix brings the C-code within the PSoC5. The intention of bringing it here is to get some ideas about different parts of the system, like how to create a customized length SPI signal or how to communicate with Tx and Rx boards.

```
/* ========================================
Copyright (c) Larry M. Arjomandi
Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the “Software”), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:
The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONIN-
FRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
```

```
* ========================================
*/
#if defined (__GNUC__)
/* Add an explicit reference to the floating point printf library */
asm (“.global _printf_float”);
#endif
#include <project.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
uint8 finishedDMA = 0u;
uint8 dataReady = 0u;
uint16 result;
unsigned  long  ADF4350_reg0,ADF4350_reg1,ADF4350_reg2,ADF4350_reg3,ADF4350_reg4,ADF4350_reg5;
#define freq_step 10
#define freq_low_band 500
#define freq_high_band 1950
#define power_level -1
#define ADC_Sample_Num 20
#define Range (freq_high_band - freq_low_band) / freq_step + 1
#define ADF4350_low_band 1100
int main(void)
{
    CyGlobalIntEnable;
    int i,j,k,l;
    int BB_Freq,channel=0;
    char str[20];
    float Values;
    char8 resultStr[16u];
    int ADF4350_Freq,ADF4350_power;

    UART_Start();  /* Initialize components */
    AMuxSeq_Start();
    AMuxSeq_Next();
    sprintf(str,”Start\n”);
    int32 adcResult;
    float adcVolts;
    char tmpStr[25];
    Driver_Sleep_Write(0);
    ADC1_Start();
    LCD_Start();
    LCD_Position(0u,0u);
    LCD_ClearDisplay();
    LCD_PrintString(“Welcome to RFID LAB”);
    CyDelay(1000u);
    LCD_ClearDisplay();

    Driver_Sleep_Write(0);  // Initiate the linear rail
    Driver_MS1_Write(0);
    Driver_MS2_Write(0);
    Driver_Direction_Write(1);
    Driver_Step_Write(0);

    LCD_Position(0u,0u);
    VCO_CLK_Write(0);
    VCO_CLK_Write(1);
    VCO_DATA_Write(1);
    CyDelayUs(1u);
    VCO_CE_Write(1);
    VCO_DATA_Write(0);
    VCO_CLK_Write(0);
    CyDelayUs(1u);
    VCO_CLK_Write(1);

    LCD_ClearDisplay();
    LCD_Position(0u,0u);
    LCD_PrintString(“Scanning...”);
    UART_PutString(“Scanning... \n”);
    CyDelay(500u);
    Driver_Sleep_Write(1);

     // This reg0 to reg5 values are for BB-VCO and come from ADF435X program. Make sure the “Ref Freq” under “Main Control” tab is correct
    // and is the same as XTAL used. (10 MHz here)
    ADF4350_reg0 = 0xC80000;
    ADF4350_reg1 = 0x8008011;
    ADF4350_reg2 = 0x4E42;
    ADF4350_reg3 = 0x4B3;
    ADF4350_reg4 = 0xB5002C;
    ADF4350_reg5 = 0x580005;
    ADF4350_reg4 = ADF4350_reg4 & 0xFFFFFFDF; //Turn off the VCO
    ADF4350_reg4 = ADF4350_reg4 | 0x00000020; //Enable RF Output
    WriteReg_ADF4350(ADF4350_reg0);
    WriteReg_ADF4350(ADF4350_reg1);
    WriteReg_ADF4350(ADF4350_reg2);
    WriteReg_ADF4350(ADF4350_reg3);
    WriteReg_ADF4350(ADF4350_reg4);
    WriteReg_ADF4350(ADF4350_reg5);
    VCO_Freq_set(500); //Under 500 MHz operation, ADF4350 shows many harmonics.
    RST_TX_Write(1); //reset HMC6300 (TX)
    RST_TX_Write(0); //Make it back to operational mode
    RST_RX_Write(1); //reset HMC6301 (RX)
    RST_RX_Write(0); //Make it back to operational mode
    LCD_ClearDisplay();
    LCD_Position(0,0);
   if(WriteAndVerify_TX(3, 246) == -1) //Check to see if you can set the frequency on TX correctly. Reg3 is the most annoying one.
    {
        LCD_PrintString(“NOOOO”);
    }
    else
    {
        LCD_PrintString(“YAEE”);
    }
    CyDelay(1000);
    // typical values for reg0 to reg23 of HMC6350 and HMC6301, from “SoC transmitter” and “SoC receiver” programs.
     // Frequency set at 60 GHz, with medium level power in TX and medium LNA gain in RX
    WriteReg_TX(0, 0); WriteReg_RX(0, 0);
    WriteReg_TX(1, 74); WriteReg_RX(1, 16);
    WriteReg_TX(2, 241); WriteReg_RX(2, 0);
    WriteReg_TX(3, 246); WriteReg_RX(3, 3);
    WriteReg_TX(4, 0); WriteReg_RX(4, 159);
    WriteReg_TX(5, 191); WriteReg_RX(5, 15);
    WriteReg_TX(6, 108); WriteReg_RX(6, 191);
    WriteReg_TX(7, 15); WriteReg_RX(7, 109);
    WriteReg_TX(8, 143); WriteReg_RX(8, 128);
    WriteReg_TX(9, 224); WriteReg_RX(9, 64);
    WriteReg_TX(10, 83); WriteReg_RX(10, 0);
    WriteReg_TX(11, 3); WriteReg_RX(11, 0);
    WriteReg_TX(12, 0); WriteReg_RX(12, 0);
    WriteReg_TX(13, 0); WriteReg_RX(13, 0);
    WriteReg_TX(14, 0); WriteReg_RX(14, 0);
    WriteReg_TX(15, 0); WriteReg_RX(15, 0);
    WriteReg_TX(16, 54); WriteReg_RX(16, 54);
    WriteReg_TX(17, 187); WriteReg_RX(17, 187);
    WriteReg_TX(18, 70); WriteReg_RX(18, 70);
    WriteReg_TX(19, 2); WriteReg_RX(19, 2);
    WriteReg_TX(20, 52); WriteReg_RX(20, 52);
    WriteReg_TX(21, 18); WriteReg_RX(21, 18);
    WriteReg_TX(22, 0); WriteReg_RX(22, 0);
    WriteReg_TX(23, 98); WriteReg_RX(23, 98);

    int TXRX_Freq;  //Tx and Rx RF frequencies
    int a=0;
    for (i=0;i<5;i++)
    {
        TXRX_Freq = i * 6 + 34; //34, 40, 46, 52, 58
        WriteAndVerify_TX(20,TXRX_Freq);
        WriteAndVerify_RX(20,TXRX_Freq);
        CyDelay(1);
        for (j=0;j<100;j++) // (j=0;j<40;j++)
        {
            ADF4350_Freq= 500+j*10; //ADF4350_Freq=500+j*10;
            CyDelay(10);
            if ((float)ADF4350_Freq/1000.0+56.5+i*1.5 < 64.0)
                VCO_Freq_set(ADF4350_Freq);
            else
                break;
            LCD_ClearDisplay();
            ADC1_StopConvert();
            ADC1_StartConvert();
              while(ADC1_IsEndConversion(ADC1_RETURN_STATUS) == 0);  //wait for the ADC to finish capturing data
           if(ADC1_IsEndConversion(ADC1_RETURN_STATUS) != 0) /* Check for result */
            {
               adcResult = ADC1_GetResult32(); /* Get Reading */
                  adcVolts = ADC1_CountsTo_Volts(adcResult); /* Convert to volts */
                sprintf(tmpStr,”7 %2.2f %05.2f”,(float)ADF4350_Freq/1000.0+56.5+i*1.5, (adcVolts/.03)); /* Create a formatted string */
                LCD_Position(0,0); /* Display on LCD */
               // LCD_PrintString(tmpStr);  //if the data needs to be written on the LCD
                 UART_PutString(tmpStr);    // write on the UART port
         sprintf(tmpStr,”RF %03.1f+%03.2f= %4.2fGHz”,56.5+i*1.5,(float)ADF4350_Freq/1000.0,56.5+i*1.5+(float)ADF4350_Freq/1000.0);
                LCD_PrintString(tmpStr);
                LCD_Position(1,0);
               sprintf(tmpStr,”Mag %05.2f dB”, (adcVolts/.03));
                LCD_PrintString(tmpStr);
            }

            ADC1_StopConvert();

           AMuxSeq_Next();   //Analog max to switch from analog gain to analog phase

            ADC1_StartConvert();
            sprintf(str,”channel %d “,AMuxSeq_GetChannel());
            // UART_PutString(str);
              while(ADC1_IsEndConversion(ADC1_RETURN_STATUS) == 0);  //wait for the ADC to finish capturing
           if(ADC1_IsEndConversion(ADC1_RETURN_STATUS) != 0) /* Check for result */
            {
                adcResult = ADC1_GetResult32() ; /* Get Reading */
                 adcVolts = ADC1_CountsTo_Volts(adcResult) ; /* Convert to volts */
                sprintf(tmpStr,” %05.2f %d \n”, (adcVolts/.01), VCO_Lock_Read()); /* Create a formatted string */
                UART_PutString(tmpStr);
             sprintf(tmpStr,”Phase %05.2f deg”, (adcVolts/.01)); /* Create a formatted string */
                LCD_Position(2,0); /* Display on LCD */
                LCD_PrintString(tmpStr);
                // To Matlab //
                //UART_PutChar(tmpStr);
                  // sprintf(tmpStr,”Phase %05.2f deg VCO_Lock? %d \n”, (adcVolts/.01), VCO_Lock_Read()); /* Create a formatted string */
            }
            AMuxSeq_Next();
            CyDelay(200u);
            Driver_Sleep_Write(1);
            for (j=0;j<22;j++)
            {
                Driver_Step_Write(1);
                Driver_Step_Write(0);
            }
            Driver_Sleep_Write(0);
        }
    }
        LCD_Position(3u,0u);
        LCD_PrintString(“Done.  “);
        Driver_Direction_Write(0);
        for (j=0;j<22*20;j++)
        {
            Driver_Step_Write(1);
            CyDelay(5u);
            Driver_Step_Write(0);
        }
        Driver_Sleep_Write(0);
         ADF4350_reg4 = ADF4350_reg4 & 0xFFFFFFDF; //VCO Power down
        WriteReg_ADF4350(ADF4350_reg4);
        WriteAndVerify_TX(4,255); // TX power down
        WriteAndVerify_RX(0,255); // RX power down
}

```

```

int VCO_Freq_set(int ADF4350_Freq)  /*Function to Set the VCO Frequency*/
{
    int Div1; //the div in reg0 calculations
    int value1,i;
    if ((ADF4350_Freq >= 500) && (ADF4350_Freq < 550))
    {
        ADF4350_reg4 = 0xB5002C;
        Div1 = 8;
    }
    if ((ADF4350_Freq >= 550) && (ADF4350_Freq < 1100))
    {
        ADF4350_reg4 = 0xA5002C;
        Div1 = 4;
    }
    if ((ADF4350_Freq >= 1100) && (ADF4350_Freq < 2200))
    {
        ADF4350_reg4 = 0x95002C;
        Div1 = 2;
    }
    if((ADF4350_Freq >= 2200) && (ADF4350_Freq < 2500))
    {
        ADF4350_reg4 = 0x85002C;
        Div1 = 1;
    }
    value1 = ADF4350_Freq * Div1 / 10;
    ADF4350_reg0 = ADF4350_reg0 & 0x80007FFF;
    ADF4350_reg0 = ADF4350_reg0 | (value1 << 15);
    ADF4350_reg4 = ADF4350_reg4 & 0xFFFFFFDF; //Disable RF Output
    WriteReg_ADF4350(ADF4350_reg4);
    WriteReg_ADF4350(ADF4350_reg0);
    ADF4350_reg4 = ADF4350_reg4 | 0x00000020; //Enable RF Output
    WriteReg_ADF4350(ADF4350_reg4);
    return 0;
}

int Putbit0_TX() //put a “0” into the Tx data line (SDI_Tx)
{
    SDI_TX_Write(0);
    SCK_TX_Write(0);
    SCK_TX_Write(1);
    SCK_TX_Write(0);
    return(0);
}

int Putbit1_TX() //put a “1” into the Tx data line (SDI_Tx)
{
    SDI_TX_Write(1);
    SCK_TX_Write(0);
    SCK_TX_Write(1);
    SCK_TX_Write(0);
    return(0);
}

int Putbit0_RX()  //put a “0” into the Rx data line (SDI_Rx)
{
    SDI_RX_Write(0);
    SCK_RX_Write(0);
    SCK_RX_Write(1);
    SCK_RX_Write(0);
    return(0);
}

int Putbit1_RX() //put a “1” into the Rx data line (SDI_Rx)
{
    SDI_RX_Write(1);
    SCK_RX_Write(0);
    SCK_RX_Write(1);
    SCK_RX_Write(0);
    return(0);
}

int WriteAndVerify_TX(unsigned int Reg_no, unsigned int R) // Write and then verify the written data in the Tx
{
    char tmpStr [100];
    WriteReg_TX(Reg_no, R);
    while (R != ReadReg_TX(Reg_no))    //if register value can’t be verified
    {
        WriteReg_TX(Reg_no, R);
        CyDelay(100);
            sprintf(tmpStr,”Regno_TX %d Read %d Actual %d\n”, Reg_no,ReadReg_TX(Reg_no),R);
        UART_PutString(tmpStr);
        return (-1);
    }
return 0;
}

int WriteAndVerify_RX(unsigned int Reg_no, unsigned int R) // Write and then verify the written data in the Rx
{
    char tmpStr [100];
    WriteReg_RX(Reg_no, R);
    while (R != ReadReg_RX(Reg_no))
    {
        WriteReg_RX(Reg_no, R);
        CyDelay(100);
            sprintf(tmpStr,”Regno_RX %d Read %d Actual %d\n”, Reg_no,ReadReg_TX(Reg_no),R);
        UART_PutString(tmpStr);
    }
}

int WriteReg_TX(unsigned int Reg_no, unsigned int R) // Wite Data to Reg_no in Tx
{
    SCK_TX_Write(0);
    SEN_TX_Write(0);
    int i;
    for(i=0;i<=7;i++)
        if (((R>>i) & 0x01) == 1)
            Putbit1_TX();
        else
            Putbit0_TX();
    for(i=0;i<=5;i++) //Register Address
        if (((Reg_no>>i) & 0x01) == 1)
            Putbit1_TX();
        else
            Putbit0_TX();
    Putbit1_TX(); // bit RW = 1
    Putbit0_TX();
    Putbit1_TX();
    Putbit1_TX();
    CyDelayUs(1);
    SEN_TX_Write(1);
    CyDelayUs(1);
    return(0);
}

int ReadReg_TX(int Reg_no)  //Read Tx register data
{
    int i, R=0xFF; //dummy input
    SCK_TX_Write(0);
    SEN_TX_Write(0);
    for(i=0;i<=7;i++)
        if (((R>>i) & 0x01) == 1)
            Putbit1_TX();
        else
            Putbit0_TX();
    for(i=0;i<=5;i++) //Register Address
        if (((Reg_no>>i) & 0x01) == 1)
```

```
            Putbit1_TX();
        else
            Putbit0_TX();
    Putbit0_TX(); // bit RW = 0
    Putbit0_TX();
    Putbit1_TX();
    Putbit1_TX();
    SEN_TX_Write(1);
    CyDelayUs(1);
    SCK_TX_Write(0);
    SCK_TX_Write(1);
    SCK_TX_Write(0);
    CyDelayUs(1);
    SEN_TX_Write(0);
    int a=0;
    R=0;
    for(i=0;i<=7;i++)
    {
        SCK_TX_Write(0);
        SCK_TX_Write(1);
        CyDelayUs(1u);
        a = Scan_TX_Read();
        CyDelayUs(1u);
        R=R + (a<<i);
        SCK_TX_Write(0);
    }
    SCK_TX_Write(0);
    SEN_TX_Write(1);
    return(R);
}

int WriteReg_RX(unsigned int Reg_no, unsigned int R)  // Wite Data to Reg_no in Rx
{
    SCK_RX_Write(0);
    SEN_RX_Write(0);
    int i;
    for(i=0;i<=7;i++)
        if (((R>>i) & 0x01) == 1)
            Putbit1_RX();
        else
            Putbit0_RX();
    for(i=0;i<=5;i++) //Register Address
        if (((Reg_no>>i) & 0x01) == 1)
            Putbit1_RX();
        else
            Putbit0_RX();
    Putbit1_RX(); // bit RW = 1
    Putbit1_RX();
    Putbit1_RX();
    Putbit1_RX();
    CyDelayUs(1);
    SEN_RX_Write(1);
    return(0);
}

int ReadReg_RX(int Reg_no)  //read Rx register data
{
    int i, R=0xFF; //dummy input
    SCK_RX_Write(0);
    SEN_RX_Write(0);
    for(i=0;i<=7;i++)
        if (((R>>i) & 0x01) == 1)
            Putbit1_RX();
        else
        Putbit0_RX();
    for(i=0;i<=5;i++) //Register Address
        if (((Reg_no>>i) & 0x01) == 1)
            Putbit1_RX();
        else
            Putbit0_RX();
    Putbit0_RX(); // bit RW = 0
    Putbit1_RX();
    Putbit1_RX();
    Putbit1_RX();
    SEN_RX_Write(1);
    CyDelayUs(1);
    SCK_RX_Write(0);
    SCK_RX_Write(1);
    SCK_RX_Write(0);
    CyDelayUs(1);
    SEN_RX_Write(0);
    int a=0;
    R=0;
    for(i=0;i<=7;i++)
    {
        SCK_RX_Write(0);
        SCK_RX_Write(1);
        CyDelayUs(1u);
        a = Scan_RX_Read();
        CyDelayUs(1u);
        R=R + (a<<i);
        SCK_RX_Write(0);
    }
    SCK_RX_Write(0);
    SEN_RX_Write(1);
    return(R);
}

int Putbit0_ADF4350() //put bit 0 on ADF4350 Data line
{
    VCO_DATA_Write(0);
    VCO_CLK_Write(0);
    VCO_CLK_Write(1);
    VCO_CLK_Write(0);
    return(0);
}

int Putbit1_ADF4350()  //put bit 1 on ADF4350 Data line
{
    VCO_DATA_Write(1);
    VCO_CLK_Write(0);
    VCO_CLK_Write(1);
    VCO_CLK_Write(0);
    return(0);
}

int WriteReg_ADF4350(unsigned long R)
{
    VCO_CE_Write(0);
    int i,j;
    for(i=31;i>=0;i--)
    {
        if ((R>>i & 0x01) == 1)
            Putbit1_ADF4350();
        else
            Putbit0_ADF4350();
    }
    VCO_CE_Write(1);
    return(0);
}
```

Appendix B
PCB Layout

The PCB of the latest version of the digital controller board developed for this book is presented here (Figure B.1).

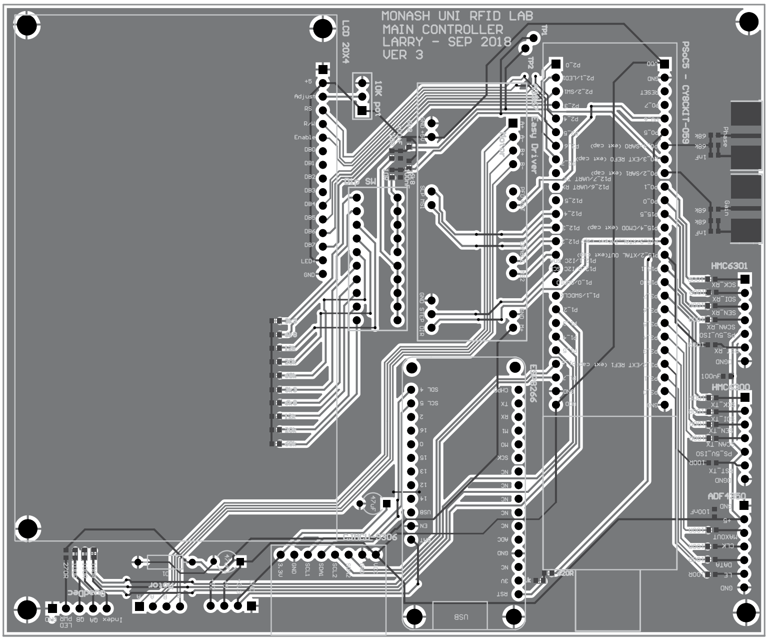


Figure B.1 The two-layer digital mainboard of Figure 3.19.

List of Acronyms

1-D	One-dimensional
2-D	Two-dimensional
3-D	Three-dimensional
5G	The fifth generation of cellular networks
ADC	Analog-to-digital converter
AI	Artificial intelligence
AIS	Artificial immune system
AMux	Analog multiplexer
ANN	Artificial neural network
API	Application Programming Interface
ASIC	Application-specific integrated circuit
AUC	Area under curve
AVI	Automated vehicle identification
AWS	Amazon Web Services
B2B	Business-to-business
BB	Baseband

BB-QN	Baseband quadrature negative
BB-QP	Baseband quadrature positive
BB-IM	Baseband in-phase negative
BB-IP	Baseband in-phase positive
BBVGA	Baseband voltage gain amplifier
BPF	Bandpass filter
BW	Bandwidth
CASB	Cloud access security broker
CMOS	Complementary metal-oxide-semiconductor
CNN	Convolutional neural network
CPU	Central processing unit
CPW	Coplanar waveguide
CST	Computer simulation technology
CW	Continuous wave
CW-SLAR	Continuous-wave side-looking airborne radar
DAC	Digital-to-analog converter
DBN	Deep belief network
dBsm	dB square meters
DF	Dissipation factor
DIP	Dual in-line package
DTS	Data transformation services
EC2	Elastic Compute Cloud
EIRP	Equivalent isotropically radiated power
EMI	Electromagnetic interference
EPC	Electronic Product Code
FFT	Fast Fourier transform
FM	Frequency modulation
FMCW	Frequency modulated continuous-wave

FTDI	Future Technology Devices International
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
GEO-SAR	Geosynchronous satellite synthetic aperture radar
GPIB	General Purpose Interface Bus
GPU	Graphics processing unit
GUI	Graphical user interface
HDD	Hard disk drive
HTTPS	Hypertext Transfer Protocol Secure
I2C	Inter-integrated circuit
IaaS	Infrastructure-as-a-service
IDT	Interdigital transducer
IF	Intermediate frequency
IoT	Internet of Things
IP	Internet Protocol
IQ	In-phase and quadrature
iSAR	Inverse synthetic aperture radar
ISM	Industrial, scientific, and medica
IT	Information technology
JSON	JavaScript Object Notation
KNN	K-nearest neighbors
LAN	Local area network
LCD	Liquid crystal display
LFMCW	Linear frequency modulation continuous wave
LNA	Low noise amplifier
LO	Local oscillator
LoS	Line of sight
LPF	Lowpass filter

LSB	Least significant bit
MCU	Microcontroller unit
MFA	Multifactor authentication
MIMO	Multiple-input multiple-output
ML	Machine learning
MLP	Multilayer perceptrons
MQTT	Message queuing telemetry transport
MRDA	Multiple-resonant dipole antenna
MS-SQL	Microsoft Structured Query Language
MSB	Most significant bit
MSE	Mean square error
NATA	National Association of Testing Authorities
NECTAR	National eResearch Collaboration Tools and Resources
NFC	Near-field communications
NFL	No free lunch
NN	Neural network
OCT	Order completion time
OOK	On-off keying
Opt-aiNET	Optimization artificial immune network
OS	Operating system
OS-ELM	Online sequential extreme learning machines
PA	Power amplifier
PaaS	Platform-as-a-service
PCA	Principal component analysis
PCB	Printed circuit board
PLF	Polarization loss factor
PLL	Phase-locked loop

PNG	Portable Graphics Format
PPM	Pulse position modulation
PROMPT	Pediatric Risk of Mortality Prediction Tool
PSoC	Programmable system-on-chip
PSoC5	Programmable system-on-chip 5
PSoC-5LP	Programmable system-on-chip 5 low power
QAM	Quadrature amplitude modulation
QR	Quick response
RAM	Random access memory
RCS	Radar cross section
ReLU	Rectified linear unit
REST	Representational State Transfer
RF	Radio frequency
RFID	Radio frequency identification
RPi	Raspberry Pi
Rx	Receiver
SaaS	Software-as-a-service
SAR	Synthetic aperture radar
SAR ADC	Successive approximation register analog-to-digital converter
SAW	Surface acoustic wave
SDA	Stacked denoising autoencoder
SFTP	SSH File Transfer Protocol
SLAR	Side-looking airborne radar
SLMPA	Stub loaded microstrip patch antenna
SNR	Signal-to-noise ratio
SPI	Serial peripheral interface
SpS	Samples per second

SQL	Structured Query Language
SRR	Split ring resonator
SSH	Secure Shell Protocol
SSL	Secure Sockets Layer
SVM	Support vector machines
tagID	Tag identification
TDR	Time-domain reflectometry
TLS	Transport Layer Security
TPU	Tensor processing unit
Tx	Transmitter
UART	Universal asynchronous receiver transmitter
USB	Universal Serial Bus
UWB	Ultrawideband
VCO	Voltage controlled oscillator
VGA	Voltage gain amplifier
VM	Virtual machine
VNA	Vector network analyzer
VPN	Virtual private network
WaveDAC	Wave digital-to-analog converter
WaveDAC8	Wave digital-to-analog converter 8 bits
Wi-Fi	Wireless fidelity
WiGig	Wireless gigabit
XML	Extensible Markup Language

About the Authors

Larry M. Arjomandi, a current senior member of the IEEE, received his BSc and MSc in electronics/telecommunications from Ferdowsi University of Mashhad, Iran, and a PhD in electrical and computer science from Monash University, Australia, in 2019. Previously, he worked as a research engineer with the Amir-Kabir University Microwave Lab and Niroo Research Center, as an information analyzer with Tehran Stock Exchange, and as a radio optimization manager with Ericsson. Currently, he is a sessional senior research engineer in embedded systems at Swinburne University, and a senior system engineer at Transport Certification Australia. Dr. Arjomandi received the PARC U.S. Research Associate award, Palo Alto, California, in 2018, and has authored two translated books, some national and international papers, and two international patents. His current research interests include chipless RFID, image-processing techniques, and real-life applications of machine learning.

Nemai Chandra Karmakar, a current senior member of the IEEE, obtained his BSc (EEE) and MSc (EEE) from Bangladesh University of Engineering and Technology, Dhaka, his MSc in electrical engineering from the University of Saskatchewan, Saskatoon, Canada, his PhD from the University of Queensland, Brisbane, Australia in 1999, postgraduate diploma in teaching in higher education from the National Institute of Education, Nanyang Technological University, Singapore, and master's in higher education from Griffith University, Australia. Previously, he worked as an assistant engineer at Electronics Institute, Atomic Energy Research Establishment, Dhaka, Bangladesh, a research assistant at the Communications Research Group, University of Saskatchewan, Canada, and as a microwave design engineer at Mitec Ltd., Brisbane, Australia, where he contributed to the development of land mobile satellite antennas for the Australian Mobilesat. Dr. Karmakar currently works in the Faculty of Engineering at Monash University as an associate professor in the Department of Electrical and Computer Systems Engineering, and he is the director of Monash Microwave, Antennas, RFID and Sensors (MMARS) Lab.

Index