LOW POWER RFID SYSTEM DESIGN AND IMPLEMENTATION

by

Jacob Bowen

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2018

Approved by:

____

Dr. Jeremy Holleman

____

Dr. Thomas Weldon

____

Dr. Andrew Willis

ABSTRACT

JACOB BOWEN. Low Power RFID System Design and Implementation. (Under the direction of DR. JEREMY HOLLEMAN)

A successful design and implementation of an RFID(Radio Frequency Identification) tag and reader.

The RFID reader was implemented using a USRP board and GNU radio. The RFID system designed here is considered a UHF(ultra high frequency) RFID. The reader transmits and receives a 915MHz signal and demodulates the receive signal to read the data from the RFID tag. The data from the RFID tag is transmitted through back-scatter and modulated using BPSK. The RFID tag has a switch and a matching network to allow for the wave to be reflected at 0 degrees or 180 degrees by switching from a short and an open.

The RFID tag's matching network was designed using ADS and is controlled with an on board MSP430 micro-controller. The matching network is L-match topology that can be toggled to just a series inductor. It has the ability to change between the two states using the MSP430 board and a second switch. The second switch is controlled at the same time as the main switch resulting in an open or short seen by the antenna.

Back-scatter is used in RFID because it uses very little power on the tag. The highest power consumption is by the MSP430 board. Even in the lowest power setting, it uses 60 micro-Watts. There are multiple ways to reduce the power consumption to reach the nano-Watt range but they were not implemented for this thesis. They are all discussed in the future work chapter.

DEDICATION

I would like to dedicate this to my parents. They have always instilled a great importance on the value of education on me. They have encouraged me throughout my entire academic career. Whenever I was overly stressed about school, I would call them and they were always willing to talk me down and put everything in perspective.

ACKNOWLEDGEMENTS

I would like to acknowledge everyone on my thesis committee, Dr. Holleman, Dr. Willis, and Dr. Weldon. I chose all of you to be on my committee because you all have had the biggest impact on my academic career.

Dr. Willis, You were the first teacher to drive real excitement for electrical engineering within me. I was starting to lose focus and determination until your DSP course and had the pleasure of experiencing your teaching style.

Dr. Weldon, you have been my advisor since I first entered engineering. You first planted the idea of getting a masters in my head. You have also have helped me so much with understanding all the electro-magnetic's occuring with this RFID system. You stayed hours late a few days just to help me diagnose problem with the system and for that I am very grateful.

Finally, Dr. Holleman, I am so glad to have gotten my masters under you. You have taught me so much and given me an education much beyond what is typical in UNCC or even many of the top Electrical Engineering schools in the country. I experienced such a variety of projects and learned a large variety of topics from software to hardware. I am extremely grateful for you giving me the opportunity to be one of your students.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPSK An acronym for Binary Phase Shift Keying

ECE An acronym for Electrical and Computer Engineering.

HF An acronym for High Frequency

IC An acronym for Integrated Circuit

IF An acronym for Intermediate Frequency

LF An acronym for Low Frequency

PCB An acronym for Printed Circuit Board

PSK An acronym for Phase Shift Keying

RF An acronym for Radio Frequency

RFID An acronym for Radio Frequency Identification.

SDR An acronym for Software Defined Radio

UHD An acronym for USRP Hardware Driver

UHF An acronym for Ultra High Frequency

USRP An acronym for Universal Software Radio Peripheral

PREFACE

The goal of this thesis was to serve as a stepping stone for expanded uses of RFID communication. It was meant to prove the viability of RFID communication for low power communication for research purposes. The main benefits of RFID communication is the low power consumption as well as the ability to create very small RFID tags.

CHAPTER 1: INTRODUCTION

RFID is an acronym for "Radio-Frequency IDentification" and is a system where digital data is encoded in an RFID tag and communicated to a RFID reader through radio waves[1]. RFID systems work off a concept called back-scatter. This allows for very low power communications over a substantial distance. Most commercial RFID tags are actually completely passive. They convert the incoming wave into a power source so they can transmit the data they contain[2]. The RFID tag discussed in this thesis is considered a semi-passive tag. It has an external battery as it's power source but still uses back-scatter to transmit its data.

A RFID Reader was also programmed within the scope of this thesis. It was programmed using a USRP board and GNU radio. It works by generating a 20kHz sine wave, which it then uses a 915MHz wave as a carrier to transmit the combined frequencies. The receiver then reads the reflections from the RFID tag and converts it back down to base-band. From there, the signal is conditioned and then a BPSK demodulation is performed. The resulting data is stored in a binary file.

1.1 Background

The first ancestor of RFID was invented by Leon Theremin. It was a listening device for the soviet union that retransmitted incident radio waves with the added audio information. Although it is not an identification tag, it is considered the first iteration of RFID since it was a passive device that was activated by waves from an outside source[2].

The first true ancestor of modern RFID was a device invented by Mario Cardullo on January 23, 1973. It was a passive radio transponder with memory, powered by the interrogating signal[2]. Where as modulated back-scatter was demonstrated in 1973 by Steven Depp, Alfred Koelle, and Robert Frayman[2].

Modulated back-scatter is a means of low power communication and is the main concept behind RFID systems. Back-scatter is where the incident wave is reflected back to the transmitter by an antenna. This can be modulated by switching the antenna to terminating with an open or a short. This will result in the reflected wave being in phase or out of phase and BPSK modulation is achieved. The antenna can be terminated with different impedance's to achieve various amplitudes and phases for higher modulation schemes.

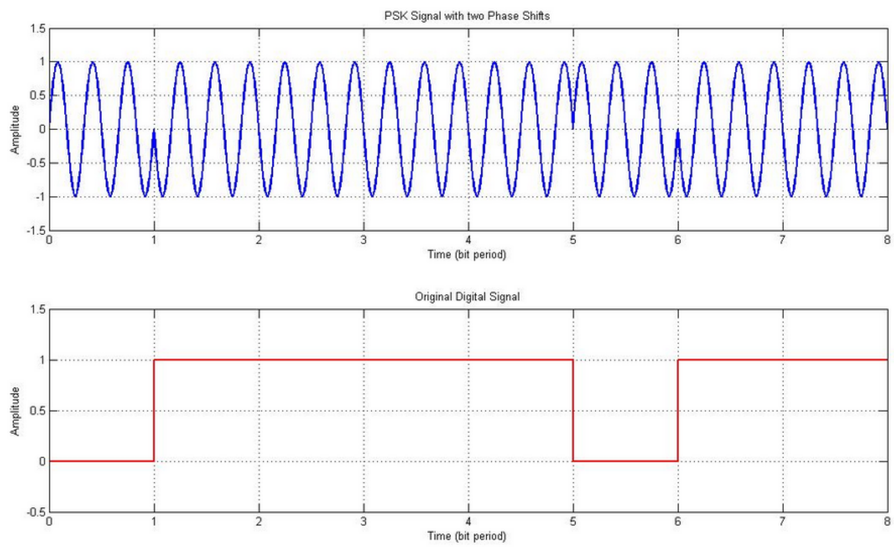


Figure 1.1: Example showing how BPSK modulates a wave.

This figure demonstrates how the wave is modulated with BPSK. The RF wave changes phase by 180 degrees when the digital signal changes from a 1 to a 0 or vice versa. Demodulation occurs by mapping the different phases to a certain bit and the digital signal is recovered. This is very similar to how a back-scattered wave is modulated.

Also in 1973, Charles Walton received a patent for a passive transponder to unlock doors without a key. He created a card with an embedded transponder that communicated with a reader near the door. When the correct ID was read by the reader, it would unlock the door[3].

The U.S. Government also contributed to RFID systems. In 1970, Los Alamos National Laboratory was tasked by the Energy Department to create a system for tracking nuclear materials. They developed a system of putting a transponder in a truck and readers at the gates of facilities. The readers would wake up the transponders in the trucks and they would respond with their ID. This system became commercialized as an automated toll payment system in the mid 1980's by a group of scientists that left Los Alamos[3].

Los Alamos also developed a passive RFID tag to track cows for the Agricultural Department. This was desired because the cows are given hormones and medicines when they are ill. It was difficult to keep track of the cows, however, so some cows were given two doses accidentally. The system developed helped identify the cows with a completely passive RFID tag on a UHF system[3].

Eventually, companies developed low-frequency(125kHz) systems with smaller transponders. They enclosed the transponder in glass and injected it under the cows skin. This is still the method used in cows today. Low-frequency transponders were also developed into cards for granting access to buildings[3].

Over time, these low-frequency systems were commercialized and, as demand increased, high-frequency systems(13.56 MHz) were developed. High frequency transponders had a greater range and faster data rates. These systems were originally used for tracking reusable containers and other assets. Today, however, high frequency systems are used in a variety of things such as access control, payment systems, contactless smart cards, and anti-theft devices in cars[3].

In the 1990's, IBM first patented an ulta-high frequency RFID system. This system greatly increased the read range, up to 20 feet, and a much faster data transfer. This technology was never commercialized by IBM however. It was sold to Intermec, a bar code systems provider, who developed many uses for this system. But the demand was low so the technology was very expensive at the time[3].

In 1999, David Brock and Sanjay Sarma, from the Massachusetts Institute of Technology, greatly improved the versatility of UHF RFID. They researched the possibility of putting low-cost RFID tags on all products to track to track them through the supply chain. They made the tag very simple to reduce price by only storing a serial number on it. The data associated with the serial number was stored in a database that the reader could access over the internet. Sarma and Brock turned RFID into a networking technology which greatly improved the usefulness of RFID in the supply chain[3].

Between 1999 and 2003, the Auto-ID Center gained funding from over 100 companies and the U.S. Department of Defense to open research labs for RFID. They developed two air interface protocols (Class 1 and Class 0), the Electronic Product Code(EPC) numbering scheme, and a network architecture for RFID tag information on the internet. This regulation helped pave the way to improving the reliability and usefulness for RFID, making it ready for broad adaptation[3].

Today, RFID is used in a variety of ways. It is used in commerce for supply chain management. Retail stores use it to tag items and prevent shop lifting. RFID tags have become very widely used in access control, replacing magnetic stripe cards. They are used in Passports and carry the same information as inside the passport. It is used in animal identification all over the world. Hospitals use RFID to manage mobile medical equipment, improve patient workflow, monitor environmental conditions, and protect patients and staff from infections. Library's also use RFID to replace barcodes on library items. Sports have started using it to improve timing of races[2].

1.2 Advantages

RFID has many advantages as opposed to other data collection methods. One includes the range on the RFID system. Depending on the frequency used, the range can be within a few meters up to a few hundred meters. Also line of sight is not required to read the data from a tag. This is in contrast with bar codes that require optical vision within a few inches to read the data.

Another advantage is the ability to scan multiple tags near simultaneously. RFID systems have a protocol to "talk" with the tags and assign an order for all the tags in range to transmit their data. This is useful for inventory management and tracking. Again, in contrast with bar code which needs to be check one at a time[4].

The final main advantage of RFID is the cost. Tags cost less than a cent but a good reader can be expensive. Bar code scanners are cheaper initially but the cost is made up for in efficiency and reduced labor costs[4].

1.3 Shortfalls

Although RFID has many advantages, it still suffers in performance sometimes. It can communicate through most materials but it performs poorly through metal and water. This can cause errors while demodulating the signal. Also, if multiple readers are operating in the same area, interference can be an issue[4].

RFID also lacks a global standardization. The frequencies used for UHF RFID in the USA are incompatible with the frequencies used by Europe or Japan. This prevents tracking through international trading with a standard tag. A special tag that is operational with all frequencies must be used[2].

Another problem with RFID is security and privacy. The tags are susceptible to an unauthorized reader stealing the information and potentially changing the data. The tags are world-readable so there is no current security or encryption to prevent the data being stolen. RFID used for access control, payment and passports operate at shorter ranges to reduce this risk but are still vulnerable to eavesdropping[2].

1.4 RFID Tag

RFID tags can be passive, semi-active/semi-passive, or active. Active tags have on-board batteries and periodically transmits its ID signal at predefined intervals. A semi-active/semi-passive or battery-assisted passive tag has a small battery on board and is toggled on by an RFID reader. They communicate through back-scatter but uses battery power for other tasks such as local control and memory access. A passive tag is powered by the readers RF energy. The RF energy awakens the tag then the tag uses back-scatter to modulate the RF signal and send the information back to the reader[2].

RFID tags are used at a variety of frequencies for different applications. Low Frequency(LF) tags are operated at 120kHz-150kHz and have a range of only 10 centimeters, these are usually used for animal identification and factory data collection. High-frequency(HF) tags operate in the ISM band of 13.56MHz. They have a read range of about 10 cm to 1 m and are used for smart cards. The UHF, ultra-high frequency, tags are divided up into two frequency ranges. The first is 433MHz, with a read range of 1-100m. It uses active tags and is used for defense applications. The other frequency range is 902-928 MHz, which the RFID system in this thesis works in, has a range of 1-12m.

An RFID tag is simply made up of three components: An IC that contains the tags information and is responsible for modulating the RF signal, a battery or RF energy capturing components, and an antenna. Below is an example of an RFID tag.

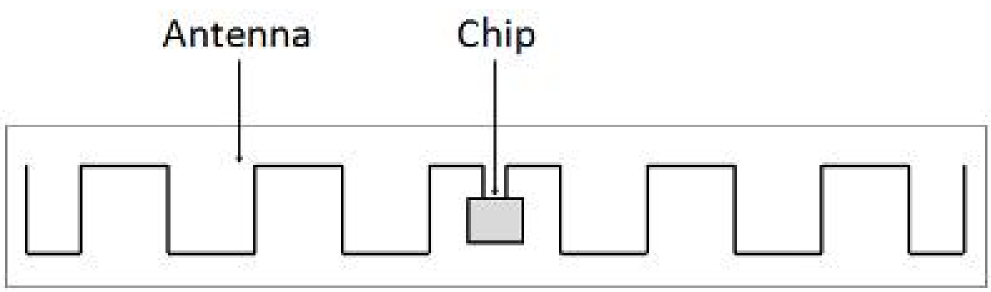


Figure 1.2: Example showing the RFID tag[5].

As shown, the antenna reserves most of the space on the tag. The chip, in this example, includes the back-scatter transistor, Voltage rectifier, and the data. This is visualized in the following image.

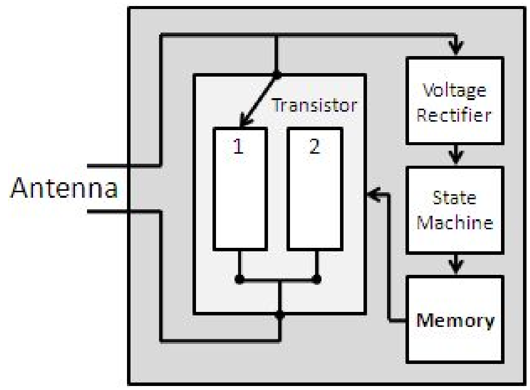


Figure 1.3: Example showing the RFID tags block diagram[5].

The transistors impedance is generally designed to have a reactance(capacitance), which results in a good reflection by the antenna to its source. The tag generally has both a resistive and a capacitive characteristic though so the signal is actually amplitude and phase modulated through backscatter. This is demonstrated and expanded upon in chapter 4[5].

1.5 RFID Reader

RFID readers are the brains behind RFID systems. There are only two types of readers, passive and active. Passive readers only receive radio signals from active tags, this is called a Passive Reader Active Tag (PRAT) system. An active reader can have either a passive tag or an active tag with it. Active Reader Passive Tag (ARPT) systems use a powerful interrogating signal to turn on the RFID tag and then sends the tag another RF signal to be modulated and back-scattered to the reader again. An Active Reader Active Tag (ARAT) is the semi-active tag systems. Where the tags are awoken by the readers then a battery is used to power the tags modulation. This allows the reader to use less power than with ARPT but the tag has to be larger to fit the battery. An ARAT system is used in this thesis, to allow for lower power from the reader as well as the tag[2].

The role of the readers are to transmit a wave to get modulated then receive the back-scattered signal and process the data. The figure below demonstrates the tag and reader.

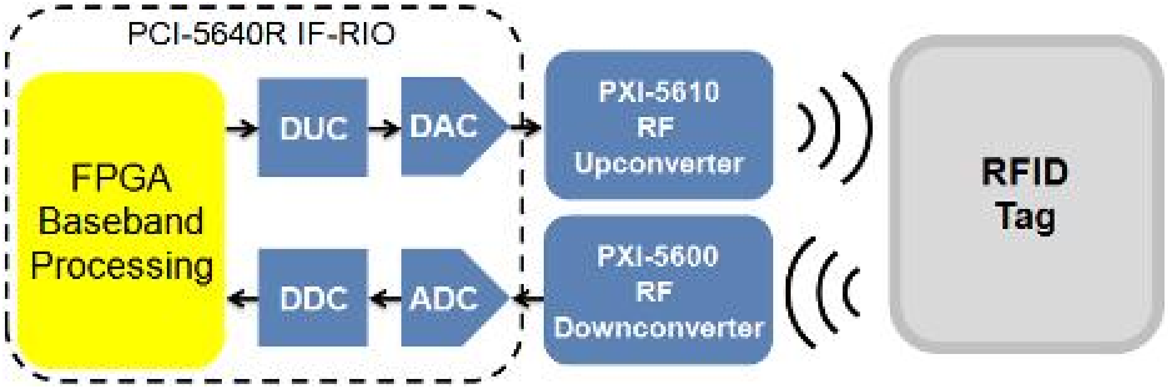


Figure 1.4: Example showing the RFID system[5].

The modulated wave is shown in figure 1.1 and it is demodulated through a constellation plot mapping. The figure below shows this.

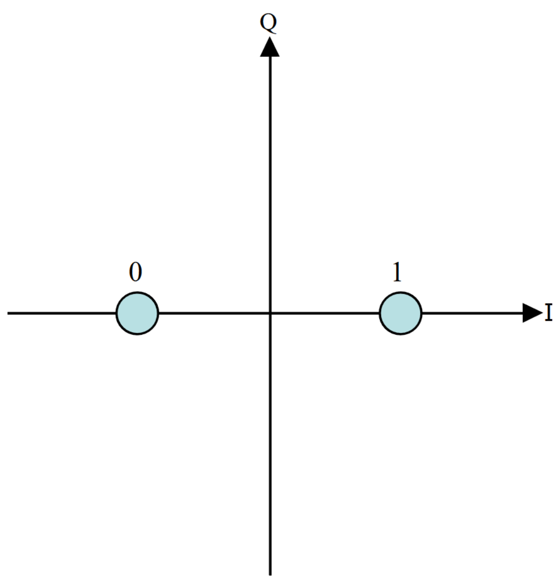


Figure 1.5: An example of the Q-I plot resulting from the modulation[6].

This is very similar to the results shown through back-scatter with slight differences shown in the later sections.

1.6 Contributions

The contributions to the complete design of this RFID system included creation of the USRP's GNU Radio code. Which was the operation of the Reader. The design and fabrication of a RFID tag on a PCB board was also contributed. The board was designed on Allegro PCB Editor and fabricated by Sierra Circuits. The final final contribution was the design and assembly of the RFID tags matching network. This was created to maximize the read range and improve the bit error rate.

The components that were not contributed and were just purchased included: The RF switch, the USRP board, and the antenna's.

1.7 Related Work

Using a USRP board as a UHF RFID reader with software defined radio(SDR) was first implemented in "A software radio-based UHF RFID reader for PHY/MAC experimentation". The paper was focused on creating a RFID reader compliant with Gen2 Protocols using the USRP. The system presented used a USRP1 which introduced latency issues due to the USB interface. A majority of the work performed was aimed at overcoming this issue caused by the hardware. EPC Gen2 protocol has strict timing for communication so it was important to overcome this issue. The authors accomplished this by keeping block sizes to the minimum allowed by the linux kernal and carefully processing the received samples. The reader was evaluated and compared to a commercial reader and proved to obtain the same read ranges with comparable transmit power. The code was released as an open source software but is not compatible with the latest version of GNU Radio[7].

The same authors also wrote a paper titled "A 'Gen2' RFID Monitor Based on the USRP". Here they created a sniffer that could decode messages between a reader and tag. They used USRP2 SDR for real-time processing and created it to study commercial RFID readers. It has accuracy up to 3 meters, which is also comparable to a commercial reader.

An extension to the SDR reader from above is presented in "Implementation and extension of a GNU-Radio RFID reader". The authors used a HF multiplexer for the USRP1 SDR to connect four transmit and receive antennas and studied the read rate an range of the modified reader[8].

In "A Cost-Effective SDR Platform for Performance Characterization of RFID Tags", Buettner's original code was modified to create a testing platform for evaluating the performance of commercial RFID tags. This updated platform enables the measuremtn of tag sensitivity and differential RCS[9].

The work performed in "Listening to Tags: Uplink RFID Measurements With an Open-Source Software-Defined Radio Tool" developed a listening device that could be used with Buettner's reader and test different clock recovery algorithms. Specifically, the Mueller and Muller algorithm, zero-crossing and polyphase filter bank clock recovery were all compared. They used Miller-8 encoding to test the read range of commercial RFID tags as well as how the environment affects the performance[10].

The paper "Complete software defined RFID system using GNU radio" focused on implementing a complete SDR UHF RFID system. Two USRP1 were used where one was a tag and the other was a reader. This system was used to test a cryptography protocol[11].

The original reader code was ported to the latest version of GNU Radio in "An UHF RFID performance evaluation architecture based on traces from a software defined transceiver". The two main contributions of this work consisted of the porting of the Gen2 RFID reader to the new version of GNU Radio and extending the code to run with offline data as an input, without the need for a USRP. The authors also compared two clock recovery algorithms while conducting the experiments using Miller-8 encoding[12].

The final line of work compared in this thesis was "WISP: A Passively Powered

UHF RFID Tag with Sensing and Computation". This paper was focused on the design and implementation of a RFID tag. The authors created a passive tag that has an on board microcontroller. The tag is powered from the transmitted RF energy and uses UHF back-scatter. They acheived ranges of 2.2m to 3.3m[13].

CHAPTER 2: TOOLS

This chapter discusses the various software and hardware tools used through out the design and implementation of the RFID system. Some tools are used in the actual implementation where as others where just used to test and fabricate parts of the system. The software tools included GNURadio, Advanced Design System(ADS), OrCAD Pspice and PCB editor, Code Composer Studio, Binary Viewer, and MATLAB. The hardware tools include oscilloscopes, source meters, and a Network Analyzer.

2.1 GNU Radio

GNU Radio is a free software development toolkit that provides signal processing blocks to implement software-defined radios and signal-processing systems[14]. GNU Radio is a python based higher level programming software. It has an extension called UHD. It allows a USRP board to be programmed by GNU Radio and results to be viewed in real time. GNU radio combined with USRP makes up the complete RFID Reader. It controls the the transmission frequency and complete demodulation scheme for the reader as well as outputs the visuals during demodulation and stores this in a binary file.

2.2 Advanced Design System

Advanced Design System (ADS) is an electronic design automation software system produced by Keysight EEsof EDA, a division of Keysight Technologies. It provides an integrated design environment to designers of RF electronic products such as mobile phones, pagers, wireless networks, satellite communications, radar systems, and high-speed data links[15]. ADS was used primarily for designing the matching network. It is a very convenient and powerful tool for RF design. The matching network was designed using a tool in ads labeled smith chart. It uses an actual smith chart to calculate and output the circuit components required to match the desired impedance.

2.3 OrCAD

OrCAD Systems Corporation was a software company that made OrCAD, a proprietary software tool suite used primarily for electronic design automation (EDA). The software is used mainly by electronic design engineers and electronic technicians to create electronic schematics and electronic prints for manufacturing printed circuit boards. OrCAD was taken over by Cadence Design Systems in 1999 and was integrated with Cadence Allegro since 2005[16].

OrCAD Capture and OrCAD PCB Designer were both used in conjunction to make the RFID tag. OrCAD Capture was used to design the circuit with the actual components on the tag. No simulations were done through OrCAD Capture, it was only used to create the PCB. After the circuit met all the design rules, it was net-listed and ported over to OrCAD PCB Designer.

OrCAD PCB Designer is a software that uses the net-list from OrCAD Capture along with user defined design rules to build the PCB. The design rules were created from the PCB manufacturer's website. The software used the actual footprints from all the components used in the RFID tag and all the routing was manually done. PCB Designer was used to produce all the GERBER files required from the PCB manufacturer. Again, no EM simulations were performed using this software which resulted in some peculiar behavior after the PCB was fabricated.

2.4 Code Composer Studio

Code Composer Studio (CCStudio or CCS) is an integrated development environment (IDE) to develop applications for Texas Instruments (TI) embedded processors[17]. CCS is a C# based IDE. It was used to program and flash the msp430 board. CCS allows for specialized functions and commands for each MSP430 board. In particular, functions were used to allow for low power operation of the MSP430 board, this in turn saved about 100x times the power. It allowed for the micro-controller to fall asleep for a determined amount of time until the output needed to be switched. It also had commands to set the msp's clock frequency which helped to have strong control over the data rate to allow for ease in the demodulation scheme.

2.5 Binary Viewer

Binary Viewer is a free software that allows for ease in viewing files in a binary format[18]. This was used to visualize the received data and ensure no errors occurred. The data received from the Reader is saved as a text file using GNU Radio, and this was opened in binary Viewer.

2.6 MATLAB

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python[19]. MATLAB was used to calculate the bit error rate. The Code used is attached to appendix B. It searches for 10,000 consecutive bits and compares them to the desired output then outputs the percent error rate.

2.7 Hardware

The hardware used in this thesis was for measurement purposes to assist in design as well as confirming the operation of the tag and reader. The main equipment used was a Network Analyzer, specifically the Agilent N5230C. This assisted in characterizing the impedance of the RFID tag at a variety of frequencies. It helped determine the parasitics and phase shift through the tag. The other main piece of equipment used was a spectrum analyzer. The one used in this thesis was the Agilent E4440A. It was used to characterize the RFID reader. Connecting the Reader to the spectrum analyzer showed the power being transmitted as well as the noise floor of the room. It also helped visualize the wave output from the reader which is modulated by the tag. These measurements helped determine the theoretical max read range as well as how the wave should look after modulation from the tag. The final equipment used was a source meter to measure the power output of the tag.

CHAPTER 3: RFID TAG

The RFID tag went through several iterations before it reached the final product. The basic concept of an RFID tag using backscatter is just a switch to transition from short to open and vice versa. This results in the transmission wave being reflected at 0 degrees or 180 degrees respectively. The wave can then be demodulated by the reader using a psk demodulation algorithm as well as some signal conditioning to improve read range.

3.1 RF switch

The switch is the main component of the RFID tag. It is the component that modulates the RF signal to transmit the data. It accomplishes this by opening and closing the switch. The specific switch used was a MASWSS0179. It is a SPDT switch that works off a 3V supply. It operates from DC to 2GHz and is a .5-micron GaAs pHEMT process. This specific switch has a dc short circuit impedance of 3 ohms and an open circuit impedance of 11k ohms.

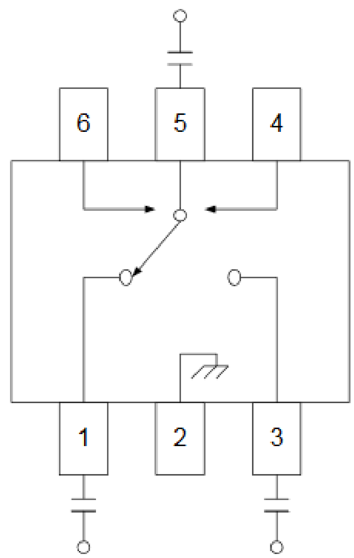


Figure 3.1: Switch diagram[20].

The switch block diagram is shown above. It shows how the control voltages are operated as well as the terminations. Pin 4 and 6 are the control voltage inputs. They must have at least a 3V differential to control the switch. The greater differential voltage results in a better connection with the termination. Pin 5 is the input RF signal. Pins 1 and 3 are the two states the switch can terminate to. Each RF port has to have a dc blocking capacitor as shown in the figure. Pin 2 is just the grounding pin.

3.2 Matching Network

The matching network was designed with Keysight ADS. It is an L-match with the ability to switch to a series inductor. This requires a second switch to be controlled in conjunction with the main RF switch. The components were chosen to emphasize an open and a short regardless of the load. This means that the state of the main RF switch doesn't affect the impedance that much. It just serves to perfect the impedance seen from the antenna which is demonstrated in the "Results" chapter.

3.3 PCB

The PCB was ordered from Sierra circuits. It was designed to allow for a lot customization with a variety of component values as well as the ability to short the footprints of any component. Unfortunately there is a significant parasitic capacitance and inductance resulting from the transmission lines which was not initially accounted for. This resulted it affecting the open condition impedance. The open circuit impedance is combined in parallel to lower the the impedance and rotate it on the smith chart.

The other aspect of the PCB is the length. At the time of designing the PCB, it was calculated to have a distance from the antenna to the main RF switch of about 1/12th of a wavelength so the reflections onto the smith chart would only rotate it slightly. However, FR-4 changes the wavelength by a factor of 2, so the distances is actually closer to just under 1/6th of a wavelength. That means the round trip back to the antenna is about 1/3rd of a wavelength resulting in a significant rotation on the smith chart. Along with the parasitic capacitance and inductance from the transmission lines, the RFID tag actually shows an open circuit as a short and a short as an open circuit. This just means the data is reversed and was accounted for by the reader.

3.4 MSP430

The MSP430 board chosen for this design was an ez430-f2013. The msp board was chosen because it has a target board that can be flashed with data and connected to the RFID tag. It also has multiple low power settings available to use. Low power mode 1 was used because any lower settings required an external clock. This was not implemented so that the RFID tag footprint could be smaller. The power could be lowered by a factor of 10-100 in the lower power modes.

The MSP board was programmed with Code Composer Studio with 4 output pins.

These are alternated in a certain pattern to control the two switches, since the switches need to be controlled by differential voltages. The data rate is set at 5kHz and clock speed is 1MHz.

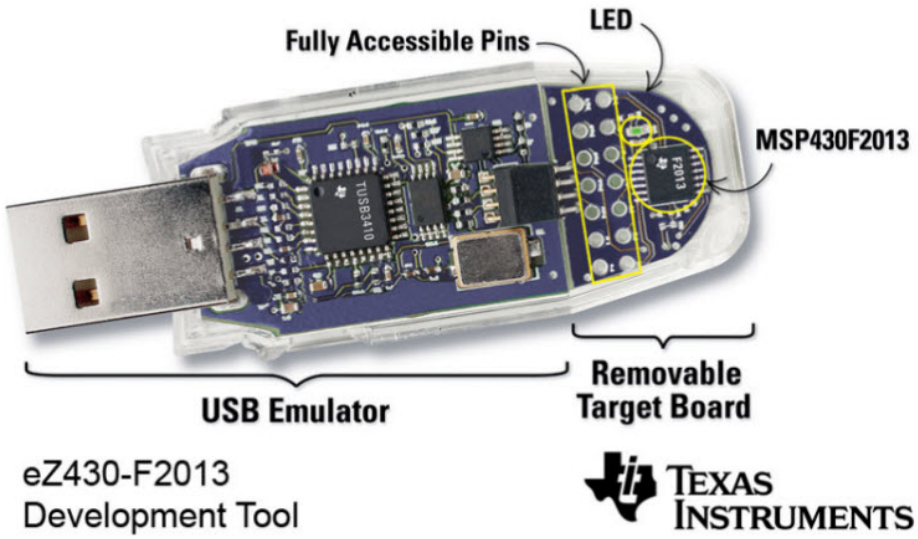


Figure 3.2: MSP430 board used[21].

CHAPTER 4: RFID Reader

The RFID Reader used in this thesis is an active reader. It sends out a RF signal to be modulated and back-scattered by the tag and received again by the reader. This signal is then demodulated in the reader and saved as a text file. The hardware used for the Reader is a USRP and requires a computer connection to operate. The software commands for the USRP are run from GNU Radio companion.

4.1 USRP

The USRP used for the RFID Reader is an Ettus Research B210 USRP board. This board is capable of RF ranges from 70MHz to 6 GHz with a 12 bit ADC/DAC. It can be programmed with GNU Radio, C++, or Python through USB 3.0 interface. It has 2 TX and 2 RX ports, with 2x2 MIMO capability. The board contains a Xilinx Spartan 6 XC6SLX150 FPGA and a sample rate of 61.44 MS/s[22].

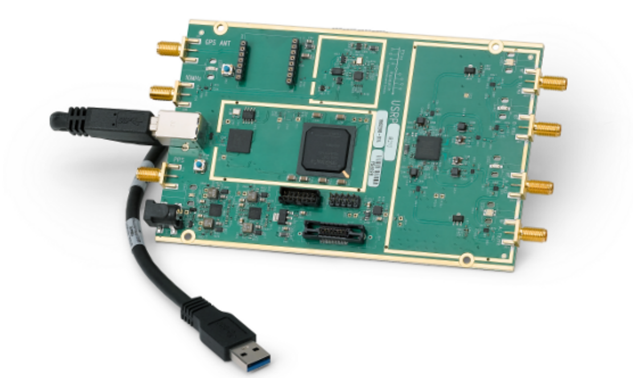


Figure 4.1: USRP Board Picture[23].

The front end of the USRP works as shown in the following figure. It converts the signal into an I and a Q value for in phase and out of phase measurements. This characteristic is why PSK was chosen as the modulating scheme.

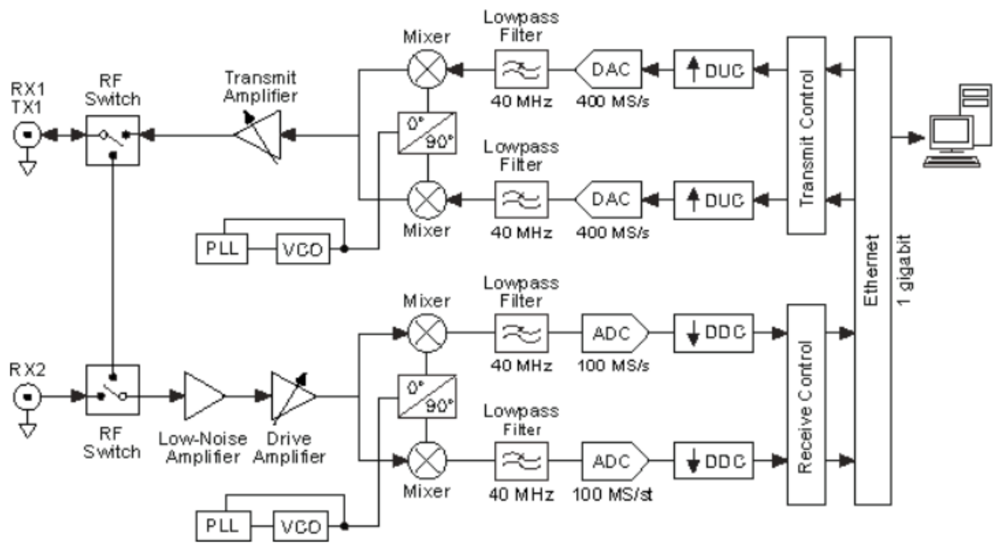


Figure 4.2: USRP front end demonstrated[24].

It consists of the IF frequency being created in the computer. Then goes to a digital up converter and a digital to analog converter. This signal is filtered then sent through a mixer to create the transmission signal at 915 MHz. The return path is just the opposite process. The receive signal is mixed down with an in-phase and a 90 degree shift to create the I-Q plot used for demodulation. The figure below shows the difference between back-scatter, ASK, and PSK.

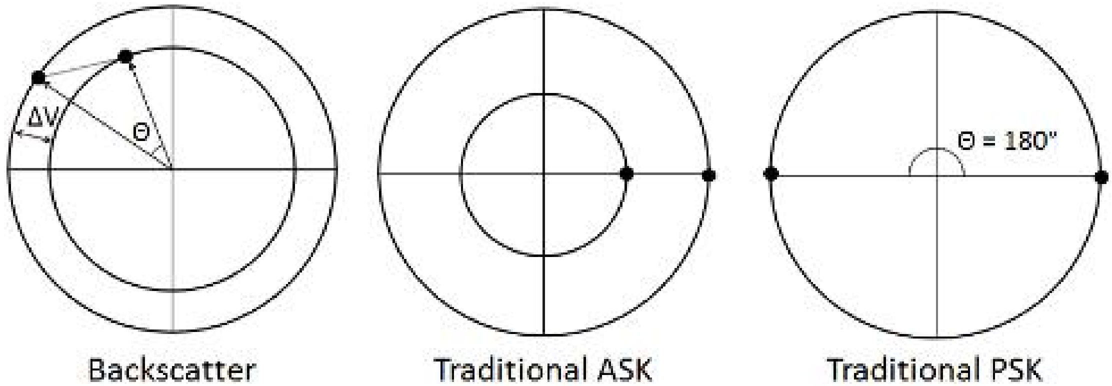


Figure 4.3: An example of the Q-I plot resulting from back-scatter[5].

This signal has to be passed through a high-pass filter in the first step to be demodulated.

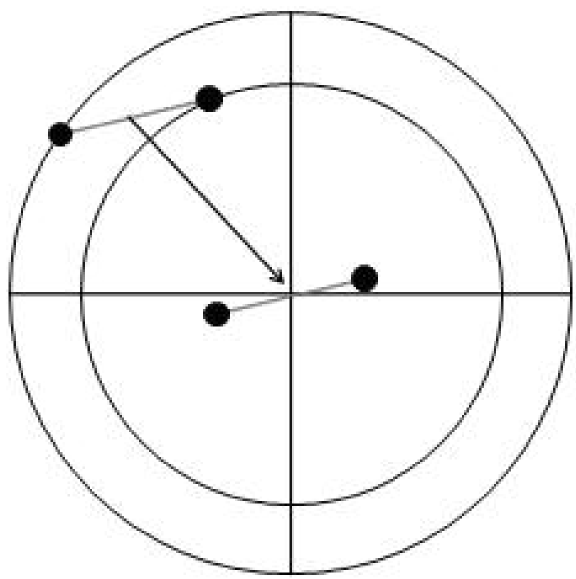


Figure 4.4: An example of the Q-I plot resulting from the filtered back-scatter signal[5].

A PSK algorithm performed on this results in mapping a one or zero to each constellation point.

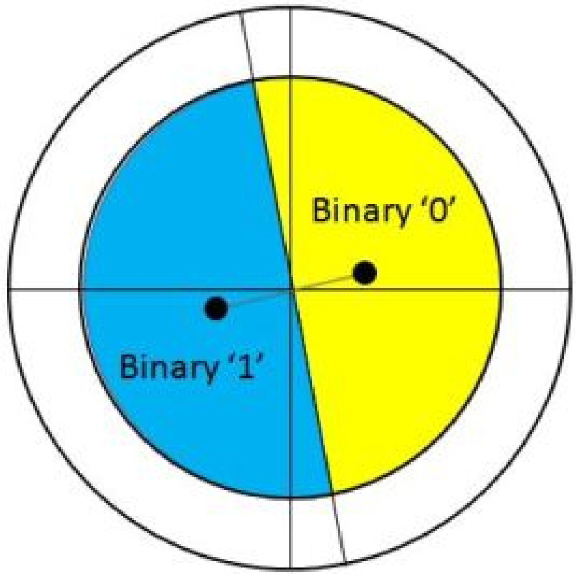


Figure 4.5: An example of the Q-I plot with mapping performed[5].

4.2 GNU Radio code

GNU Radio Companion was used to program the USRP board. It follows the flow chart in figure 4.6. GNU Radio companion allows for programming through a use of blocks similar to a flow chart. These blocks are capable of anything from logic to signal conditioning as well as real time monitoring. This flow graph is shown below:

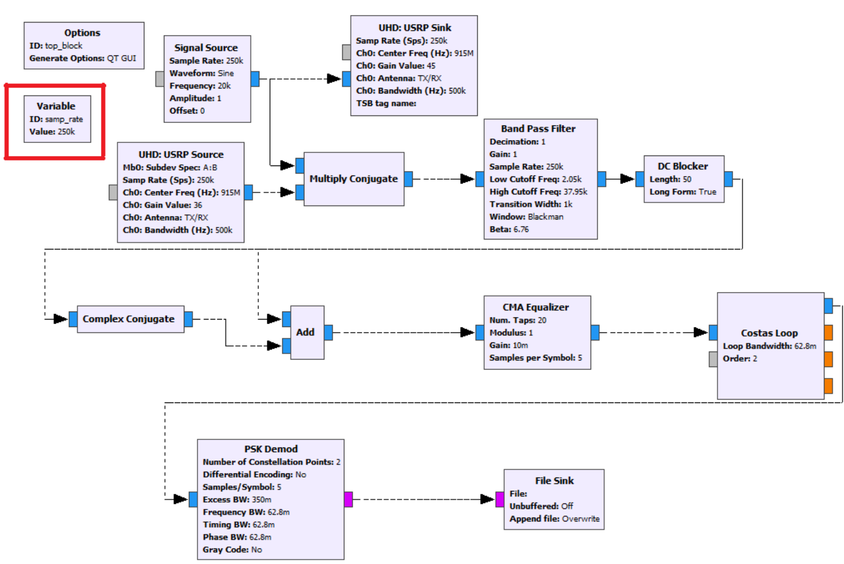


Figure 4.6: GNU Radio code sample rate block.

The first design decision made for this code was the sampling rate. 250k samples per second was chosen due to hardware restrictions. When the code was initially written, it proved to be demanding on the laptop it was running off of so the lowest possible sample rate was used.

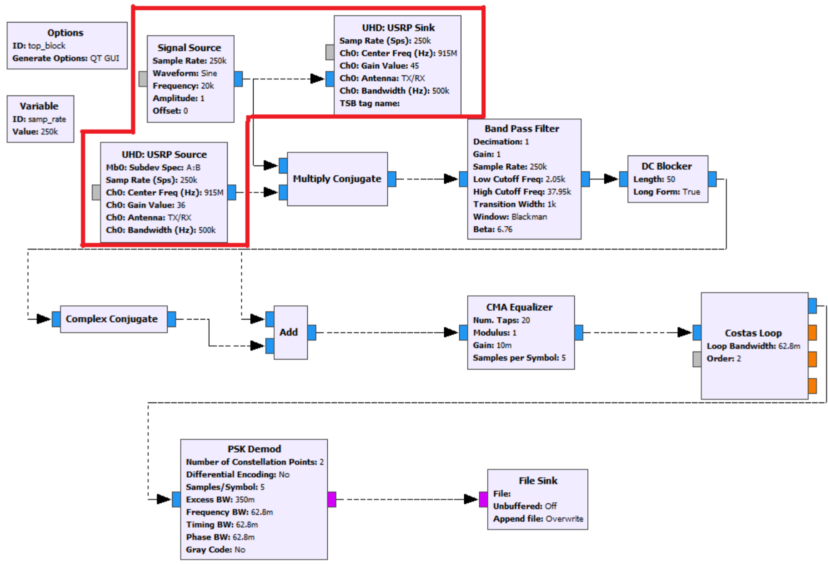


Figure 4.7: GNU Radio Code for transmitting and receiving

Next the intermediate frequency(IF) had to be chosen and generated. The design decisions for choosing the intermediate frequency consisted of making it small enough that the sample rate will take enough samples to get accurate data, as well as large enough to avoid 1/f noise. This resulted in choosing a 20kHz intermediate frequency, which is still low but it was at least 10x smaller than the sample rate so that criteria was deemed more important. The IF is generated in the signal source block.

The IF signal then goes to the transmitter through the USRP sink block. It up-converts the digital 20kHz IF to an analog 915MHz signal. 915MHz was chosen because it is in the ISM band. The receiver, USRP Source, picks up near 915MHz frequencies and down-converts them back to the digital intermediate frequency. This signal is sent through the Multiply Conjugate block with the original IF signal, this converts the modulated IF signal down to base-band and so the two states for the

Q-I plot is centered around 0. It accomplishes this because the multiply conjugate creates a sum and a difference, so it becomes centered around 0 and 40kHz.

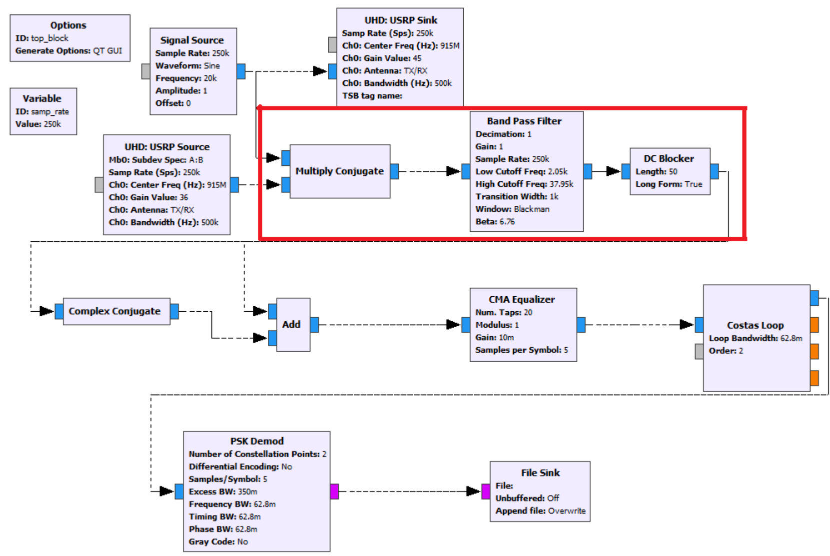


Figure 4.8: GNU Radio Code for filtering

The next block is a band-pass filter, with the pass-band between DC and the sum (40kHz) which just allows the modulated frequencies to remain. The following block is a second DC filter, it is required because the direct path between the TX and RX antennas is very strong to it continues to show up through the first filter.

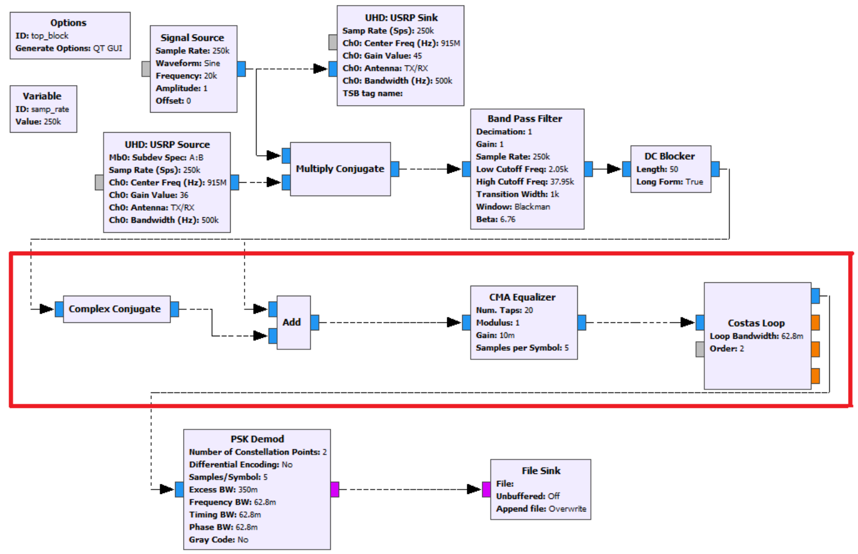


Figure 4.9: GNU Radio Code for signal conditioning

The blocks highlighted above are intended as signal conditioning to improve the results from the demodulation. The "Complex Conjugate" block and the "add" block are both tied to the output of the DC Blocker. They add the complex conjugate of the signal to itself. This operation improves the SNR greatly and moves the constellation points to the real axis, which also helps the demodulation accuracy. This however does create a problem when the constellation points are switching between 90 degrees and 270 degrees because the real parts are zero and the imaginary parts cancel out. This occurs once every wave length so it leads to "dead" zones in operation. These are characterized in the "Results" section but these blocks were shown to improve the operation greatly for the most part so they were kept. Next is a Constant Modulus Algorithm(CMA) Equalizer block, which is an adaptive block to improve the signal received. It uses a blind algorithm over time to separate the noise from the two constellation points and provide a better estimation of the intended bits[25]. The main design variables in this block are the number of taps and the samples per symbol. Number of taps was chosen by keeping the number low enough to prevent the algorithm from being to demanding but high enough to provide accurate results.

This number was found to be 20 taps. The samples per symbol was determined from the sample rate and the data rate. With the sample rate at 250k samples per second and a data rate of 10k symbols per second, the samples per symbol equates to 25. Since a later block also down-samples the data, 5 was chosen. Next a Costas Loop block is used, which is a type of phase locked loop to further fine tune the constellation points, remove noise, and prevent random frequencies from rotating the constellation plot. It was made a 2nd order loop to allow for speed in the computation.

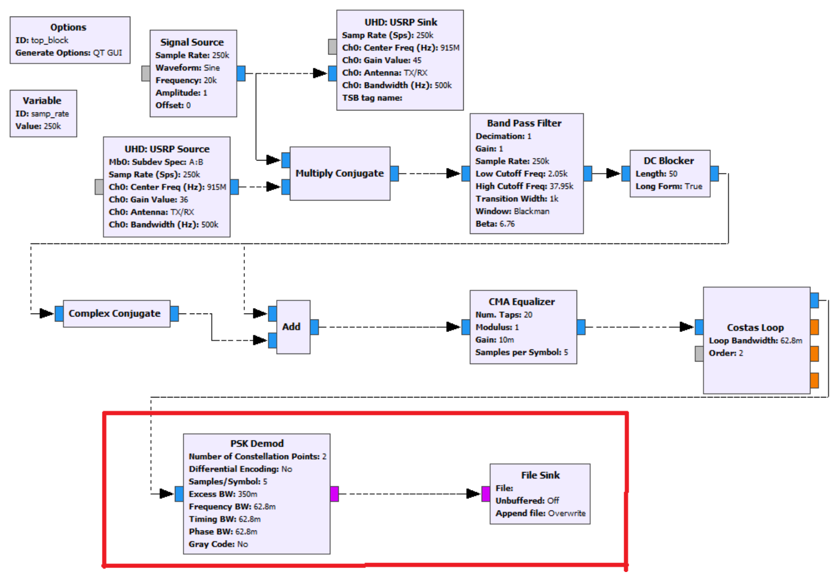


Figure 4.10: GNU Radio Code for demodulation

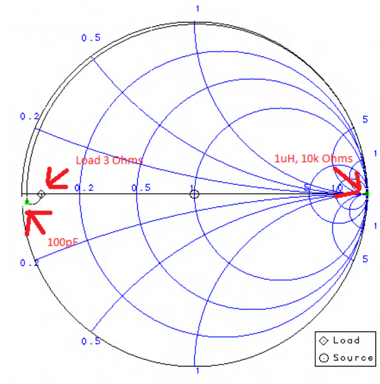
The final, and most important block, is the PSK demodulation block. It implements an algorithm to determine a threshold on the constellation plot and assigns a 1 or a 0 to each side of the threshold[25]. The only two variables changed in this block were the number of constellation points and the Samples per Symbol. The number of constellation points was chosen as 2 because it is BPSK so there are only two unique symbols. The Samples per Symbol was chosen as 5 for the same reason as in the CMA Equalizer. With the two combined, it results in a down sample by 25 so only 1 Sample per Symbol is left at the output. The output of this block is of type binary and gets saved to a file.

CHAPTER 5: Results

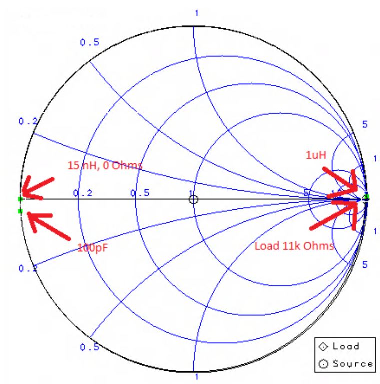
This chapter discusses the various design processes used as well as the results from the fabricated RFID tag and reader. It compares the simulated/calculated results to the tested results and explains any differences observed.

5.1 TAG Performance

The RFID tag was designed with a matching network to help get better reflections for the back-scattered signal. The switches impedance is 3 ohms when shorted and 11k ohms when open. This was determined using a multimeter and toggling the switch to the two distinct states. This method only measured the DC impedance of the switch. At 915MHz the impedance is transformed by the parasitics and distance from the the antenna. These values were not originally known and accounted for so a matching network was designed to try and be broad enough to not need those values. The matching network is an L-match with the ability to switch to a pi-match network. It was designed to convert an arbitrary value on the Smith Chart to an open or a short depending what mode the matching network is in. This was attempted by using large enough component values that the original impedance would play less of a role on the transformed impedance. This is demonstrated in the following images.



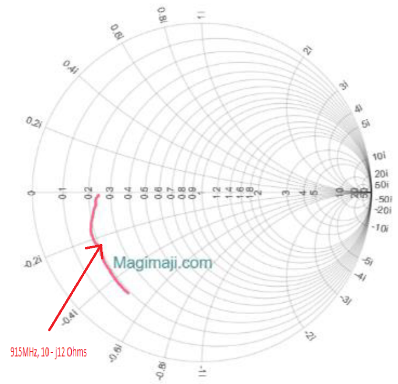
(a) ADS results of shorted switch with matching network



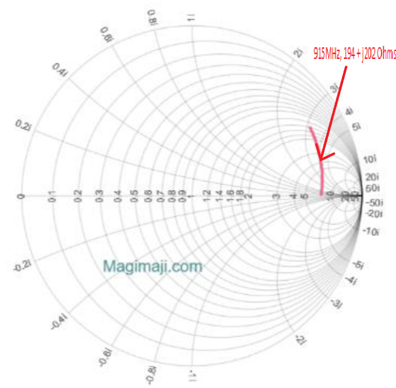
(b) ADS results of opened switch with matching network

Figure 5.1: ADS simulation of the closed and opened switch respectively with a matching network and no parasitics.

The starting point on both were 3 ohms and 11k ohms respectively. As shown on the first image, the L-match values are intended to transform the impedance to an open. The second image shows how the pi match transforms the impedance to a short no matter where the starting value is. The real world results from this topology are shown below, and the marked point corresponds to the "Load" from the simulation:



(a) Network Analyzer results of shorted switch with matching network[26]



(b) Network Analyzer results of opened switch with matching network[26]

Figure 5.2: Smith charts of the closed and opened switch respectively with a matching network.

The discrepancy seen is due to not accounting for parasitics and the wavelength within FR-4. Since they affect it along the transmission lines not the load like the network was originally designed for, the network didn't work as intended. A matching network was then recalculated with the math is shown below:

$$\lambda = C_0/(f * \sqrt{\epsilon}) \tag{5.1}$$

Where $\epsilon = 4$ in FR- 4, so the equation becomes:

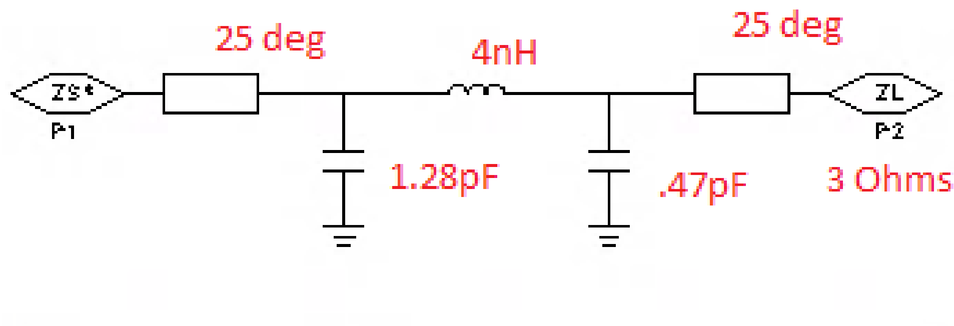
$$= 3e8/((915e6) * \sqrt{4}) \ = 16.39cm \tag{5.2}$$

This distance from the antenna to the the switch is about 3.1cm. This results in the phase shift shown below:

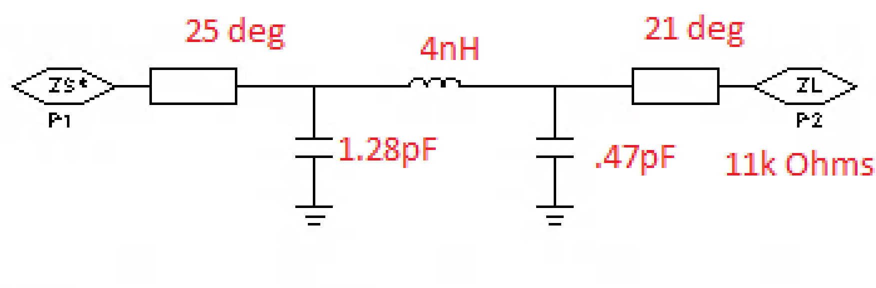
$$\phi = (3.1/16.39) * (2\pi) * (180/\pi) * 2\phi = 136^o \quad (5.3)$$

The extra multiplication of 2 at the end is because the wave travels to the switch and back, so the distance is doubled. Next the parasitic impedance's of the lines were to be accounted for. Using a microstrip calculator[27], it was found that the PCB used for the tag has a capacitance of $96\rho F/m$ and $.3\mu H/m$ for the 100 mil line. This means the first line has a capacitance of $1.28\rho F$ and 4nH of inductance. Using the microstrip calculator again[27], a 20 mil line has a capacitance of $47.5\rho F/m$ and $.6\mu H/m$. This results in a capacitance of $.47\rho F$ and inductance of 6nH.

The ADS simulation set up for this matching network is shown here:



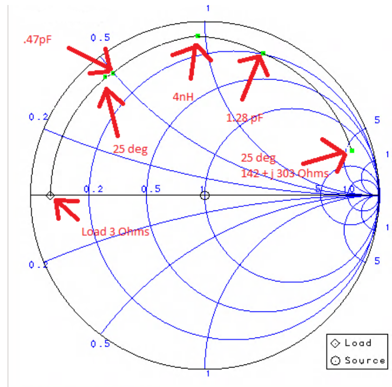
(a) ADS simulation set up, Short



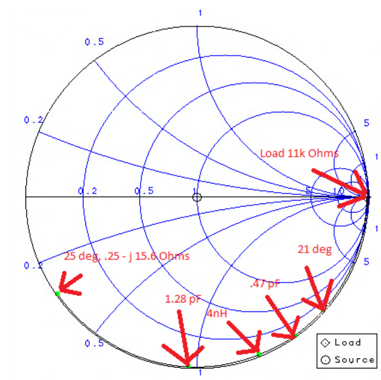
(b) ADS simulation set up, Open

Figure 5.3: ADS set up of the closed and opened switch respectively.

The only difference in parasitics between the two conditions is the first transmission line length. The transmission line terminates slightly farther out when the switch is closed than when it's open. Through the equations used above, the extra distance equates to a 4 deg greater shift. The ADS simulation of these conditions are shown below:



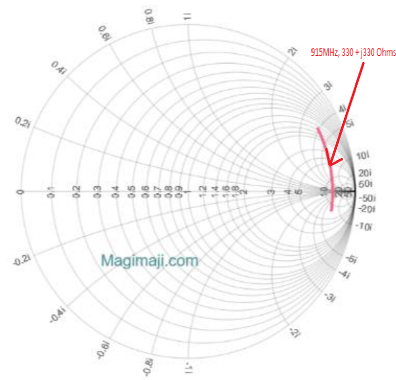
(a) ADS results of shorted switch



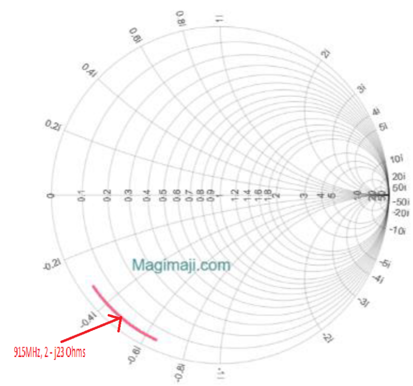
(b) ADS results of opened switch

Figure 5.4: Smith charts of the closed and opened switch respectively ADS simulation.

An RFID tag was assembled with just one switch as in the simulation above and was tested on a Network Analyzer. The results are shown here and again the marked point equates to the "Load" on the ADS simulation:



(a) Network Analyzer results of shorted switch[26]

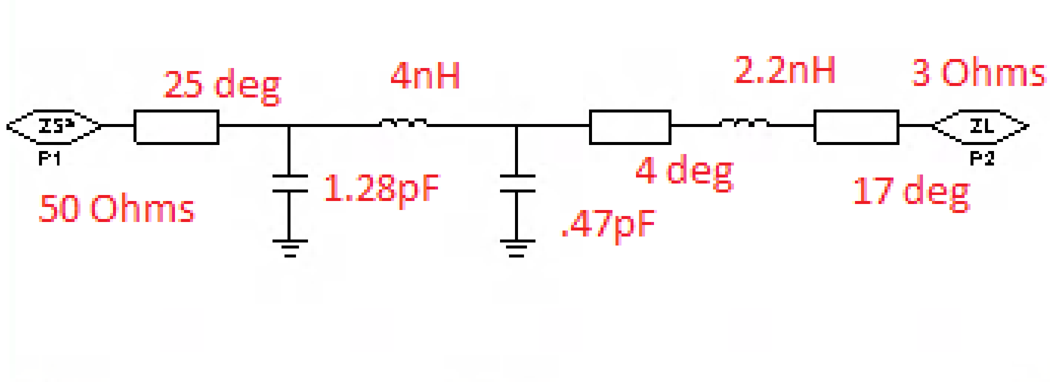


(b) Network Analyzer results of opened switch[26]

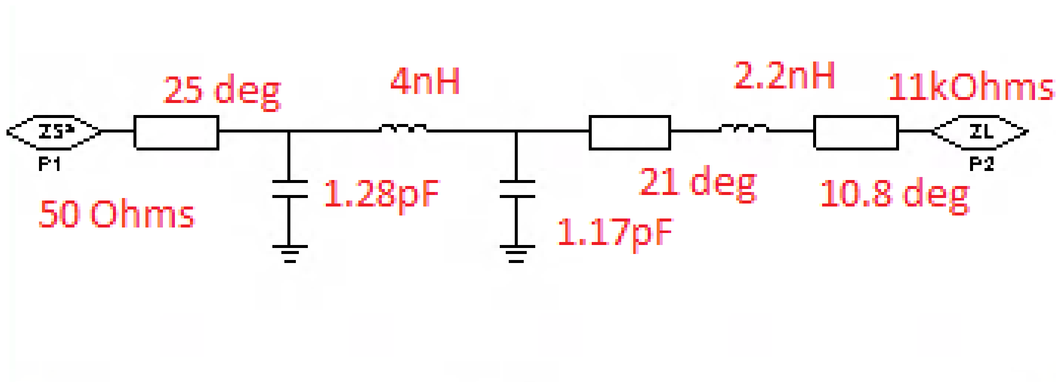
Figure 5.5: Smith charts of the closed and opened switch respectively.

This shows that the simulation is not perfect because some parasitics are neglected but it is close enough to being accurate while still keeping the math simple enough for an easy design process.

An improved matching network was created using this model through ADS. The set up is shown here:



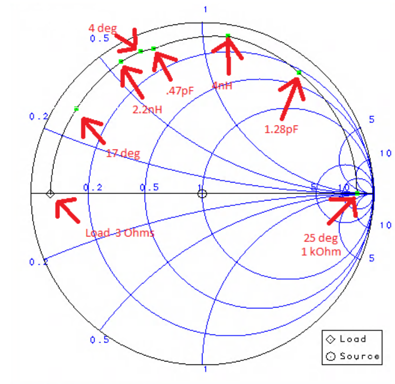
(a) ADS set up of shorted switch with improved matching network



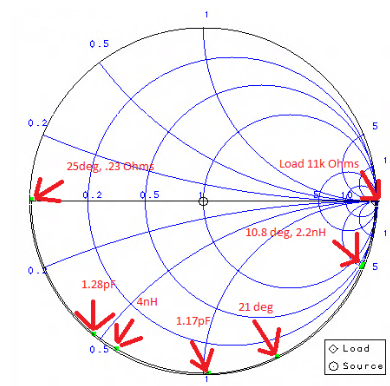
(b) ADS set up of opened switch with improved matching network

Figure 5.6: ADS set up of the closed and opened switch respectively with an improved matching network.

The components added to this matching network were the 2.2 nH inductor and a $.7\rho F$ capacitor in an L match configuration. A second switch was used here again to toggle the capacitor so the two states of the matching network are just a series inductor or an L match. The transmission line lengths also change due to the switches so the degree shifts are adjusted in each state. The ADS simulation of these set up are shown here:



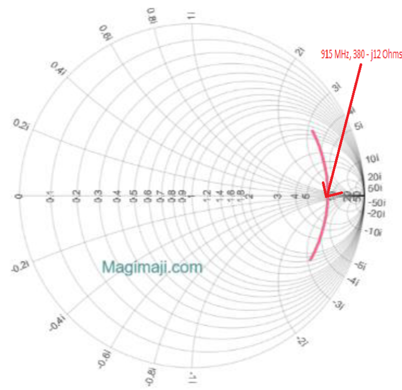
(a) ADS results of shorted switch with improved matching network



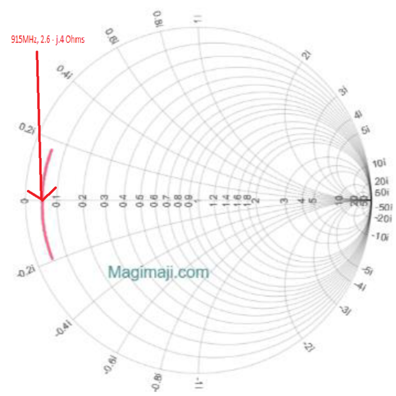
(b) ADS results of opened switch with improved matching network

Figure 5.7: Smith charts of the closed and opened switch respectively with an improved matching network using ADS.

And the lab results of this are shown here:



(a) Network Analyzer results of shorted switch with improved matching network[26]



(b) Network Analyzer results of opened switch with improved matching network [26]

Figure 5.8: Smith charts of the closed and opened switch respectively with an improved matching network.

Each of these RFID tags described above were tested with the reader. The table below demonstrates the bit error rate vs distance for each configuration:

Table 5.1: Tag performance over distance.

Distance	Switch	First Matching Network	Second Matching Network	Control
2ft	0%	0%	0%	48.07%
4ft	0%	0%	0%	48.07%
6ft	0%	0%	0%	48.07%
8ft	0%	0%	0%	48.07%
10ft	0%	0.12%	0%	48.07%
12ft	0%	0%	0.01%	48.07%
14ft	0.02%	0%	0.13%	48.07%
16ft	17.18%	48.34%	6.83%	48.07%
18ft	48.15%	48.02%	48.07%	48.07%

The above results were gathered by setting a bit pattern of alternating 1s and 0s continuously on the tag and tested with the reader for 15 seconds at each distance. The bits recovered by the reader were saved to a file and uploaded to Matlab, where a script was run that checked the pattern for 10000 sequential bits throughout the entire file. The lowest error rate was selected from this and output as a percentage from the Matlab script. This was chosen as the test method because ten thousand bits is a substantial amount more than will usually be transmitted by an RFID tag. So the error rate being low for that number of bits shows it will operate well in real world applications. As mentioned in the "RFID Reader" chapter, the complex conjugate addition step on GNU Radio caused errors when the constellation points were at 90 and 270 degrees, so each tag was moved forward or back word a few inches until the constellation points were at 0 and 180 degrees. This allowed for a best case result on all the tags and a fair comparison between them.

The "Control" section of the table was tested by running the reader without having the tag on to show what the output bits are in the test room itself. The results show that around 14 to 16 feet the tags begin to fail and drop off sharply after 16 feet. The results from the 16ft tests show that the improved matching network did provide better reflections so the error rate was lower. It also shows how the first matching network was actually worse than just a switch due to the impedances they switched between.

Next, the antennas were tested to see how they performed at different angles from the horizon. The results are shown below:

Table 5.2: Tag performance over antenna angle.

Distance	Switch	First Matching Network	Second Matching Network	Control
90^0	0%	0%	0%	48.07%
60^0	0%	0%	0%	48.07%
30^0	0%	0%	0%	48.07%
20^0	0%	48.19%	0.33%	48.07%
10^0	48.26%	48.53%	48.04%	48.07%
0^0	48.41%	48.38%	48.32%	48.07%

Each tag was tested only to provide an average cut off angle of the antenna due to the difficulty in control over this test. The tag was placed on a flat surface 4 feet away from the reader and tag's antenna was tested at the angles shown in the table. This shows that at around 10 to 20 degrees above the horizontal the antenna stops reflecting the RF wave. This is due to the use of monopole antennas which have a limited radiation pattern.

To further test the antennas, each tag was tested at different elevations in comparison to the Reader. The results are shown below:

Table 5.3: Tag performance over height.

Distance	Switch	First Matching Network	Second Matching Network	Control
0in	0%	0%	0%	48.07%
6in	0%	0%	0%	48.07%
12in	0%	0%	0%	48.07%
18in	0%	0%	0%	48.07%
24in	0%	0%	0%	48.07%
30in	0%	0%	0%	48.07%
36in	48.47%	48.61%	.01%	48.07%
42in	48.22%	48.56%	48.48%	48.07%

This test was performed by testing each tag at 4 ft from the reader and lowering the tag from level with the reader by the amount in the table. This test showed that at around 36in below the Reader, the tag stops reflecting the RF wave. Again this is due to the limited radiation pattern of the monopole antennas.

5.2 Reader Performance

The theoretical readers max range is calculated with Friis theorem here[28]:

$$P_r = (P_t * G_t * G_r * \lambda^2)/(4\pi R)^2 \tag{5.4}$$

Where P_r is power received, P_t is power transmitted, G_t is the transmitting antenna gain, G_r is the receiving antenna gain, λ is the wavelength, and R is the radius[28].

Reorganizing this equation results in:

$$R = \sqrt{(P_t * G_t * G_r * \lambda^2)/((4\pi)^2 * P_r)} \tag{5.5}$$

The power transmitted is -20dBm, the gain of the transmitting antenna and the receive antenna are both equal to 1, λ is 33cm, and P_r has to be at least -70dBm to be demodulated and discerned from the noise floor (the noise floor of the test room was determined through attaching an antenna to a spectrum analyzer and measuring the average power received at 800MHz to 1GHz with no external RF sources in the room). This results in approximately:

$$R = 10.4m \tag{5.6}$$

However, the wave has to travel there and back so the distance is halved. $R = 5.2m$ Which is equal to 17 feet. The max range actually achieved is just about equal to that. The loss in the power reflected by the tag accounts for the max range not exactly being achieved.

CHAPTER 6: CONCLUSIONS

In conclusion, this project was a successful implementation of the first steps towards an RFID system. This can be continued and improved in the future, for which many suggestions are discussed in the next chapter. This thesis confirmed many of the theories behind RFID communication from the range to the power output and bit error rate.

The results of this thesis have supported Friis theorem, back-scatter, PSK modulation, Matching networks, and transmission line impedance/ wavelength phase shifts along a Smith chart.

The goal of this thesis was to prove that RFID was a viable means of low power communication for laboratory data gathering. It was a stepping stone towards a more complete and efficient means to achieve the aforementioned. Both of these have been proven to be true by the completed work during this thesis. RFID communication was completely reliable at 8 feet away and was achieved at 16 feet away. The power from the tag was low at 600 μW and has the potential to achieve even lower power.

CHAPTER 7: FUTURE WORK

Possibilities for continuation of the work performed in this thesis is plentiful. Some of this will be completed by the beginning of January where as other work is just discussed as possibilities. Future work is necessary to improve the read range as well as lower the power consumption.

The first continuation of this is implementing an encoding scheme on the RFID tag. A chip has already been fabricated that uses a Reed-Solomon encoding scheme and back-scatter the signal using that. This encoding scheme improves bit error rate based off the encoding scheme. This will be tested shortly to prove that it works.

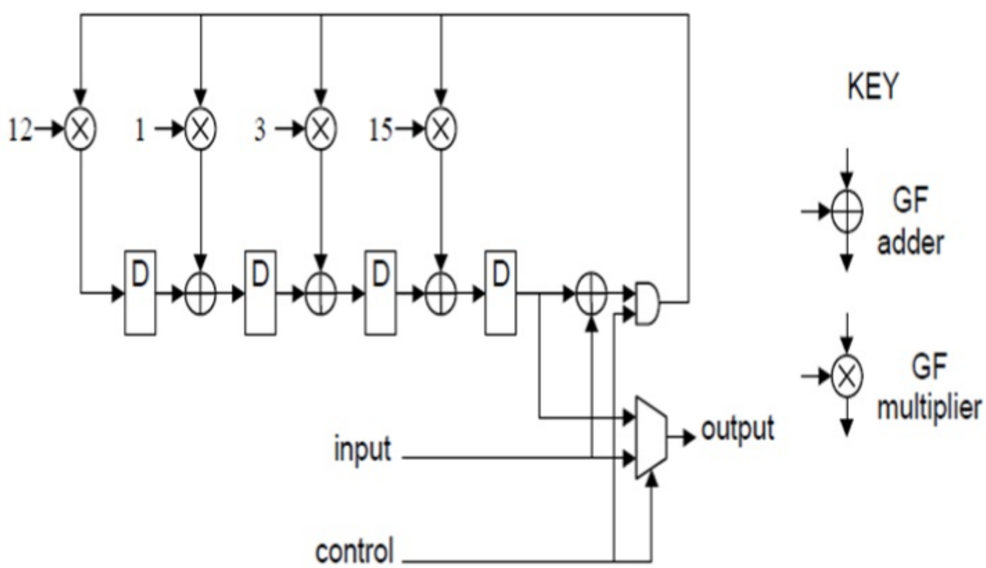


Figure 7.1: Example of a Reed-Solomon error correction chip[29].

Another possibility for future work is improving the matching network. A greater difference in impedance, as shown in the results section, would create less power loss in the tag itself and allow for maximum distance to be achieved. This can be done by focusing more on the parasitics of each component as well as the transmission lines and lengths on the tag itself. This will improve the error rate in closer ranges because the difference in phase will also be greater.

Making the tag into a passive tag is also recommended. This just involves adding a power conversion circuit to the tag. It will require 10x more power from the reader but it will allow the tag to be much smaller[2]. A smaller tag means it can be used for a larger variety of applications.

Designing or purchasing new antennas would make a significant difference towards versatility as well. The dipole antennas used in this implementation prohibit 360 degree reception. This means it's most effective when there is direct line of sight as well as being level with the reader. Better antennas will allow for the tag to be moving in any direction around the reader and still receive the bits.

To allow for lower power operation, an off chip oscillator could be used as the clock for the micro-controller. This will allow for the lower power modes of the micro-controller to be used which consume up to 100x less power than the mode used in this thesis.

Another recommendation would be to improve upon the complex conjugate addition step of the GNU Radio code. Using an improved method of accomplishing the same result could remove the "Dead" zones the tag currently experiences.

One possible application could be implementing three readers and using beamforming to triangulate the tags signal for tracking. This can be useful for many research purposes.

REFERENCES

[1] ABR, "What is rfid and how does rfid work?," 2018. Last retrieved 2018-11-26.

[2] Wikipedia, "Radio-frequency identification," January 2003. Last retrieved 2018-11-05.

[3] B. Violino, "The history of rfid technology," Jan 2005. Last retrieved 2018-11-26.

[4] C. Finch, "Advantages disadvantages of rfid." Last retrieved 2018-11-26.

[5] N. Instruments, "Advanced rfid measurements: Basic theory to protocol conformance test," Dec 2013. Last retrieved 2018-11-26.

[6] Wikipedia, "Phase-shift keying." Last retrieved 2018-11-26.

[7] M. B. D. Wetherall, "A software radio-based uhf rfid reader for phy/mac experimentation," April 2011. Last retrieved 2018-12-6.

[8] M. F. G. Smietanka, S. Brato and J. Gotze, "Implementation and extension of a gnu-radio rfid reader," 2013. Last retrieved 2018-12-6.

[9] L. Catarinucci, D. D. Donno, R. Colella, F. Ricciato, and L. Tarricone, "A cost-effective sdr platform for performance characterization of rfid tags," April 2012.

[10] D. D. Donno, F. Ricciato, and L. Tarricone, "Listening to tags: Uplink rfid measurements with an open-source software-defined radio tool," Jan 2013.

[11] A. Briand, B. B. Albert, and E. Gurjao, "Complete software defined rfid system using gnu radio," 11 2012.

[12] A. Bothe, C. Schraeder, and N. Aschenbruck, "An uhf rfid performance evaluation architecture based on traces from a software defined transceiver," Sept 2014.

[13] D. Y. A. S. J. Smith, "Wisp: A passively powered uhf rfid tag with sensing and computation," March 2008. Last retrieved 2018-12-6.

[14] R. Stallman, "Gnu general public license, version 3," June 2007. Last retrieved 2018-11-02.

[15] K. of EEsof EDA, "Advanced design system," 1985. Last retrieved 2018-11-02.

[16] C. D. Systems, "Orcad," 1985. Last retrieved 2018-11-02.

[17] T. Instruments, "Code composer studio," 1999. Last retrieved 2018-11-02.

[18] Softonic, "Binary viewer," January 2017. Last retrieved 2018-11-02.

[19] MathWorks, "Matlab," September 2018. Last retrieved 2018-11-26.

[20] MACOM, "Maswss0179 3 v gaas spdt switch." Last retrieved 2018-11-26.

[21] T. Instruments, “Msp430 usb stick development tool ez430-f2013.” Last retrieved 2018-11-26.

[22] E. Research, “Usrp b200/b210 bus series,” April 2016. Last retrieved 2018-11-05.

[23] E. Research, “Usrp b210 (board only).” Last retrieved 2018-11-26.

[24] E. Research, “About usrp bandwidths and sampling rates,” May 2016. Last retrieved 2018-11-26.

[25] Wikipidia, “Guided tutorial psk demodulation,” September 2018. Last retrieved 2018-11-26.

[26] magimaji.com, “Online touchstone file viewer,” 2018. Last retrieved 2018-11-05.

[27] N. Asuni, “Pcb impedance and capacitance of microstrip,” 2018. Last retrieved 2018-11-05.

[28] E. RF, “Friis tranmission calculator,” 2015. Last retrieved 2018-11-26.

[29] EmbDev.net, “Vhdl implementation of reed solomon encoder,” May 2012. Last retrieved 2018-11-26.

APPENDIX A: MSP430 Code

```
#include <msp430.h>
#include <stdio.h>
#include<stdlib.h>
void timer_init(unsigned long us);

unsigned int data[16] = { 0x06, 0x09, 0x06, 0x09, 0x06, 0x09, 0x06, 0x09, 0x06, 0x09, 0x06, 0x09, 0x06,
0x09, 0x06, 0x09};
unsigned int mcount = sizeof(data)/sizeof(data[0])-1;
unsigned int j = 0;
unsigned long counter = 0;

void main(void)
{

    WDTCTL = WDTPW+WDTHOLD;;   // stop watchdog timer
    if (CALBC1_1MHZ==0xFF)                    // If calibration constants erased
      {
        while(1);                               // do not load, trap CPU!!
      }
    DCOCTL = 0;                               // Select lowest DCOx and MODx settings
    BCSCTL1 = CALBC1_1MHZ;                    // Set range
    DCOCTL = CALDCO_1MHZ;                     // Set DCO step + modulation
    P1DIR |= 0xff;
    P1OUT = 0x00;

    timer_init(100);
    while(1)
    {
        _bis_SR_register(LPM1_bits + GIE);
    }
}

void timer_init(unsigned long us)
{
    CCTL0 = CCIE;                             // CCR0 interrupt enabled
    CCR0 = us;
    TACTL = TASSEL_2 + MC_1;                  // SMCLK, Upmode
}


#if defined(__TI_COMPILER_VERSION__) || defined(__IAR_SYSTEMS_ICC__)
#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A(void)
#elif defined(__GNUC__)
void __attribute__ ((interrupt(TIMERA1_VECTOR))) Timer_A (void)
#else
#error Compiler not supported!
#endif
{
        if(j < mcount)
        {
            P1OUT = data[j];
            j++;
        }
        else if(j == mcount)
        {
            P1OUT = data[j];
            j = 0;
        }
        else
        {
            j = 0;
        }
}
```

Figure A.1: MSP430 code from code composer studio.

APPENDIX B: Matlab Code

```
clear;
clc;
fileName = "Switch_error_2ft_const";
fileID = fopen(fileName);

A = fread(fileID);
B = ones(10000,1);
for k = 1: length(B)
    if mod(k, 2) == 0
        B(k) = 0;
    end
end
j = 1;
error = zeros(length(A)-10001,1);
while(j < length(A)-10000)
    for x = 1:10000
        if ((A(j+x,1) - B(x,1)) == 0)
            continue;
        else
            error(j) = error(j) + 1;
        end
     end
    j = j+1;
end

[c,idx] = min(error);

percent_error = c/100;
fclose('all');
```

Figure B.1: Matlab Code for calculating error rate.