PART 15 COMPLIANT FREQUENCY HOPPING BACKSCATTER COMMUNICATION AT 5.8 GHZ

A Thesis
Presented to
The Academic Faculty

by

Robert W. Corless

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2018

PART 15 COMPLIANT FREQUENCY HOPPING BACKSCATTER COMMUNICATION AT 5.8 GHZ

Approved by:

Dr. Gregory D. Durgin, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Andrew F. Peterson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mary Ann Weitnauer
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 26, 2018

ACKNOWLEDGEMENTS

First and foremost, I want to thank my advisor Professor Gregory Durgin for giving me the opportunity to study in The Propagation Group. You have taught me a tremendous amount about engineering design and problem solving that will help me learn and grow as an engineer. Your leadership of The Propagation Group has created a friendly and positive learning environment where ideas are shared freely and the members willingly provide their time to help each other.

To the members of The Propagation Group—Mohammed, Cheng, Mike, Francesco, Joanna, and Eric—thank you for making me feel at home and for fostering a positive, learning atmosphere. I especially want to thank Francesco for taking the time to teach me about R.E.S.T. 1.0's microcontroller programming and Cheng for your invaluable discussions in developing R.E.S.T. 2.0's frequency hopping microcontroller code.

To my committee members—Professor Andrew Peterson and Professor Mary Ann Weitnauer—thank you for providing your invaluable time and knowledgeable feedback in the completion of this work.

Finally, to my wife and children—Christina, Max, and Aiden—thank you for your patience and support throughout this journey. You provided the purpose and motivation to focus on the small steps in the completion of this venture.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

B	Channel bandwidth (Hz)
d	Distance between transmitter and receiver (m)
G_r	Receiver antenna gain (dB)
G_t	Transmitter antenna gain (dB)
M	Number of frequencies in a hopset
P_r	Received signal power (dBm)
P_t	Transmit signal power (dBm)
T_c	Chipping duration (s)
T_h	Hop duration (s)
T_s	Data symbol duration (s)
W	Hopping band (Hz)

ADC	Analog to Digital Conversion
BPSK	Binary Phase Shift Keying
CCS	Code Composer Studio
CFR	Code of Federal Regulations
CRC	Cyclic Redundancy Check
CSMA-CA	Carrier Sense Multiple Access – Collision Avoidance
CW	Continuous Wave
DAC	Digital to Analog Conversion
DBPSK	Differential Binary Phase Shift Keying
DSSS	Direct Sequence Spread Spectrum

DUT	Device Under Test
EIRP	Equivalent Isotropically Radiated Power
FCC	Federal Communications Commission
FET	Field Effect Transistor
FHSS	Frequency Hopping Spread Spectrum
FPGA	Field Programmable Gate Array
GI	Guard Interval
GND	Ground
GPS	Global Positioning System
GTX	Georgia Tech eXperimental Air Interface
I	In Phase
I2C	Inter Integrated Circuit
ISM	Industrial, Scientific, and Medical
ISR	Interrupt Service Routine
KDB	Knowledge Database
LFRX	Low Frequency Receiver
LO	Local Oscillator
L&T	Localization and Tracking
LSB	Least Significant Bit
MAI	Multiple Access Environments
MISO	Master In Slave Out
MOSI	Master Out Slave In
MSB	Most Significant Bit
OET	Office of Engineering and Technology
OOK	On Off Keying

PC	Personal Computer
PG	Processing Gain
PLL	Phase-Locked Loop
PN	Pseudo-Noise
PWM	Pulse Width Modulation
Q	Quadrature
QTR	Quantum Tunnel Reflector
RBW	Resolution Bandwidth
R.E.S.T.	RFID-Enabled Sensing Testbed
RF	Radio Frequency
RFID	Radio Frequency Identification
SCLK	Serial Clock
SDR	Software Defined Radios
SPDT	Single Pole Double Throw
SPI	Serial Peripheral Interface
SS	Slave Select
TI	Texas Instruments
TICS	TI Clocks and Synthesizers
UHF	Ultra High Frequency
USRP	Universal Software Radio Peripheral
USB	Universal Serial Bus
VBW	Video Bandwidth
VHF	Very High Frequency
VCO	Voltage Controlled Oscillator
W	Watt

SUMMARY

The expansion of Internet of Things (IoT) devices continues to increase year after year, and as more devices and networks are employed, overused unlicensed spectrums become more and more congested. A sensible solution to this congestion is to use adaptive interference-rejection protocols and techniques that operate on under-utilized, unlicensed spectrums thereby making the overall use of all unlicensed bands more efficient. New advances in hardware have shown significant increases in backscatter communication read range without increasing the power needed for communication; therefore, the exponential expansion of IoT deployments coupled with congested low-frequency Industrial, Scientific, and Medical (ISM) bands and these new advances in hardware make backscatter communication a more attractive choice for IoT solutions. Regulatory requirements mandate the implementation of spread spectrum communication techniques for any intentional radiators operating in unlicensed ISM bands to increase collision avoidance and ensure the band is shared fairly among all devices; thus, knowledge of and familiarity with Part 15 compliance testing is essential for IoT hardware engineers designing systems that operate in ISM bands.

The Georgia Institute of Technology Propagation Group has a custom-built 5.8 GHz backscatter communication system that uses semi-passive and passive radio frequency identification (RFID) tags for enhanced backscatter capabilities and energy harvesting research, respectively. The reader has undergone three updates with the fourth update currently under development. In the third version, the RFID-Enabled Sensing Testbed (R.E.S.T.), the reader design focused on a flexible hardware and software solution

that utilized interchangeable daughterboards with a frequency-hopping programmable radio frequency (RF) front end. The fourth version, R.E.S.T. 2.0, seeks to extend this flexible hardware and software solution using off-the-shelf microcontroller development kits and RF synthesizer evaluation boards, which significantly decrease the system's size and power requirements.

Since R.E.S.T. operates in the 5.8 GHz unlicensed ISM Band (5775 – 5850 MHz), it must incorporate spread spectrum communications to mitigate interference as outlined in the Code of Federal Regulations, Title 47, Part 15, sub-part C.247 (herein referred to as Part 15). This research will summarize the history of the Propagation Group backscatter communications system, cover the basics of backscatter communication and spread spectrum communications, and then design a frequency hopping protocol for the R.E.S.T. 2.0 system followed by RF measurements and testing to confirm the system satisfies Part 15 requirements.

CHAPTER 1. INTRODUCTION

1.1 Research Motivation

Every year, Internet of Things (IoT) solutions make the modern business process more efficient by automating supply chain management, streamlining data collection, improving worker safety, and enhancing customers' shopping experience just to name a few [1]. In fact, a recent IDC study indicated that by 2020 the IoT industry would reach an estimated $1.7Trillion with over one-third of that, the largest of all categories surveyed, going to hardware [1]. The hardware aspect of IoT encompasses the sensors and "smart" devices used to collect the data that enable business process efficiencies. IoT devices come in all shapes and sizes; however, three major aspects that dominate a sensor's effectiveness are its battery life, network connectivity, and communications range [2]. Common IoT hardware solutions frequently use low-power components coupled with low-power protocols that typically operate in unlicensed ISM bands. Two important considerations in using ISM bands are that they often become congested when several technologies that use them achieve widespread adoption, especially when unlicensed worldwide, and that devices operating in ISM bands must employ spread spectrum techniques for interference mitigation. Backscatter communication is a technique that can address the battery life aspect that dominates IoT devices, and recent hardware advances in backscatter communications research have demonstrated a ten-fold increase in the read range with no increase in power consumption by using quantum tunnel reflectors (QTRs), which makes backscatter communication a more attractive solution for future IoT applications [3].

This project has two primary motivations. The first is to develop a frequency hopping spread spectrum protocol to enable the next generation of 5.8 GHz backscatter communications research at Georgia Tech, and the second is to explore the requirements associated with Federal Communications Commission (FCC) Part 15 frequency hopping compliance testing. The Georgia Tech R.E.S.T. 1.0 system has a frequency hopping capability, but its design cannot support research in space-constrained, stand-alone, low-power situations; therefore, the R.E.S.T. 2.0 system seeks to meet this requirement using a low-cost, off the shelf, low power microcontroller and RF synthesizer.

1.2 Backscatter Communication Basics

In a traditional narrowband communication system, two transceivers—a transmitter and receiver in a single unit—each with their own dedicated power source communicate using a common frequency. In contrast, a backscatter communication system uses only a single transceiver, typically called a reader, to communicate with a transponder—a transmit responder—commonly known as a Radio Frequency (RF) tag, which has no dedicated power source. In propagation-based backscatter communication, the tag converts a continuous wave (CW) signal from the reader into a temporary power source to enable RF communication. To convert the CW signal, the tag uses a diode to restrict the current flow to a single direction rectifying the received voltage with a storage capacitor to create a steady voltage to power the tag's circuitry [4][5]. This method of powering the tag makes backscatter communication systems an inherently low power, low data rate method of communication [5]. This low data rate limitation coupled with the fact that the communication method does not require a clear line of sight between the transceiver and

the tag led to the most common application being non-line of sight identification, colloquially known as radio frequency identification (RFID).

In RFID systems, the CW signal is only used to power the passive tag and does not indicate whether the reader or the tag initiates communication. Depending on the particular application, an RFID system can use a reader-talk-first approach where the reader queries the tags within its read range, which then prompts the tags to respond (transmit), or a tag-talk-first approach where the tag transmits its tag identity when it senses the CW signal indicating it has entered the read range of an RFID reader [4]. In the reader-talk-first design, the time between queries is known as an inventory period, which becomes the basis for organizing reader-tag communications making it a fundamental building block to an in-depth study of passive UHF RFID standards [4].

Global RFID standards define five different types of tags based on their power source and capabilities [5]. Categorizing the five types of tags using only the power source narrows the five categories down to three: passive, semi-passive, and active. Passive tags have no dedicated power source and operate in a manner described in the previous paragraph whereas semi-passive tags, also known as battery-assisted tags, have a small battery to power the tag's circuitry. Semi-passive tags still use the CW signal from the reader, known as the downlink, to generate the RF signal that communicates back to the reader on the uplink [4]. In contrast, active tags have a dedicated power source used to both power the tag's circuitry and transmit on the uplink path, which allows active tags to transmit over longer distances when compared to passive and semi-passive systems.

Propagation-based RFID systems are primarily used at VHF (primarily active RFID), UHF (passive and semi-passive), and microwave (passive and semi-passive) frequencies [5]. These frequencies typically fall within the unlicensed Industrial, Scientific, and Medical (ISM) bands, which can vary from country to country. Passive tags using VHF frequencies are typically employed for item-level tracking because the tags are inexpensive and have limited functionality. In comparison, passive and semi-passive tags using microwave frequencies are more expensive but have more functionality than UHF tags, more available bandwidth, and use higher data rates for communication [6]. The higher data rates and additional bandwidth in the microwave bands enables new areas of research such as factory automation, toll collection systems, barrier-based access control, localization and tracking, health monitoring, and contactless data transfer applications [5][6]; however, since these RFID systems are employed within the ISM bands, the readers must share the spectrum with other devices and must accept interference from other devices.

The two microwave ISM bands available for research are at 2.4 GHz and 5.8 GHz. A unique aspect of the 2.4 GHz band is that it is a worldwide license-exempt band, which is a major advantage for device manufacturers because a single hardware design can satisfy regulations worldwide but is a disadvantage at the user level because the band is often too congested [5]. In contrast, the 5.8 GHz ISM band, while not a worldwide license-exempt band, has much less congestion and the additional benefit of high gain read antennas and more available bandwidth which translates to higher data rates [6].

Even though RFID is a more ubiquitous term, this project will use the term backscatter communication in lieu of RFID because the term is more appropriate for the

enhanced backscatter capabilities research conducted using the R.E.S.T. system—mm-scale localization and tracking, antennas to resist shielding from high voltage power lines, retrodirective-phase modulation for low-powered wireless sensors, and QTRs to increase the range of low power communications—which is more than basic non-line-of-sight identification [7].

1.3 Research Overview

This thesis will cover the basics of the two most common spread spectrum communications methods in Chapter 2 followed by the history of the Georgia Tech Propagation Group Backscatter Communication System in Chapter 3, specifically focusing on the hardware and any use of spread spectrum communications. Chapter 4 will cover the development of the frequency hopping spread spectrum protocol for the R.E.S.T. 2.0 system including potential improvements, and Chapter 5 will cover the measurements and testing of R.E.S.T. 2.0 in its current configuration to show compliance with Part 15 requirements.

CHAPTER 2. SPREAD SPECTRUM COMMUNICATIONS

Spread spectrum communication is a technique that has its origin in World War II military radio networks where it served as a radio-guided torpedo interference and jamming countermeasure by spreading a narrowband signal over a wider bandwidth before transmission and using a receiver to de-spread the received transmission to obtain the original narrowband signal [8][9]. With spread spectrum communications, the system uses less power than narrowband signals and that power is spread over a wider bandwidth than what is required for the underlying signal, which improves interference rejection, reduces the effects of multipath and fading, increases transmission security, and provides multiple access capability [10]. Spread spectrum communications were initially thought to be inefficient compared to narrowband radios; however, researchers eventually discovered that spread spectrum systems were very bandwidth efficient in multiple-user, multiple access environments (MAI), which is a core characteristic of ISM bands [12].

Spread spectrum communications were primarily found in military radio networks until in the 1980s when the FCC changed the regulations in 47 C.F.R. §15, hereinafter referred to as Part 15, to allow the use of spread spectrum radios transmitting up to 1 Watt (W) within unlicensed ISM designated frequency bands [13]. This change quickly increased commercial interest in spread spectrum communications leading to the development and marketing of commercial products from cellular phones to wireless local area networks and the Global Positioning System (GPS) [13]. Many spread spectrum communication methods exist but the two predominant ones are Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS).

2.1 Direct Sequence Spread Spectrum

While an exhaustive explanation of DSSS is beyond the scope of this project, a cursory explanation is necessary in understanding the evolution of the Propagation Group backscatter communication system because one of the versions implemented DSSS in its design. In DSSS, the original information signal is multiplied with a pseudo-noise (PN) spreading code, also known as a chipping sequence, which spreads the original signal over a wider bandwidth, and the receiver then uses this same sequence to de-spread the signal and recover the original data [9]. The PN code is a randomly-ordered binary code with noise-like properties where there are generally equal numbers of 1s and 0s and is known by both transmitter and receiver [12]. Each of the binary numbers used in the PN code are known as chips and the binary numbers in the encoded (spread) signal are known as bits to distinguish them from the PN code and the original unencoded message [8]. The rate at which encoding occurs, known as the chipping rate, must be higher than the underlying data symbol rate, which causes DSSS to consume more power and require greater bandwidth than other spread spectrum techniques [8]. However, spreading the original data over a wider spectrum makes DSSS an ideal choice for operating in high interference environments such as ISM bands.

An essential parameter in understanding a DSSS system's effectiveness is the spreading ratio, more commonly known as the coding gain or processing gain (PG) [8] [11]. PG is a measure of the system's ability to suppress interference; the greater the PG, the more immune the system will be to interference [8][11][12]. The PG ratio has several equivalent mathematical definitions, but the simplest to understand is the time-based

version given in Equation 1, which is the ratio of the slower data symbol duration, T_s, to the faster chip duration, T_c [12].

$$PG = \frac{T_s}{T_c} \tag{1}$$

Most regulatory bodies require a minimum PG value, which is typically 10 dB, but generally do not restrict the maximum value [8]. The maximum value is often dictated by economics because a higher PG requires more sophisticated, and thus more expensive, RF components to generate the faster chipping rate required to produce a higher gain [8].

In conclusion, DSSS consumes more power than other spread spectrum techniques and, in general, is more difficult to implement because the system components are more complex than frequency hopping components, which are nearly identical to the narrowband radios used prior to the expanded use of spread spectrum communications [13].

2.2 Frequency Hopping Spread Spectrum

FHSS is exactly what its name implies; a system that hops, or changes, frequencies to spread out communication over a wider spectrum. In FHSS, the set of *M* frequencies used in communication is called the hopset, and the hopping band, *W*, is the range covered from the smallest frequency to the highest frequency, which covers all *M* frequency channels [10]. Each frequency channel has a spectral region with a single carrier frequency at its center that covers a bandwidth, *B*, [10]. The frequency hopping pattern is the pseudo-random order of frequencies used during hopping, and the rate at which frequencies change is called the hop rate [10]. The hop interval is the time between hops, and the hop duration,

denoted by T_h, is the time that the system occupies a frequency [10]. Guard intervals (GI), may or may not exist between frequencies and at the ends of the hopping band; whether GIs are present or not, the hopping band must satisfy the relationship $W \geq MB$ [10].

The hop duration, T_h, can be further divided into the dwell time, the time period when actual communication occurs, and the switching time which includes the dead time when no signal is present and the rise and fall times on either side of the dwell time [10]. Additionally, guard times, if used, would contribute to the overall switching time [10]. The four essential parameters of the hop duration form the frequency hopping pulse as shown in Figure 1.

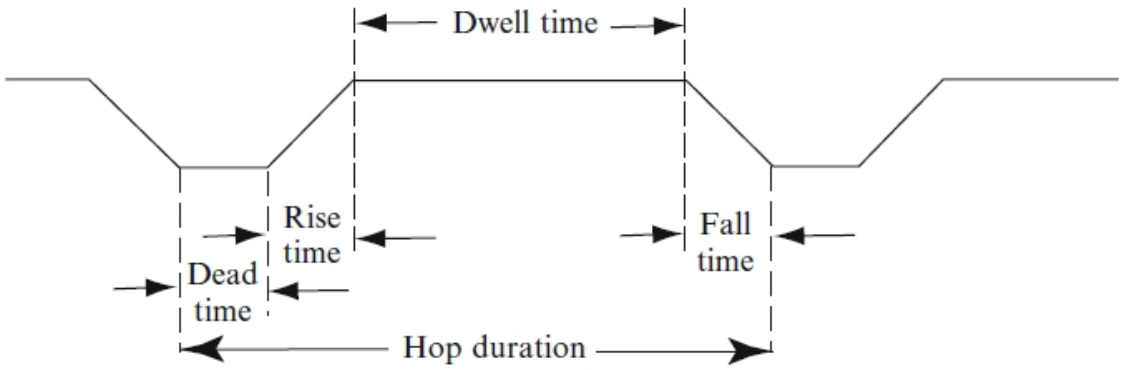

Figure 1 - Time durations of a frequency-hopping pulse [10]

Frequency hopping is described as slow frequency hopping or fast frequency hopping depending on how many information symbols are transmitted during a hop duration. If one or more information symbols are transmitted during a hop duration, the system is considered slow frequency hopping whereas fast frequency hopping occurs when

a single information symbol is transmitted over multiple hops [10]. Fast frequency hopping is only feasible if the hop rate is greater than the information symbol rate [10]. In most applications, slow frequency hopping is preferred because it minimizes the overhead cost incurred from the switching time and it simplifies the transmitted waveform [10].

2.3 Frequency Hopping Backscatter Communications

Frequency hopping in a passive and semi-passive backscatter communications system has some distinct advantages over traditional communications systems with the most obvious being the elimination of transceiver synchronization. In frequency hopping, the transceivers must synchronize their hops otherwise they will occupy different frequencies at different times, which will negatively affect communications, and synchronization is typically the most difficult part to implement in frequency hopping systems [10]. Backscatter communications systems have an advantage in that transceiver synchronization is automatically achieved since the tag transmits on whatever frequency it receives from the reader. When a reader must hop to a new frequency, it must first turn off the power source to the tag, which is then unable to transmit because passive tags are completely reliant upon the reader's CW signal for its power.

Another unique challenge in frequency hopping backscatter communication involves the coordination between timing a frequency hop and packet receipt. Specifically, the action a reader takes if it has reached its maximum dwell time but is still in the middle of an inventory period. The EPCglobal Class 1 Generation 2 standard is for passive tags operating in the 860 MHz – 960 MHz ISM band and recommends that readers maintain power to the tag when the tag is replying to a query; however, if maintaining power is not

an option due to the need to change frequencies, the standard provides a mechanism for a tag to maintain a cryptographic state while the reader changes frequencies and then re-acquire the tag during the next inventory round [14].

The higher data rates available in the 5.8 GHz ISM band offer some unique sensor applications using frequency hopping backscatter communication. One such application area is the use of tags employing an accelerometer to refine localization and tracking (L&T) measurements by combining the accelerometer data with more traditional L&T methods such as received signal strength and phase signal of arrival [15]. Incorporating a frequency hopping protocol introduces multiple frequencies, thus multiple phases, which can be used to refine L&T measurements even further [16].

CHAPTER 3. THE PROPAGATION GROUP BACKSCATTER COMMUNICATIONS SYSTEM

This chapter will document the history of the Propagation Group backscatter communication system from its initial development in 2009 through the current version undergoing development. Each section will primarily focus on the evolution of the transmitter and reader hardware designs including any spread spectrum methods used because the antenna and tag designs are typically application-specific. Through subsequent designs, the system has gotten smaller and more flexible in its use of core components to assist in rapid, low cost prototyping that minimizes the potential sources of error when testing new hardware applications or designs.

3.1 The Propagation Group Backscatter Testbed (2009)

Griffin developed the first Propagation Group backscatter communications system in 2009 using three custom, direct-conversion receivers for separate bistatic and monostatic configurations. The direct-conversion receivers consisted of two custom-designed boards—an RF front end board and a baseband amplification board—connected through external coax cables using standard SMA connectors [17]. The choice to use a direct-conversion receiver, also known as a homodyne receiver, was made because most of its baseband signal amplification occurs after removing any self-interference signals [17]. Using custom-designed receivers achieved several goals with the chief ones being self-interference signal mitigation to provide coherent reception and maximizing flexibility with the option to operate over the entire 5.8 GHz ISM band [17]. The Testbed did not

implement spread spectrum communications into its design. An Agilent E8247C signal generator along with two Mini Circuits splitters and an amplifier provided an unmodulated 5.79 GHz signal to serve as the transmitter and a local oscillator (LO) source for the direct-conversion receiver [17]. Figure 2 provides a block diagram of the Propagation Group Backscatter Testbed in its bistatic and monostatic configurations.

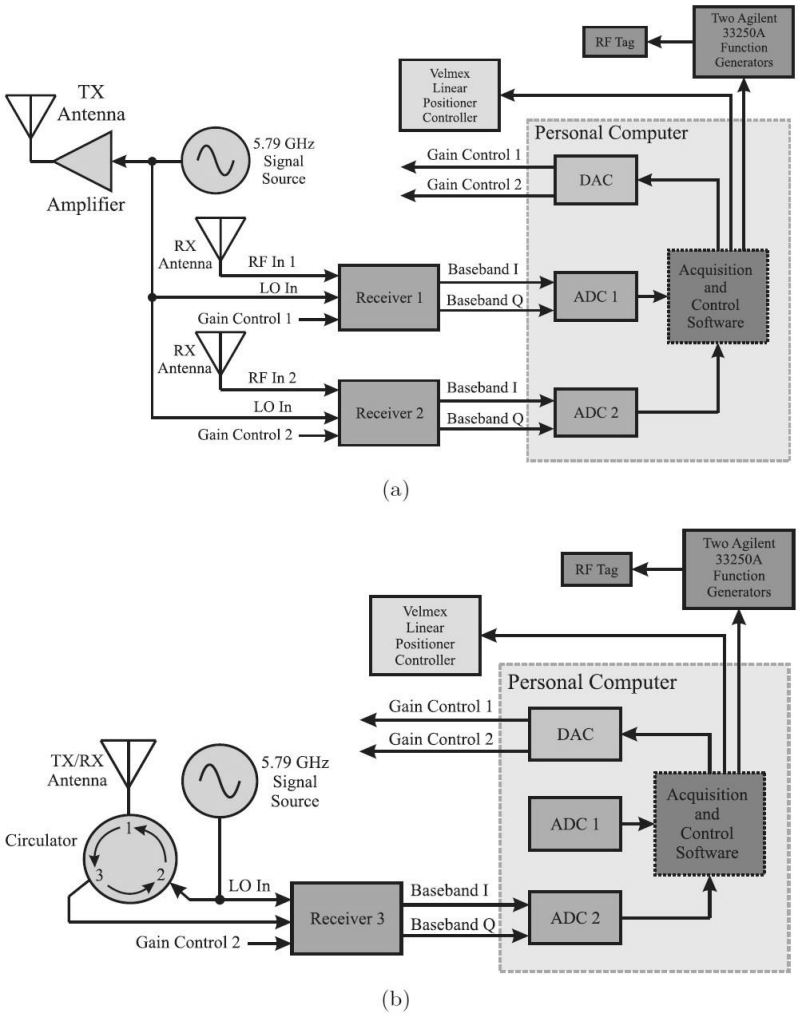

Figure 2 - Propagation Group bistatic (a) and monostatic (b) backscatter testbed block diagrams [17]

The system's signal processing occurred with a personal computer (PC) using two Exacq analog-digital- conversion (ADC) boards and one Exacq digital-analog-conversion (DAC) board [17]. The ADCs sampled the baseband in-phase (I) and quadrature (Q) signals from the direct-conversion receivers while the DAC controlled the gain of the receivers [17]. The signal generator providing the 5.79 GHz signal source also provided a 10 MHz reference signal that was conditioned with several Mini Circuits components and a DC offset to provide an amplified 80 MHz baseband signal to the ADC and DAC boards [17]. Both ADCs and the DAC were controlled using a C++ program written in Matlab [17].

Griffin developed a bistatic and monostatic configuration tor testing the performance of both designs in the Backscatter Testbed. The bistatic setup used one transmit antenna and two receivers, each with their own antenna, while the monostatic setup used a single receiver and antenna with a microwave circulator to transmit and receive [17]. Since this was the Propagation Group's first complete backscatter communications system, Griffin had to design and build all primary components from the receiver to the tags and the antennas in the interest of exploring multipath fading in backscatter channels. The Backscatter Testbed did not incorporate spread spectrum techniques into the design due to the focus on multipath fading measurements; however, these multipath fading measurements were essential in the development of future versions of the backscatter communications system.

3.2 GTX 1.0 (2010)

The second version of the Propagation Group backscatter communication system started through a partnership with a company that produces products marketed to electric utility companies. The system was optimized for a high voltage environment using a proprietary current measurement, relay protection system to quickly identify faults in transmission lines thereby reducing the impact of power outages [18]. The goal of the Georgia Tech eXperimental air interface version 1.0 (GTX 1.0) was to use a 5.8 GHz backscatter radio relay system to perfectly reproduce an analog signal from an RF sensor tag to quickly identify faults on transmission lines [18]. The operating environment required a system designed for high-speed, high-reliability, low-latency, and interference-tolerant data transfer in high voltage environments [18]. The entire system, except for the RF sensor tag, was contained in a single enclosure installed below a set of high-voltage power lines. The system, excluding tags and antennas, consisted of five custom-designed stages: 5.8 GHz signal generation, RF power amplification, RF down-conversion, baseband amplification, and demodulation. Figure 3 provides a block diagram for the entire GTX 1.0 system. The dotted black lines in the figure represent individual, custom-designed circuit boards, and the numbered black circles represent the interfaces and associated inputs/outputs between the various components [18].

The transmitter consists of the RF signal generation and RF power amplification stages, each implemented on their own custom-designed circuit board. The signal generation board's primary components are an Analog Devices ADF4107 Phase Locked Loop (PLL) frequency synthesizer board and a Texas Instruments (TI) MSP430 microcontroller to control the PLL's CW sinusoidal frequency-hopped output [18]. The

signal generator output connected to the RF down-conversion board for down-converting the received signal and to the RF power amplifier board for transmitting to the sensor tag [18]. This signal generator board replaced the benchtop Agilent signal generator used in the first version, which was necessary because the entire system was intended for deployment as a self-contained, stand-alone device.

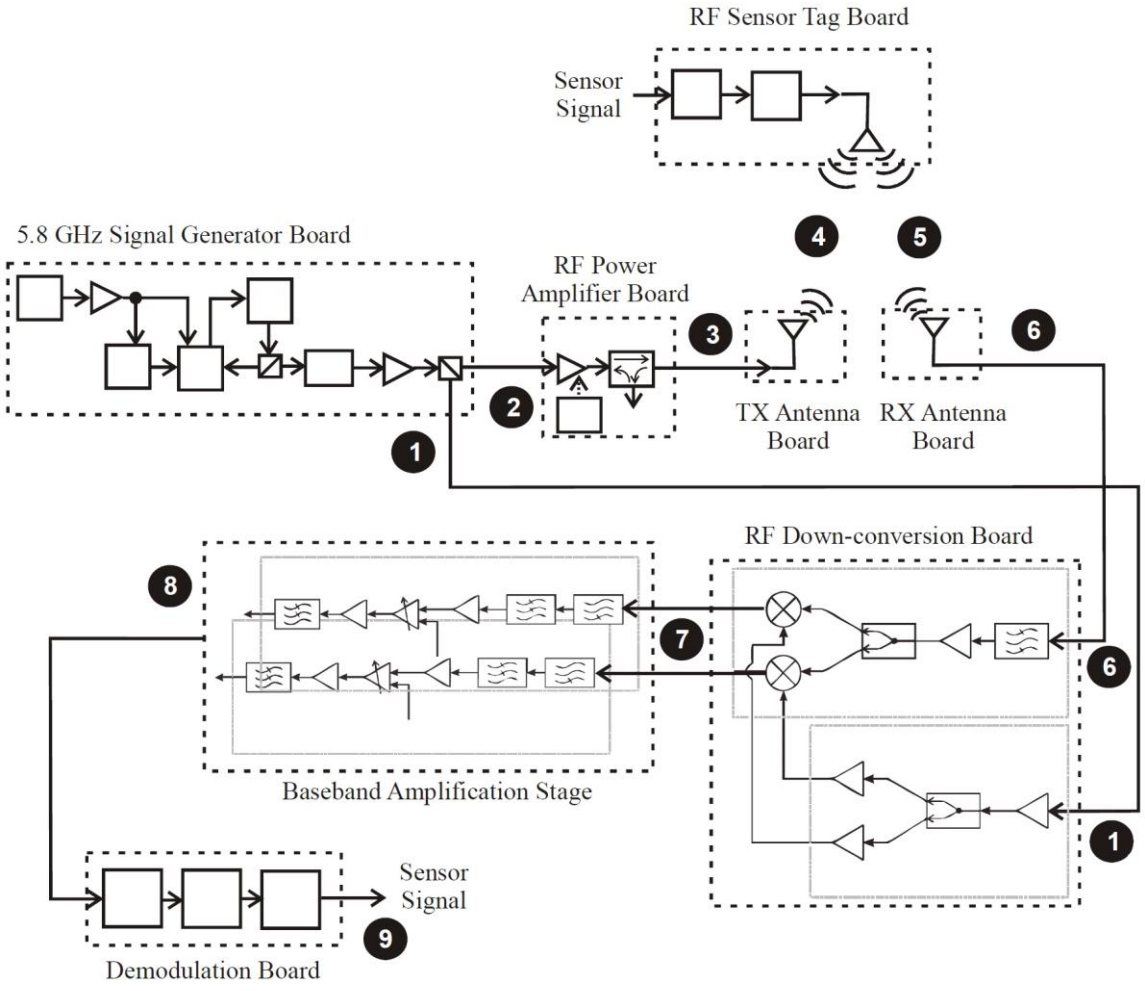

Figure 3 - GTX 1.0 block diagram; dashed lines represent individual circuit boards and the numbered black circles represent each board's input/output [18]

GTX 1.0 was unique in that the system implemented FHSS in the CW signal to the sensor tag and DSSS in the modulated signal from the tag to the reader. On the signal generator board, the MSP430's microcontroller code controlled the ADF4107 PLL to hop its output frequency between 5.763 GHz and 5.837 GHz with 1 MHz channel spacings and a 400ms dwell time [18]. The code generated a pseudorandom list of seventy-five channels by starting with a pseudorandom binary sequence that was converted to a decimal number and then shifting the sequence by one and converting to decimal again to create additional channel numbers [18]. The list was then scaled from one to seventy-five to generate the hopping sequence. In contrast, the RF sensor tag on the transmission line implemented DSSS to mitigate interference to its modulated signal back to the reader [18]. The tag used a 63-bit long Kasami sequence and chip period of 40 μs to producing a processing gain of 9.29 dB [18]. To increase the decoding reliability at the receiver, each chip was sampled 3 times producing a sampling rate of 7.5 Msamples/sec [18].

The GTX 1.0 receiver consists of the RF down-conversion stage, baseband amplification stage, and the demodulation stage. The RF down-conversion board combined several standard Mini-Circuits components onto a single board to down-convert the signal from the receive antenna. An external DC power supply provided power to the circuit components in the down-conversion stage and the first half of the baseband amplification stage; the first half of the baseband amplification stage and the down conversion stage were co-located on the same circuit board due to the shared external power supply [18]. The demodulation board implemented a demodulation and decoding algorithm on an Altera DE2-70—a Field Programmable Gate Array (FPGA) development board—connected to a proprietary ADC/DAC converter designed by the partner company

to generate a reconstructed analog signal from the sensor on the transmission line [18]. The use of an FPGA for demodulation and decoding was necessary because of the proprietary ADC/DAC board.

GTX 1.0 made three significant improvements to the Backscatter Testbed: it was a completely self-contained, standalone device; it implemented both FHSS and DSSS; and it was optimized for use in a high-voltage environment. Additionally, the system did not use any of the of major components from The Backscatter Testbed and became the foundation for a commercially-available device currently available to reduce the impact power outages have on daily life.

3.3 RFID-Enabled Sensing Testbed (R.E.S.T.) 1.0 (2012)

In 2012, Valenta developed the RFID Enabled Sensing Testbed (R.E.S.T.) to support low-cost, rapid prototyping of wireless sensors across a wide range of frequencies [19]. The principle motivation behind R.E.S.T. was to reduce the monetary cost and time associated with multiple iterations of sensor prototyping by designing a universal digital portion of the sensor tag that can then interface with the sensor and RF components [19]. This universal digital portion has the added benefit of isolating the components under development thereby reducing the sources of error encountered during prototyping and decreasing development time even further [19]. The R.E.S.T. 1.0 system consists of a motherboard that functions as the sensor's transceiver and interfaces with interchangeable daughterboards for sensing and communications [19]. R.E.S.T. 1.0 uses Griffin's custom-built direct down-conversion receiver from the Backscatter Testbed but improved the receiver by replacing the 5.79 GHz source from the Agilent signal generator with the signal

generator board from GTX 1.0 [19]. Figure 4 provides a block diagram for R.E.S.T. 1.0 along with four of the different daughterboards developed.

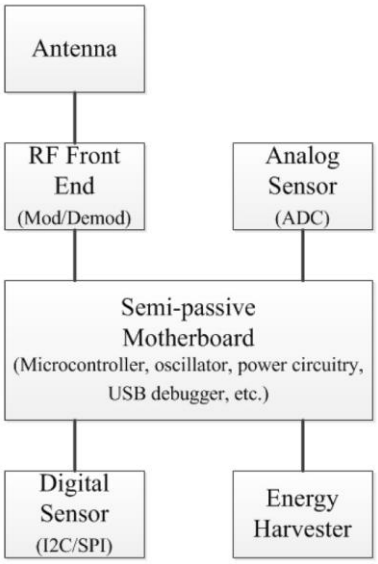

Figure 4 - R.E.S.T. 1.0 block diagram with interchangeable daughterboards [19]

The R.E.S.T. 1.0 motherboard consists of a TI MSP430F5419a microcontroller to provide a 25 MHz clock, Serial Peripheral Interface (SPI) and Inter Integrated Circuit (I2C) communications capability, as well as pulse-width modulation (PWM) and ADC functionality [19]. The motherboard's programming uses the C language and implements cyclic redundancy check (CRC) for error correction while connecting to various interchangeable daughterboards that provide communication and sensor capabilities [19]. Power is supplied through either a TI Field Effect Transistor (FET) debugger, local battery pack mounted to the system, or other external power source [19]. Two examples of the

daughterboards highlighted include an RF Front End using a Mini Circuits single-pole double-throw (SPDT) switch to provide Binary Phase Shift Keying (BPSK) or Differential BPSK (DBPSK) modulation and a motion-capture sensor equipped with a three-axis accelerometer, three-axis gyrometer, and I2C to communicate with the microcontroller [19].

R.E.S.T. 1.0 uses the Backscatter Testbed down-conversion receiver but replaced the PC with Exacq ADC and DAC cards with Ettus Research USRP N-200 software defined radios (SDR) using an LFRX daughtercard to perform the I and Q channel sampling as well as baseband processing [19]. Additionally, the signal generation board from GTX 1.0 was incorporated into the down-conversion receiver to replace the 5.79 GHz signal source from the Agilent E8247C signal generator. In R.E.S.T. 1.0, the signal generator provides the CW excitation to the transmit antenna and receiver and can be programmed for frequency hopping [19]. The N-200 SDRs are programmed in Python with the GNUradio open source software development toolkit using the 'gr-digital' toolbox [19]. R.E.S.T. 1.0 made significant contributions to backscatter communication research at Georgia Tech while providing a valuable hands-on platform for teaching students about microwave matching, S-parameters, and transmission lines [19].

3.4 R.E.S.T. 2.0 (2018)

Version 4 of The Propagation Group's backscatter communications system, R.E.S.T. 2.0, seeks to continue the modular configuration concept from R.E.S.T. 1.0's transceiver design but decrease its size and power requirements. The fundamental components of R.E.S.T. 2.0 in its current configuration utilize the TI LAUNCHXL-F28027

microcontroller, herein referred to as F28027, for all programming and digital signal processing operations and the TI LMX2592EVM PLL RF synthesizer board, herein referred to as LMX2592, to transmit in the unlicensed 5.8 GHz ISM band. R.E.S.T. 1.0's microcontroller platform is still used for enhanced backscatter capabilities sensor research; however, a need existed for future research that implemented the same flexible hardware and software solution the N-200 SDRs provided but in a low power, space constrained design.

The system uses the internal 1 MHz reference frequency from the LMX2592 and continues to use BPSK and DBPSK modulation. The LMX2592 has an output range of 20 to 9800 MHz and supports both integer-N and fractional-N modes to select the output frequency [20]. The overall design is still under development, so some of the components may be replaced with devices that have more functionality but will still use a direct conversion receiver design.

Both the F28027 and LMX2592 operate on a 3.3V input with variable current requirements depending on the configuration. For the LMX2592, the output signal power is the primary parameter that impacts its current requirements; however, the F28027 has several parameters that can affect its power consumption. The F28027 is currently configured to use 60 MHz internal clocks for cputimer0 and cputimer1; however, 40 MHz and 50 MHz clocks are available, which if used, can reduce power consumption [21]. In addition to using a slower internal clock, power consumption can be optimized in programming by turning off any unused clocks and disabling unused output pins [21]. Finally, additional current consumption reductions are available in the F28027 by incorporating three different low-power modes during operation, which involve having the

processor enter either an idle, standby, or halt configuration to further reduce power consumption [21]. R.E.S.T. 2.0 is still under development, so its final power consumption performance is currently unavailable.

During initial design and testing, a USB cable connected the F28027 to a personal computer running TI's Code Composer Studio (CCS) version 7.1.0 software. The USB cable provided both data communication and power to the F28027 while the LMX2592 was powered using an external regulated power supply. The F28027 transmits data to the LMX2592 using the SPI standard. The specific pin connections used are given in Table 1:

Table 1 - F28027 to LMX2592 SPI connections [20][22][23][24][25]

SPI Connection	F28027	LMX2592 (uWire)
SCLK	J1 Pin 7	Pin 8
MOSI	J2 Pin 6	Pin 4
MISO	Not used for frequency hopping	
SS	J2 Pin 2	Pin 2
GND	GND	Pin 9

Part 15.203 requires that marketed devices have a permanently attached antenna or that it employ a unique antenna connector [26]. R.E.S.T. 2.0 currently uses standard SMA connectors between the LMX2592 and the patch antennas which does not meet the requirements of 15.203. If Part 15 compliance certification is pursued in the future, the system must either have permanently attached antennas, utilize antenna connectors not readily available to the public, or implement any of the approved methods described by the FCC's Office of Engineering and Technology (OET) [27].

CHAPTER 4. PART 15 REQUIREMENTS

Part 15 is divided into numerical sections with groupings of sections categorized into Subparts; R.E.S.T. 2.0's requirements are within Subparts A and C. Subpart A covers sections 1 through 38 and contains general guidance applicable to all radio frequency devices. Subpart C governs the operation of intentional radiators with the general requirements outlined in sections 201 through 214 and specific frequency hopping requirements outlined in section 247 for point to multi-point devices operating within ISM bands. Most of the requirements apply to devices seeking FCC certification for intentional marketing and sale as a commercial device. This chapter will focus on the frequency hopping design requirements outlined in §15.247. The general requirements in sections 1 to 38 of Subpart A and sections 201 to 214 of Subpart C will be addressed in the next Chapter since they only pertain to the testing of an FHSS device and do not dictate requirements needed in the development of the frequency hopping protocol.

4.1 Part 15 Frequency Hopping Requirements

Section 247 covers all point to multi-point devices operating within the 902 MHz, 2.4 GHz, and 5.8 GHz ISM bands. The 5.8 GHz band starts at 5725 MHz and ends at 5850 MHz providing a total of 125 MHz for the hopping band. Frequency hopping systems must use a hopset of at least 75 frequencies with a minimum carrier frequency separation of either 25 kHz or the 20-dB bandwidth of the hopping channel, whichever is greater [26]. The frequency hopping system must have a hop duration of no more than 0.4 seconds in a thirty second period, which corresponds to a hop rate of at least 2.5 hops per second [26].

The frequency hopping pattern must be a pseudo randomly ordered list in which the transmitter uses each frequency equally on average [26].

4.2 Frequency Hopping Pattern Design

This R.E.S.T. 2.0 hopping pattern is designed to be as close to single frequency communication as possible; therefore, the dwell time and channel bandwidth are maximized while the hopset size is minimized. The hopping pattern and hop rate can always be re-designed to support specific advanced backscatter communications research such as localization and tracking.

4.2.1 Hopset

Designing a specific hopset that satisfies the requirements in Part 15 started with dividing the 125 MHz hopping bandwidth by 75, the minimum number of authorized channels, producing a maximum 1.667 MHz bandwidth per channel. To provide some limitation on the channel resolution, the number of total channels increased to 78 resulting in a 1.6 MHz channel bandwidth. At 1.6 MHz per channel and 78 channels, there is only 0.2 MHz remaining for a 0.1 MHz buffer at the top and bottom of the band.

The configuration of the LMX2592 limited the output frequencies' resolution to 0.2 MHz increments of even decimal fractions. The 0.1 MHz buffer at the bottom of the band created an unforeseen problem by causing the output frequencies to have odd decimal fractions. The solution to this was to either (1) shift the entire 0.2 MHz buffer to the bottom or top of the band or (2) sacrifice one of the channels to increase both buffers creating a total of 77 channels with even decimal fractions. While the first option is the ideal one to

maximize hopping band usage, the risk of transmitting outside the authorized band because the last channel ends right at 5850 MHz makes it impractical; therefore, the second option was chosen resulting in 77 channels of 1.6 MHz instantaneous bandwidth and a 0.8 MHz and 1.0 MHz buffer to the lower and upper part, respectively, of the hopping band. The 77 channels in the hopping band do not have guard intervals between them.

4.2.2 Hop Sequence

To develop the hop sequence, the 77 channels developed in the hopset step were first listed in a Microsoft Excel spreadsheet then the random number generator function =RAND() used to associate each carrier frequency with a random number from zero to one. These random values were locked to prevent them from changing and then the list of frequencies sorted to provide a preliminary random hop sequence. The frequencies were then divided into seven sub-hopsets with eleven frequencies per sub-hopset. Adjacent frequencies were compared to each other, and if hopping occurred within the same sub-hopset, one of the frequencies' position in the sequence was switched with a nearby one until the entire sequence had adjacent frequencies from different sub-hopsets.

This first randomization procedure produced a suitably random sequence; however, it cannot account for situations when hopping occurs from a frequency toward the top of a sub-hopset to a frequency from the lower part of the adjacent sub-hopset. At 1.6 MHz bandwidth per channel and eleven frequencies per sub-hopset, each sub-hopset has a total bandwidth of 17.6 MHz. Comparing adjacent frequencies to ensure they are separated by at least 17.6 MHz will ensure that adjacent frequencies are separated by at least one sub-hopset bandwidth. Adjacent frequencies were compared and if hopping occurred within

a sub-hopset bandwidth, one of the frequencies' position in the sequence was switched with a nearby one until the entire sequence had adjacent frequencies separated by at least one sub-hopset bandwidth. This second randomization procedure guarantees that hopping is sufficiently spread throughout the hopping band. The final hop sequence is shown in Table 2.

Table 2 – R.E.S.T. 2.0 frequency hopping sequence

#	f (MHz)	#	f (MHz)	#	f (MHz)	#	f (MHz)	#	f (MHz)
1	5824.2	17	5827.4	33	5806.6	49	5742.6	65	5816.2
2	5745.8	18	5731.4	34	5736.2	50	5835.4	66	5777.8
3	5830.6	19	5766.6	35	5789.0	51	5758.6	67	5811.4
4	5779.4	20	5800.2	36	5749.0	52	5781.0	68	5739.4
5	5805.0	21	5734.6	37	5729.8	53	5817.8	69	5761.8
6	5750.6	22	5769.8	38	5841.8	54	5755.4	70	5829.0
7	5813.0	23	5819.4	39	5801.8	55	5833.8	71	5768.2
8	5787.4	24	5790.6	40	5832.2	56	5814.6	72	5798.6
9	5733.0	25	5771.4	41	5809.8	57	5765.0	73	5840.2
10	5773.0	26	5843.4	42	5744.2	58	5846.6	74	5741.0
11	5848.2	27	5728.2	43	5838.6	59	5757.0	75	5776.2
12	5803.4	28	5793.8	44	5792.2	60	5784.2	76	5752.2
13	5760.2	29	5822.6	45	5845.0	61	5825.8	77	5774.6
14	5785.8	30	5737.8	46	5763.4	62	5797.0		
15	5821.0	31	5837.0	47	5808.2	63	5753.8		
16	5726.6	32	5747.4	48	5782.6	64	5795.4		

Table 3 summarizes the basic statistical properties between adjacent channels in the hop sequence. From the table, one can see that the median change between adjacent frequencies is 46.4 MHz with the smallest change being 19.2 MHz and the most frequent being 25.6 MHz. These values indicate that the transmitter will not occupy a localized section of the hopping band for adjacent hopping periods which satisfies the description of a pseudorandom frequency hopping sequence [28].

Table 3 – R.E.S.T. 2.0 statistical analysis of adjacent frequency hopping channels

Smallest Hop	19.2 MHz
Greatest Hop	115.2 MHz
Average Hop	53.5 MHz
Median Hop	46.4 MHz
Most Frequent (Mode)	25.6 MHz

4.2.3 Hop Rate

Section 247 specifies that the system must not occupy a single frequency in the hopset for longer than 0.4 seconds in a thirty-second period [26]. A 0.4 second hop duration corresponds to a minimum hop rate of 2.5 hops per second. Dividing the hopset of 77 frequencies into thirty seconds generates a minimum dwell time of 0.38961 seconds; therefore, after rounding up, 0.39 seconds is the smallest possible hop duration without breaking the requirement of occupying a frequency for no more than 0.4 seconds in a thirty second period assuming the frequency is occupied only once during a thirty second period. In other words, if the hop duration were less than 0.39 seconds (assuming frequency occupation only once in a thirty-second period) then the system would occupy the initial frequencies in the hopset twice within a thirty second period, which would exceed the maximum allowed 0.4 seconds. If faster frequency changes are needed, then the dwell time chosen should be set to integer multiples of 0.39 seconds. For example, a hop duration of 0.195 seconds would allow to the system to occupy frequencies twice in a thirty-second period, and a 0.13 second hop duration would occupy hopset frequencies three times in a thirty-second period but still not exceed the 0.4 second total occupation limit.

R.E.S.T. 2.0 uses slow frequency hopping; therefore, the system must account for situations in which the reader is still receiving a packet but has reached the end of its hop duration. To account for this, the frequency hopping protocol will add a 1ms hop duration extension, with a maximum of ten 1ms extensions, to the standard 0.39 second hop duration until it has completed packet receipt. The system will hop to the next frequency after completing packet receipt and the current 1ms hop duration extension ends. In the unlikely scenario where the reader is still receiving a packet at the end of the tenth hop duration extension, the reader will stop receiving the packet and hop to the next frequency in the hopset to avoiding frequency occupation longer than 0.4 seconds in a thirty-second period.

Figure 5 provides a graphical representation of the frequency hopping pattern over time based on the standard 0.39-second hop duration. This graphical representation shows that the carrier distribution over a single hopset cycle appears random while appearing evenly distributed over multiple hopset cycles [26]

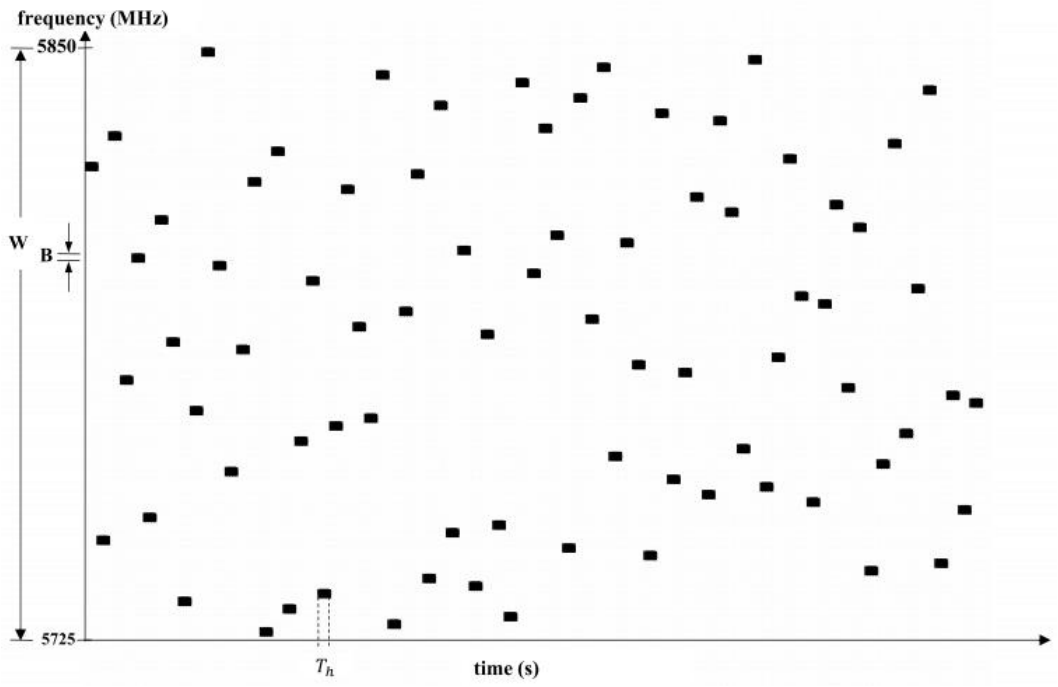

Figure 5 – R.E.S.T. 2.0 frequency hopping pattern

4.2.4 Frequency Hopping Implementation

When this project started, R.E.S.T. 2.0 already had code written in TI's CCS version 7.1.0 for the F28027 to program the LMX2592 to transmit at a single frequency. The LMX2592 has forty-three 24-bit registers for programming its various functions with two registers—38 and 45—controlling the output frequency in the 5.8 GHz ISM band [20]. The specific bit fields of interest within these registers are bits 12 to 1 in register 38 for the integer part of the N-divider and bits 15 to 0 in register 45 for the least significant bit (LSB) of the N-divider fraction numerator [20]. Bits 15 to 0 in register 44 contain the most significant bit (MSB) of the N-divider fraction numerator; however, this was not needed for the frequencies in the 5.8 GHz ISM band [20].

With the hopset and frequency hopping pattern established, the next step was to determine the register values for each of the frequencies in the hopset. The LMX2592 user guide provides equations to calculate the appropriate N divider values which then require conversion from decimal to hexadecimal; however, the method used in this project was to use the TI Clocks and Synthesizers (TICS) Pro Software on a personal computer directly connected to the LMX2592 using a TI USB2ANY Interface Adapter. The TICS Pro software allows the user to simply choose a desired output frequency then let the software calculate the necessary N-divider fraction parameters and write the appropriate registers to the LMX2592 to change frequencies. The TICS Pro software was used to determine the exact values in hexadecimal of registers 38 and 45 for all the frequencies in the hopset. These values were then written into an array in the baseline code to start the frequency hopping routine.

The LMX2592 has a five-step programming sequence when the device is initially powered on and a three-step recommended sequence for changing frequencies after the device is powered on and transmitting [20]. The basic single-frequency code from the start of the project already established the five-step power up sequence using the F28027's internal cpu-timer0 as an Interrupt Service Routine (ISR) to initiate the sequence. Within the register values of the power up sequence, the appropriate bit fields for the output frequency were changed to transmit the first frequency in the hopset. All other register values were written into an array with the last value in the array being the first frequency in the hopset; this method was required because the hopping sequence is different from the power up sequence. A new ISR was then written using cpu-timer1 set to 390ms and a frequency count variable used to increment within the hopset and select the appropriate

register values from the array for the next frequency. The three-step frequency change sequence is to program the new N divider value in register 38; then program the new PLL numerator and denominator in registers 45 and 44, respectively; and finally set bit 3 in register 0 to 1 to enable frequency calibration [20]. Using an incremental frequency count to select the register values from an array ensures that all frequencies are used equally [26].

The LMX2592 has a typical calibration time around 590μs with a fast calibration option available to reduce this time down to less than 25μs or less [20]. The fast calibration option requires programming additional registers to optimize the Voltage Controlled Oscillator (VCO) by getting it closer to the expected final output frequency value [20]. The fast calibration optimization procedure was not used in this project but may be incorporated in the future to assist with advanced backscatter communications research. Appendix A contains the full F28027 code for the power up and frequency hopping sequences.

4.3 Part 15 General Requirements

In addition to the specific requirements of Part 15, there are several general requirements associated with filing for Part 15 compliance certification. 47 C.F.R. §2.1 defines a frequency hopping system as a spread spectrum system that uses conventional modulation methods on a carrier frequency that "changes at fixed intervals under the direction of a coded sequence" [28][29]. R.E.S.T. 2.0 uses On Off Keying (OOK), a conventional modulation method, on carrier frequencies that change based on a coded sequence. Additionally, the definition states a frequency hopping system must have a "near term distribution of hops [that] appears random, long term distribution [that] appears

evenly distributed over the hopset, and sequential hops [that] are randomly distributed in both direction and magnitude of change" [28]. The two-step randomization method used ensures that the changes in frequencies appear random over a single hopset while appearing evenly distributed over multiple hopset cycles and that the changes are random in both magnitude and direction.

A spread spectrum system operating under §15.247 is excluded from routine RF exposure limits testing but must still show it meets exposure compliance when applying for compliance certification unless it operates at "substantially low output power levels, with a low gain antenna(s)" [29]. R.E.S.T. 2.0 is designed specifically for low power applications and uses patch antennas averaging 6 dBi gain, which do not exceed FCC RF exposure guidelines; therefore, documentation will not be required to verify RF exposure compliance if pursuing certification in the future [29][30].

Finally, the FCC recognizes that backscatter communication is a unique form of wireless communication; therefore, the FCC's Part 15 passive and semi-passive RFID compliance testing policy states that only the reader must undergo testing if the tags transmit at the same frequency as the reader [31][32]. Additionally, §15.247(a)(1) requires that a Part 15 frequency hopping receiver has an input bandwidth equal to the hopping channel bandwidth and synchronizes its frequency shift with the transmitted signal [26][29]. R.E.S.T. 2.0 associated battery-assisted and passive tags use the same frequency as the reader thus automatically synchronizing its hopping with the transmitter; therefore, R.E.S.T. 2.0 passive and semi-passive tags do not require Part 15 compliance testing.

4.4 Future Work

While the hopping protocol in this project is complete, it still has several improvements that can be made. Two immediate improvements would be to incorporate the 1ms hop duration extension and to incorporate carrier sense multiple access collision avoidance (CSMA-CA), also known as listen before talk, to increase the system's collision avoidance. Another improvement would be to explore the LMX2592's fast calibration option to reduce the switching time, which would be beneficial if the hopping protocol was optimized to minimize the dwell time rather than maximize it as the current protocol does. Implementing the fast calibration option can reduce the calibration time to less than 25μs and only requires adjusting some of the register values in the initial programming sequence. The fast calibration option works by providing an initial start value for calibration that is closer to the desired output frequency and by reducing the range of frequencies over which the calibration searches [20].

15.247(h) prohibits any coordination between FHSS devices other than for the express purpose of avoiding the simultaneous occupation of individual frequencies; therefore, the hopset randomization procedure initially used in this project needs to be automated within the F28027 code to ensure that each reader will have its own unique hopping sequence using the 77 frequencies in the hopset [26]. Having a unique hopping sequence for each transmitter will not only provide interference mitigation among readers but also provide a layer of security since each hopping sequence will be unique. A final future improvement, especially if Part 15 compliance certification is pursued, would be to analyze the spurious emissions to determine their source and then reduce them based on the recommendations in the LMX2592 documentation [20]

CHAPTER 5. PART 15 COMPLIANCE TESTING

The intent of this Chapter is to show that R.E.S.T. 2.0 in its current hardware design meets the requirements outlined in Part 15. Part 15, Sections 1 through 38, which together encompass Subpart A, covers all general requirements for Part 15 devices. §15.31, measurement standards, paragraphs (m) and (o) and §15.33, frequency range of radiated measurements, paragraph (a) are the most relevant in Subpart A to the testing of R.E.S.T 2.0's Part 15 compliance.

§15.31(o) indicates that spurious emissions outside the ISM band that are attenuated to 20 dB or more need not be reported unless the emission falls within a restricted band, and §15.33(a)(1) states that intentional radiators operating below 10 GHz must be tested to the 10^{th} harmonic of the highest fundamental frequency or 40 GHz, whichever is lower [26]. For the purposes of this testing, spurious emissions refer to the harmonics of the fundamental emission, and the two terms are used interchangeably in this chapter. §15.205 provides the restricted bands of operation for all intentional radiators [26].

§15.31(m) specifics that devices operating over a frequency range greater than 10 MHz need to be tested using at least three frequencies with one near the top, one near the middle, and one near the bottom of the band [26]. The frequency hop testing of R.E.S.T. 2.0 only utilized a single frequency, or when appropriate, two adjacent frequencies in the hopset; therefore, duplicate measurements at different frequencies using the system's final hardware design will be required if R.E.S.T. 2.0 Part 15 certification is pursued in the future.

The measurements and testing within this chapter follow the FCC's filing and measurement guidelines for FHSS devices. The guidelines provide eight specific measurements along with several general requirements requesting a description of how the device under test (DUT) complies with the requirements in Part 15. This chapter will first describe the testing setup followed by the eight specific measurements required. The general requirement descriptions were covered in the previous chapter. When possible, this chapter will reference specific sections and paragraphs to aid in locating each requirement.

5.1 Testing Setup

All testing on the current R.E.S.T. 2.0 configuration took place in room E560 of the Van Leer Building on the Georgia Institute of Technology campus. The F28027 and LMX2592 were connected to a personal computer as described in Chapter 2. An Agilent E3631A power supply provided a maximum of 3.3V and 500mA to the LMX2592. The LMX2592 was connected to an Agilent E4407B spectrum analyzer using a standard SMA to SMA coax cable connected to the RFoutAP port. Before connecting the LMX2592 to the spectrum analyzer, an internal manual alignment was performed. Testing the LMX2592's frequency transmission for accuracy indicated the frequency jitter reduced from approximately 6 kHz to less than 1 kHz over an approximately ten-minute period; therefore, both the LMX2592 and spectrum analyzer operated for at least ten minutes prior to capturing measurements to improve the accuracy of the testing. The E4407B has limited screen capture export options; therefore, all measurement screen captures were performed using a digital camera. The FCC measurement guidelines provide specific parameters for

the spectrum analyzer settings in the eight specific tests; the specific settings used in the testing of R.E.S.T. 2.0 are provided in each testing description below.

5.2 Carrier Frequency Separation

§15.247(a)(1) states that hopset frequencies must be separated by at least 25 kHz or the 20 dB bandwidth of the hopping channel, whichever is greater [26]. Section 5.5 details the 20 dB bandwidth measurement for R.E.S.T. 2.0, which is 195 kHz in its current configuration. The two frequencies tested were 5776.2 MHz and 5774.6 MHz. This test used a center frequency of 5.7754 GHz, span of 2.2 MHz, resolution bandwidth (RBW) of 30 kHz, video bandwidth (VBW) of 30 kHz, sweep set to auto, detector function set to peak, and the trace set to max hold to obtain the results shown in Figure 6. After the trace stabilized, the marker delta function was used to determine the separation between the adjacent peaks. From the figure, the separation between carrier frequencies is 1.6 MHz.

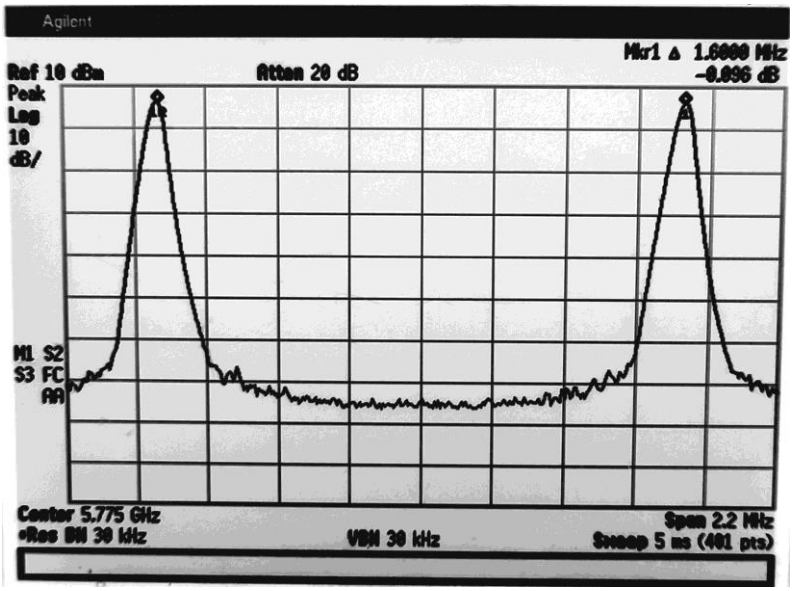

Figure 6 - The 5774.6 MHz and 5776.2 MHz signals are separated by 1.6 MHz

5.3 Number of Hopping Frequencies

Testing for the number of hopping frequencies used required two screen captures—one for the lower half of the hopping band and another for the upper half. The lower half used a 5756.25 MHz center frequency while the upper half used 5818.75 MHz with both using a span of 62.5 MHz. The FCC's FHSS testing guidelines state that the RBW must be greater than or equal to 1% of the span; therefore, a 1 MHz RBW was used because of the spectrum analyzer's limited RBW options. The sweep was set to auto, detector function set to peak, and the trace set to max hold. Figure 7 provides the results for the lower half of the band and Figure 8 provides the results for the upper half with the 39th frequency peak split between the two halves. The frequency peaks are numbered for clarification; the significant overlap in the peaks is due to the 1 MHz RBW restriction, which had to be greater than or equal to 1% of the span.

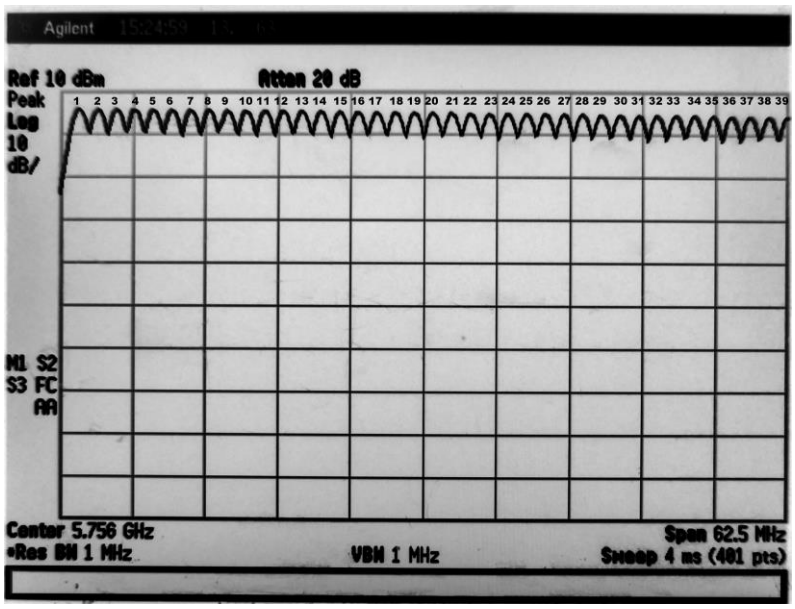

Figure 7 – The hopping frequencies in the lower half of the hopset

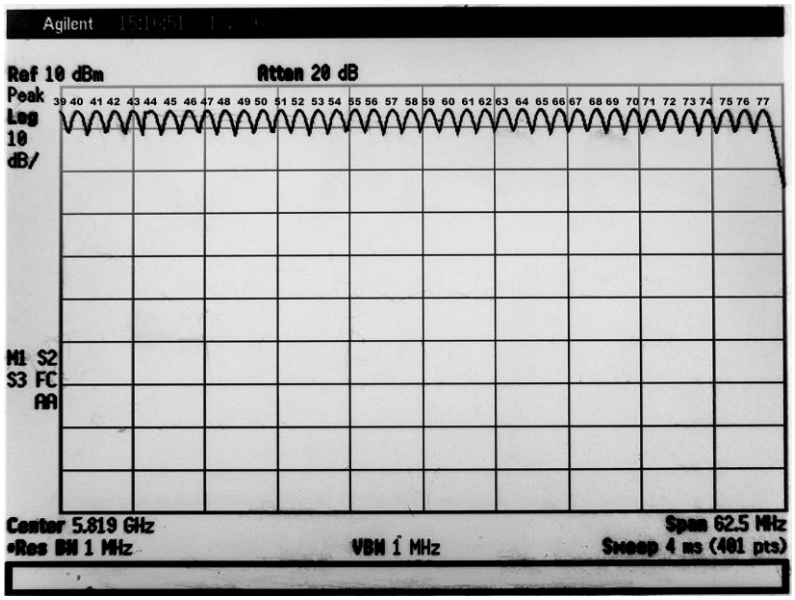

Figure 8 – The hopping frequencies in the upper half of the hopset

5.4 Time of Occupancy (Dwell Time)

Testing the hop duration, or dwell time, required a span set to zero and used a center frequency of 5848.2 MHz, RBW and VBW of 1 MHz, sweep set to 780ms to cover at least two hop durations, detector function set to peak, and trace set to max hold. After the trace stabilized the marker delta function was used to measure the dwell time. Figure 9 provides the results of the hop duration test.

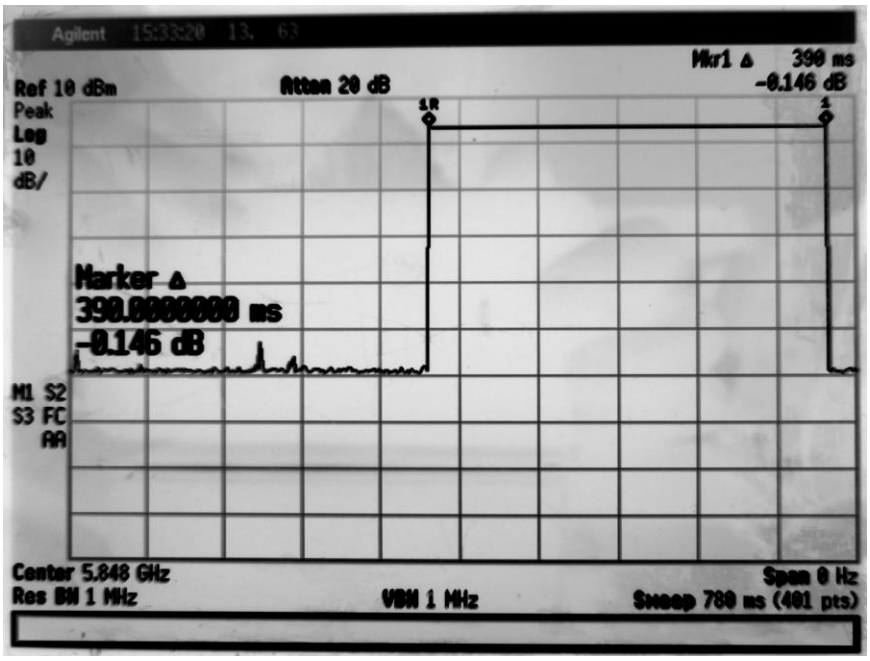

Figure 9 – The hop duration (dwell time) of each frequency must be 400 ms or less

5.5 20 dB Bandwidth

15.247(a)(1)(ii) restricts the 20 dB bandwidth of devices operating in the 5.8 GHz ISM band to a maximum of 1 MHz [26]. The 20 dB bandwidth test used a center frequency

of 5745.8 MHz, span of 2 MHz, RBW and VBW of 30 kHz, sweep set to auto, detector function set to peak, and trace set to max hold. After the trace stabilized, the peak search function set a marker to the peak at 5745.8 MHz. The marker delta function was then used to find the frequency that was approximately -20 dB on the lower sideband. Next, the marker was reset to this point and then the marker delta function used to find the same -20 dB point on the upper sideband by getting the marker delta as close to zero as possible. Figure 10 provides the results of the 20-dB bandwidth test, which shows that 195 kHz is the 20-dB bandwidth of the hopping channel for R.E.S.T. 2.0's current configuration.

Figure 10 – The 20 dB bandwidth must be greater than 25 kHz but less than 1 MHz

5.6 Peak Output Power

15.247(b)(1) restricts the maximum peak output power for devices in the 5.8 GHz ISM band to 1 Watt (W), which is equivalent to +30dBm. The peak output power test used a center frequency of 5745.8 MHz, span of 5 MHz, RBW and VBW of 300 kHz, sweep set to auto, detector function set to peak, and trace set to max hold. For this measurement, 300 kHz was chosen because the RBW must be greater than the 20 dB bandwidth of the hopping channel, and 300 kHz was the lowest resolution on the spectrum analyzer that was greater than 195 kHz. The trace stabilized and then a peak search conducted, which set a marker to the peak of 5745.8 MHz. Figure 11 provides the results of the peak output power test indicating an output power of 5.43 dBm, which is well below the 30 dBm maximum.

An output power that is significantly lower than the authorized level invites the obvious question: how much external amplification can we provide to this signal while remaining within the limits specific in Part 15? The measured amount of output power allows some flexibility for an external amplifier to increase the signal strength by as much as 24.5 dB; however, this test would have to be completely redone because the testing guidance states the testing must be completed with any external signal amplifiers connected [26]. Additionally, the amount of amplification must be reduced by the amount of any additional gain over 6 dBi if the antennas used have a gain greater than 6 dBi, and if external amplification is used, the 20 dB bandwidth of the transmitted signal must be re-measured to ensure it does not exceed 1 MHz [26].

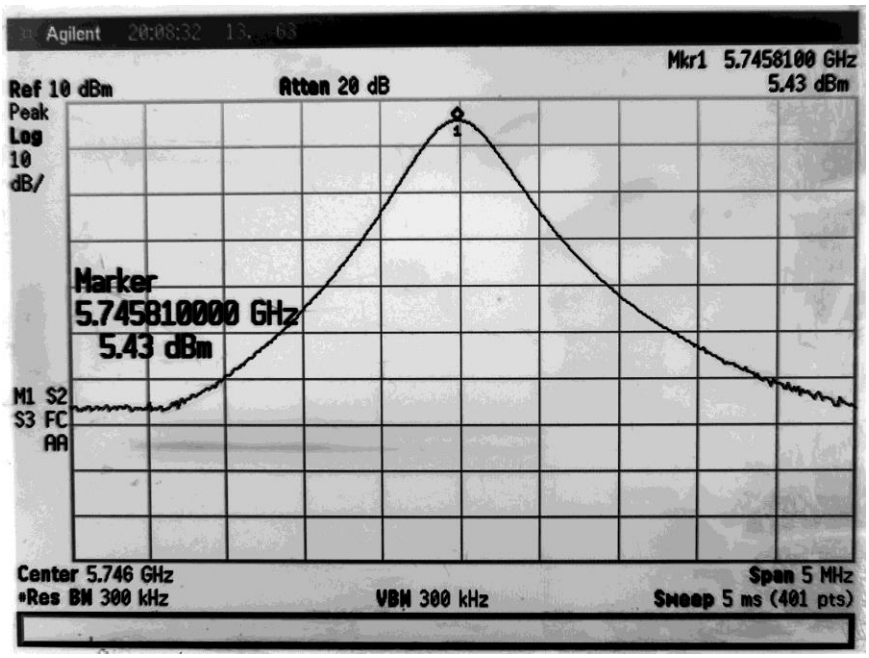

Figure 11 - Peak output power must be less than +30 dBm

5.7 Band-Edge Compliance of RF Conducted Emissions

15.247(d) requires that devices restrict emissions to the intended band of operation. Any unwanted emissions that occur outside the band of operation but before the first harmonic, an area known as the out-of-band domain, is defined as an out-of-band emission [28]. Part 15 requires that any out-of-band emissions be attenuated to 20 dB or more from the characteristic frequency [26]. The band-edge test should use a span wide enough to capture the channel closest to each band edge including any modulation products that appear outside of the authorized band [29]. The test requires two plots from each band edge with the first showing the marker delta of the lowest and highest frequencies in the hopset from the band edge and the second showing that any out-of-band emissions are attenuated at least 20 dB or more from the lowest and highest frequencies in the hopset

[29]. R.E.S.T. 2.0 was tested in its current configuration with no modulation; therefore, band edge testing must be redone in its final configuration to confirm that it still complies with Part 15 band-edge requirements.

The band-edge compliance tests used a center frequency of 5724.2 MHz for the channel closest to the lower edge of the band and 5850.6 MHz for the channel closest to the upper edge of the band. Both tests used a span of 6.4 MHz, RBW and VBW of 100 kHz, sweep set to auto, detector function set to peak, and trace set to max hold. The span of 6.4 MHz was chosen due to the RBW and VBW span dependence and the spectrum analyzer's limited RBW and VBW settings [29]. The band edge compliance test only requires transmitting on the lowest/highest frequency in the hopset whereas the out-of-band emissions test requires frequency hopping to be enabled [29]. For the band edge compliance test, the marker delta function was used after the trace stabilized with one marker on the lowest/highest peak and the other marker identifying the band edge. The results of the band edge tests are provided in Figure 12 (below 5.8 GHz) and Figure 14 (above 5.8 GHz).

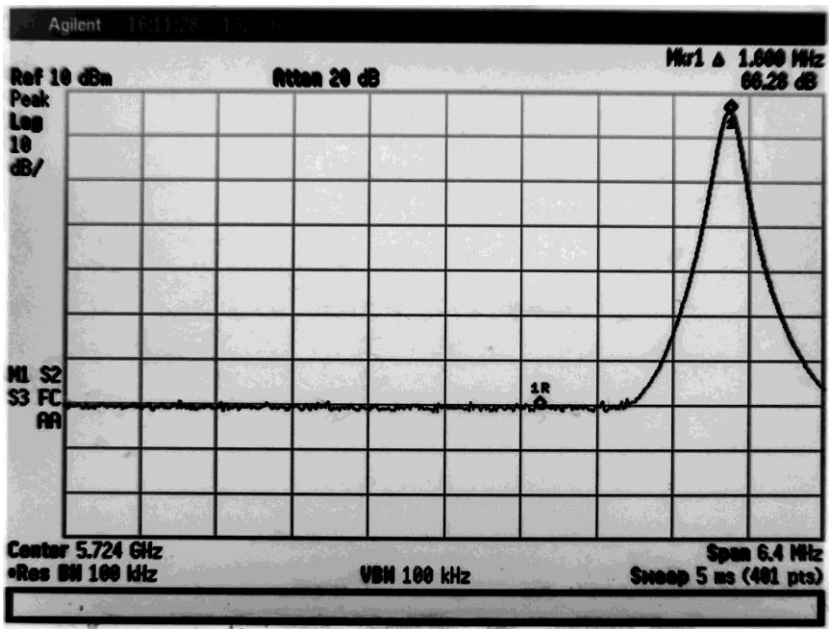

Figure 12 - Band Edge Compliance below below 5.8 GHz. The 1 markers is at the lowest frequency in the hopset, and 1R is at 5725 MHz, the band's lower edge

The out-of-band emissions test used the same instrument settings as the band edge test but with frequency hopping enabled. The system ran with a max hold trace for at least thirty seconds to progress through all hopset frequencies and to allow the trace to stabilize. Then the marker delta function was used with one marker placed on the lowest/highest frequency in the hopset and the other marker placed at the highest observed peak outside the band edge. The results of the out-of-band emissions tests are provided in Figure 13 (below 5.8 GHz) and Figure 15 (above 5.8 GHz). The highest observed peak below the 5.8 GHz band was attenuated by -57.07 dB, and the highest observed peak above the 5.8 GHz band was attenuated by -62.92 dB, which complies with the 20 dB attenuation requirement. Occasional signal spikes in the out-of-band domain below 5.8 GHz were observed during multiple thirty-second tests; however, they were not reproduceable and

thus were determined to not be from R.E.S.T. 2.0. Future testing in an RF isolated environment such as an anechoic chamber is recommended to ensure only emissions from R.E.S.T. 2.0 are observed.

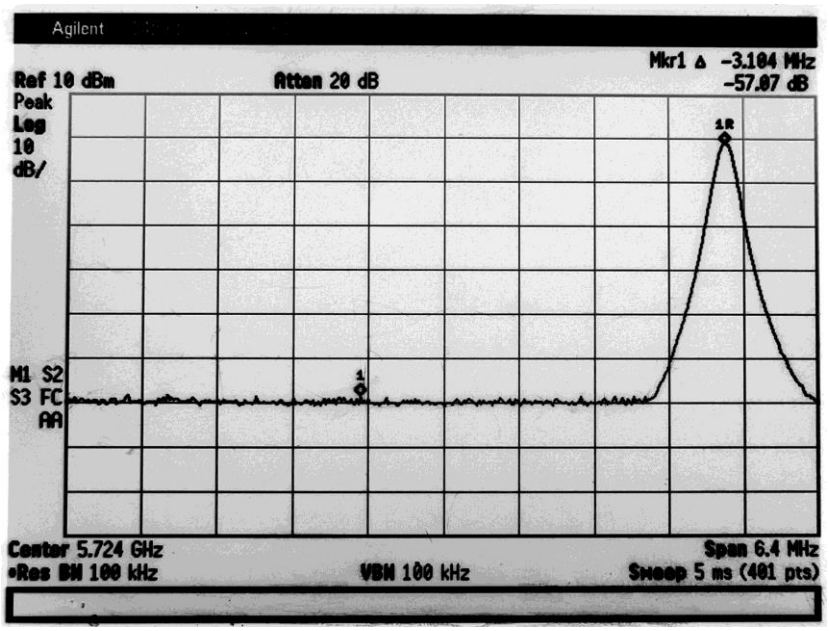

Figure 13- Out-of-band spurious emmissions below 5.8 GHz. No out-of-band spurious emissions from R.E.S.T. 2.0 were observed below the band edge

Figure 14 - Band Edge Compliance above 5.8 GHz The 1 marker is at the highest frequency in the hopset, and the 1R is at 5850 MHz, the upper band edge

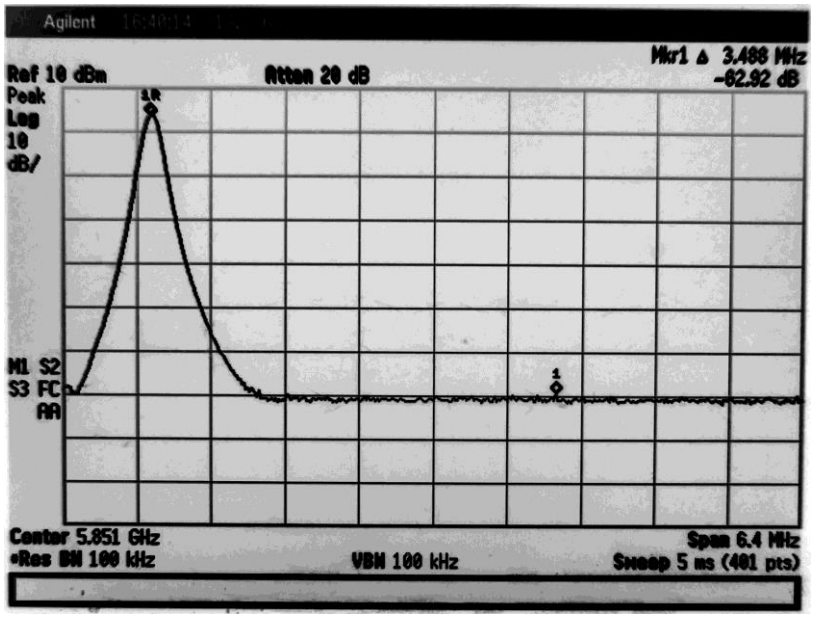

Figure 15 - Out-of-Band Spurious Emissions above 5.8 GHz. No out-of-band spurious emissions from R.E.S.T. 2.0 were observed above the band edge

5.8 Spurious RF Conducted Emissions

Spurious RF emissions are any unwanted emissions that occur in the spurious domain which has a lower bound that begins with the first harmonic and an upper bound of 40 GHz based on the guidance in §15.33(a)(1) [26][28]. The highest frequency in the R.E.S.T. 2.0 frequency hopping protocol is 5848.2 MHz, which produces the harmonics up to 40 GHz shown in Table 4. The sixth harmonic of the highest fundamental frequency is above 40 GHz; therefore, spurious emissions testing is only needed through the fifth harmonic based on the guidance in §15.33(a)(1) [26]. The spectrum analyzer used during testing, unfortunately, has a measurement limit of 26.5 GHz; therefore, extrapolation will be used to estimate the power at the fourth and fifth harmonics.

Table 4 – All 5848.2 MHz harmonics through 40 GHz

Harmonic	Frequency (GHz)
1st	11.6964
2nd	17.5446
3rd	23.3928
4th	29.241
5th	35.0892
6th	40.9374

Due to the span of the harmonics, the spurious emissions testing was captured using multiple plots beginning with the fundamental frequency and relying on the marker delta to measure each harmonic's attenuation from the previous harmonic. The spurious emissions tests used a 6 GHz span, 100 kHz RBW, 100 kHz VBW, and trace set to max hold for all measurements. The center frequencies used were 8.7723 GHz, 14.6205 GHz,

and 20.4687 GHz. Figure 16 provides the measured power at the fundamental frequency and Figures 17, 18, and 19 provide the plots for the first, second, and third harmonics, respectively. Table 5 summarizes the measured power levels from the spurious emissions testing marker delta functions.

Figure 16 - Highest hopset frequency's power and first harmonic

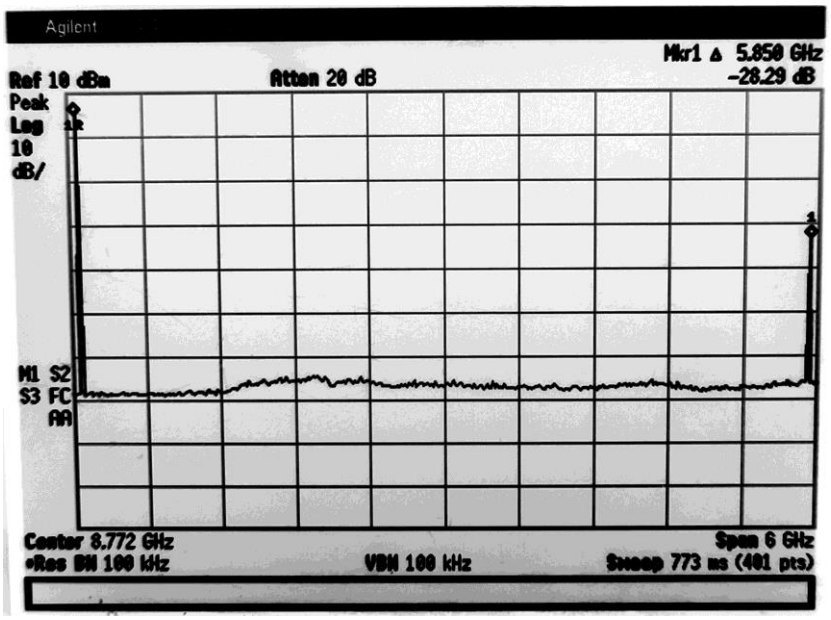

Figure 17 - Highest hopset frequency's first harmonic attenuated by -28.29 dB

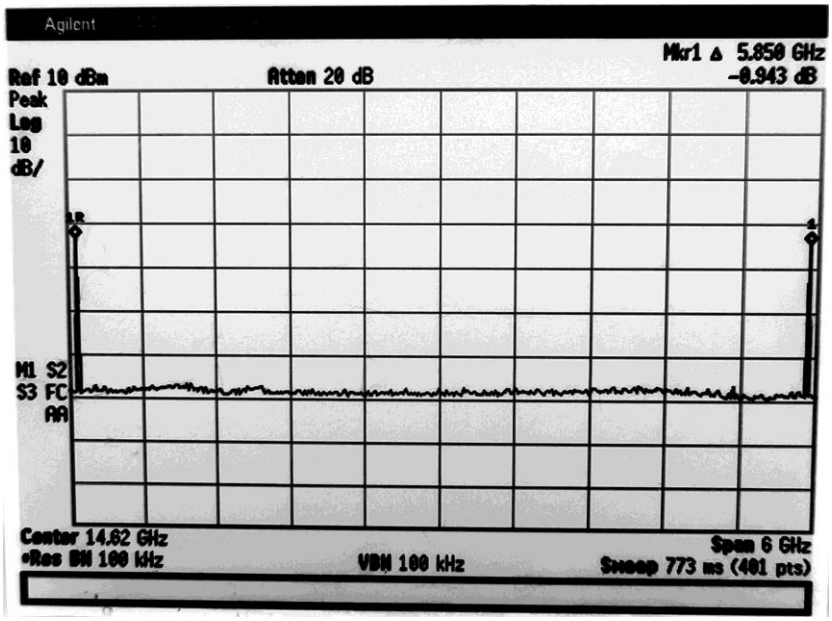

Figure 18 - Highest hopset frequency's second harmonic attenuated by -0.943 dB

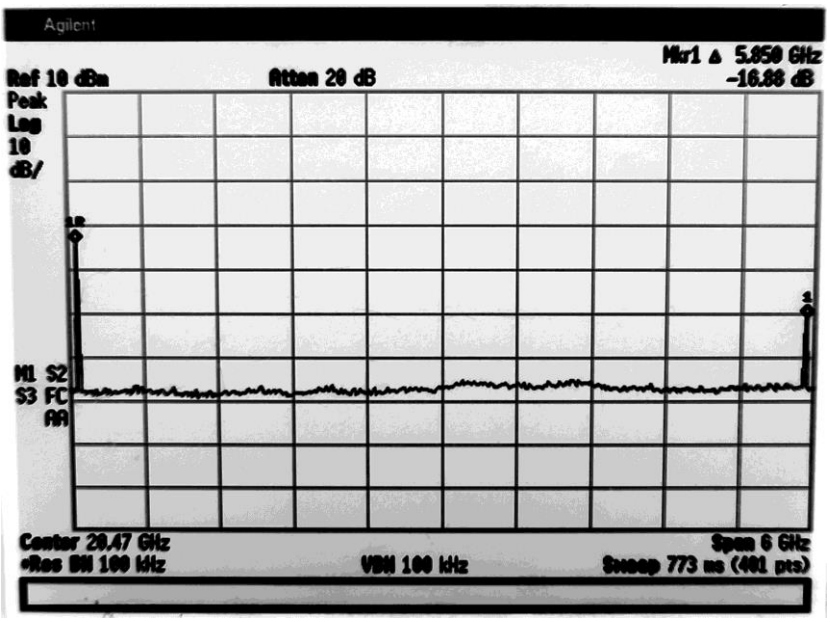

Figure 19 - Highest hopset frequency's third harmonic attenuated by -16.88 dB

Table 5 – Fundamental emission and associated harmonic's power

Harmonic	Frequency (GHz)	Power (dBm)
	5.8482	5.067
1st	11.6964	-23.223
2nd	17.5446	-24.166
3rd	23.3928	-41.046

Using the measured values in Table 5, a graph and trendline was developed to extrapolate the expected power levels at the fourth and fifth harmonics due to the spectrum analyzer's 26.5 GHz measurement limitation. For the trendline, a third order polynomial generated a statistical correlation of 1; however, the calculated power for the fourth and fifth harmonics was significantly lower than expected. In contrast, the logarithmic trendline generated estimates closer to expected power levels despite having a lower

statistical correlation due to the first and second harmonic measurements being nearly identical. Figure 20 provides the plot and associated trendline of the spurious emissions measurements. The trendline in Figure 20 along with the fourth and fifth harmonic frequencies in Table 4 produces calculated power levels of -45.834 dBm (-50.901 dB attenuation) and -51.43 dBm (-56.498 dB attenuation) for the fourth and fifth harmonics, respectively. The measured attenuation of the first three harmonics and calculated attenuation of the fourth and fifth harmonics is greater than -20 dBm, which complies with the requirements of §15.247(d).

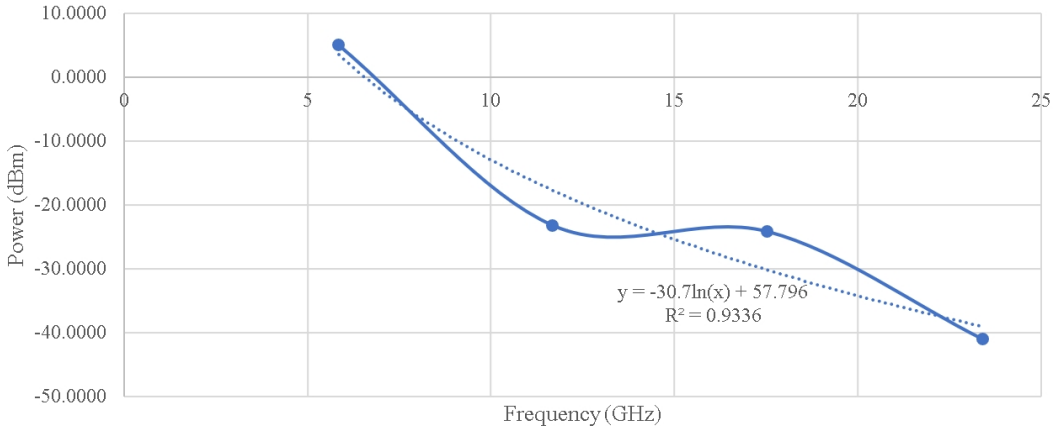

Figure 20 - Trendline correlation for the fourth and fifth harmonics' power

5.9 Spurious Radiated Emissions

§15.205 requires additional testing of spurious radiated emissions that fall within any restricted band of operation, and §15.209 specifies that the field strength of these spurious radiated emissions must not exceed 500 μV/m at 3m for all intentional radiators

operating above 960 MHz [26]. Table 6 shows the restricted bands that contain the first and third harmonics—the only two within restricted bands—and the nearest restricted bands to the second, fourth, and fifth harmonics.

Table 6 - Harmonics that fall within restricted bands [26]

Restricted Band (GHz)	Harmonic	Frequency (GHz)
10.6 - 12.7	1st	11.6964
15.35 – 16.2	2nd (no conflict)	17.5446
17.7 – 21.4		
22.01 – 23.12	3rd	23.3928
23.6 – 24.0	4th (no conflict)	29.241
31.2 – 31.8	5th (no conflict)	35.0892
36.43 – 36.5		

Despite Part 15's requirement for field testing, actual field measurements are beyond the scope of this project because R.E.S.T. 2.0 is still undergoing development and, when complete, there is no intent to seek Part 15 compliance certification; therefore, theoretical calculations will be used to satisfy the requirements for spurious unwanted emissions. The Friis transmission formula in logarithmic form, given in Equation 2, is used to calculate the receive power in dBm at 3m for the first and third harmonics.

$$P_r = P_t + G_t + G_r + 20Log_{10}\left(\frac{\lambda}{4\pi d}\right) \quad (2)$$

The first and third harmonics measured in Table 5 have associated wavelengths of 25.563mm and 12.782mm respectively. Discussions with Cheng Qi, the lead engineer in

designing R.E.S.T. 2.0, revealed that the patch antennas used have an average gain of 6 dBi. Using the measured signal strengths in Table 5 along with the wavelength, antenna gains, and 3m distance; the results of Equation 2 produce a calculated receive signal strength of –74.598 dBm and -98.442 dBm for the first and third harmonics, respectively. A 500 μV/m field strength at 3m corresponds to an equivalent isotropically radiated power (EIRP) of -41.25 dBm [5][33]. The calculated first and third harmonic received powers are far less than the -41.25 dBm limit.

Given these low calculated signal strengths, we can revisit the signal amplification question explored in section 5.6 and determine if using external amplification to maximize the signal strength will cause R.E.S.T. 2.0 to exceed the restricted band EIRP limit at 3 m. Applying the maximum external amplification of 24.5 dB will increase the first and third harmonic transmit powers to 1.277 dBm and -16.546 dBm, respectively. Using Equation 2, the calculated strength of these amplified harmonic signals at 3 m will be -50.1 dBm and -73.9 dBm, which is still less than the -41.25 dBm limit; therefore, R.E.S.T. 2.0 should satisfy the Part 15 requirement for spurious radiated emissions in restricted bands even if maximum external amplification is applied. Note that the calculations presented here are only to show preliminary compliance because Part 15 requires that actual field measurements be performed before filing for compliance certification.

CHAPTER 6. CONCLUSION

Backscatter communication offers exciting possibilities in sensor applications research due to the unique powerless nature of the technology. Passive tags require no dedicated power source because they utilize RF energy from a reader to both power the tag's circuitry and transmit information back to the reader. Backscatter communication systems are typically found within the UHF and microwave ISM bands; however, the 2.4 GHz ISM band, while unlicensed globally, tends to experience more congestion than the 5.8 GHz band, which makes it impractical for many backscatter communication applications. In contrast, the 5.8 GHz band experiences less congestion, has more available bandwidth, and supports higher data rates, which makes it an ideal candidate for backscatter communications.

The Georgia Institute of Technology Propagation Group has a 5.8 GHz backscatter communications system originally built in 2009 that has undergone two revisions with a third currently in progress. The third version, the RFID-Enabled Sensing Testbed (R.E.S.T.), is a flexible hardware and software solution that utilizes interchangeable daughterboards with a frequency-hopping programmable RF front end to support rapid prototyping and troubleshooting of sensors. R.E.S.T. 1.0 is currently still used for enhanced backscatter capabilities sensor research; however, a need existed for future research that implemented the same flexible hardware and software solution but in a low power, space constrained design. The current version under development, R.E.S.T. 2.0, utilizes off-the-shelf microcontroller development kits and RF synthesizer evaluation boards, which significantly decrease the system's size and power requirements.

This project covered the basics of backscatter communication and discussed the important concepts in spread spectrum communications, documented the history of the Georgia Institute of Technology Propagation Group's backscatter communications system from its inception in 2009 through the fourth version currently under development, and developed a frequency hopping protocol using a low-cost, low power microcontroller unit and RF synthesizer board. Measurements and testing were then used to show the fourth version in its current configuration meets Part 15 requirements for intentional radiators. The fourth version is still under development, so the Part 15 measurements and testing compliance was only for its current configuration. If compliance certification is pursued in the future, the measurements performed here must be repeated using the system's final design in all its possible configurations.

APPENDIX A. F28027 FREQUENCY HOPPING CODE

```
//###########################################################################
// Frequency Hopping code for Texas Instruments LaunchXL-F28027 Launchpad
// LaunchXL-F28027 interfaces with Texas Instruments LMX2592EVM
//
// Code meets CFR 47, Chapter 1, Part 15 requirements
//
// Robert W. Corless
// Propagation Group, Georgia Institute of Technology
// November 2017
// Written in Code Composer Studio ver7.1.0
//###########################################################################
#include "DSP28x_Project.h"     // Device Headerfile and Examples Include File
// interrupt void ISRTimer2(void);
__interrupt void cpu_timer0_isr(void);
__interrupt void cpu_timer1_isr(void);
void delay_loop(void);
void spi_fifo_init(void);
void error();
void pll_init(void);
uint16_t sdata[3];     // Send data buffer
uint16_t rdata[2];     // Receive data buffer
uint16_t rdata_point;  // Keep track of where we are
                       // in the data stream to check received data
```

```
uint16_t pll_register[43] = {0x4000, 0x3E00, 0x3D00, 0x3B00, 0x3000, 0x2F00, 0x2E00,
0x2D00, 0x2C00, 0x2B00, 0x2A00, 0x2900, 0x2800, 0x2700, 0x2600, 0x2500, 0x2400,
0x2300, 0x2200, 0x2100, 0x2000, 0x1F00, 0x1E00, 0x1D00, 0x1C00, 0x1900, 0x1800,
0x1700, 0x1600, 0x1400, 0x1300, 0x0E00, 0x0D00, 0x0C00, 0x0B00, 0x0A00, 0x0900,
0x0800, 0x0700, 0x0400, 0x0200, 0x0100, 0x0000};

uint16_t pll_data1[43] =    {0x0000, 0x0000, 0x0000, 0x0000, 0x0300, 0x0800, 0x0F00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0300, 0x0000, 0x8200, 0x0000, 0x4000, 0x0000,
0x0200, 0xC300, 0x2A00, 0x2100, 0x0400, 0x0000, 0x0000, 0x2900, 0x0000, 0x0500,
0x8800, 0x2300, 0x0100, 0x0900, 0x0100, 0x4000, 0x7000, 0x0000, 0x1000, 0x0300,
0x1000, 0x2800, 0x1900, 0x0500, 0x0800, 0x2200};

uint16_t pll_data2[43] =    {0x7700, 0x0000, 0x0100, 0x0000, 0xFC00, 0xCF00, 0xA300,
0x7900, 0x0000, 0x0000, 0x0000, 0xE800, 0x0000, 0x0400, 0x3A00, 0x0000, 0x1100,
0x1F00, 0xEA00, 0x0A00, 0x0A00, 0x0100, 0x3400, 0x8400, 0x2400, 0x0000, 0x0900,
0x4200, 0x0000, 0x2C00, 0x6500, 0x8C00, 0x0000, 0x0100, 0x1800, 0xD800, 0x0200,
0x8400, 0xB200, 0x4300, 0x0000, 0x0800, 0x1C00};

uint16_t pll_point = 0;


uint16_t pll_FrHpNdivider[77]   = {0x3800, 0x3A00, 0x3800, 0x3A00, 0x3800, 0x3A00,
0x3800, 0x3800, 0x3800, 0x3A00, 0x3A00, 0x3800, 0x3800, 0x3A00, 0x3800, 0x3A00,
0x3800, 0x3800, 0x3A00, 0x3800, 0x3800, 0x3A00, 0x3800, 0x3800, 0x3A00, 0x3800,
0x3800, 0x3A00, 0x3800, 0x3A00, 0x3800, 0x3A00, 0x3800, 0x3800, 0x3800, 0x3800,
0x3A00, 0x3A00, 0x3A00, 0x3A00, 0x3800, 0x3A00, 0x3800, 0x3A00, 0x3800, 0x3A00,
0x3800, 0x3800, 0x3A00, 0x3800, 0x3800, 0x3A00, 0x3800, 0x3A00, 0x3A00, 0x3800,
0x3A00, 0x3800, 0x3800, 0x3A00, 0x3800, 0x3800, 0x3800, 0x3A00, 0x3800, 0x3A00,
0x3800, 0x3800, 0x3A00, 0x3800, 0x3800, 0x3A00, 0x3800, 0x3800, 0x3800, 0x3800,
0x3A00}; // N div: R38

uint16_t pll_FrHpNumerator1[77] = {0x0200, 0x0000, 0x0300, 0x0000, 0x0200, 0x0000,
0x0300, 0x0200, 0x0300, 0x0000, 0x0000, 0x0300, 0x0300, 0x0000, 0x0200, 0x0000,
0x0200, 0x0300, 0x0000, 0x0200, 0x0300, 0x0000, 0x0300, 0x0300, 0x0000, 0x0200,
0x0300, 0x0000, 0x0200, 0x0000, 0x0200, 0x0000, 0x0200, 0x0300, 0x0200, 0x0200,
0x0000, 0x0000, 0x0000, 0x0000, 0x0200, 0x0000, 0x0300, 0x0000, 0x0300, 0x0000,
0x0300, 0x0200, 0x0000, 0x0300, 0x0300, 0x0000, 0x0300, 0x0000, 0x0000, 0x0300,
0x0000, 0x0300, 0x0300, 0x0000, 0x0300, 0x0300, 0x0300, 0x0000, 0x0300, 0x0000,
0x0200, 0x0300, 0x0000, 0x0300, 0x0300, 0x0000, 0x0200, 0x0300, 0x0200, 0x0300,
0x0000}; // PLL Num: R45 MS 8 bits

uint16_t pll_FrHpNumerator2[77] = {0xD900, 0x9900, 0x8100, 0x1900, 0xF100, 0x4100,
0xA900, 0x9900, 0x6100, 0xF100, 0x1100, 0x2100, 0xA100, 0x6900, 0x7900, 0x8900,
0x9100, 0x4100, 0x0100, 0xA100, 0x5100, 0x6100, 0xB900, 0x5900, 0xD900, 0x8100,
0xC900, 0x7100, 0xB100, 0xB900, 0xE100, 0x2100, 0xA900, 0xB100, 0xE900, 0x8900,
0xD100, 0x0900, 0xA100, 0x3100, 0xD100, 0xC100, 0xC100, 0xE100, 0x3100, 0x2900,
```

```
0x9100, 0xC900, 0xB100, 0x1900, 0x8900, 0x5900, 0x0900, 0xA900, 0x4900, 0x3900,
0xE900, 0x1100, 0x9900, 0x8100, 0xD900, 0x0100, 0xD100, 0x5100, 0x7900, 0x3900,
0xB900, 0x2900, 0x9100, 0x4900, 0xE100, 0xC900, 0xC100, 0x7100, 0xF900, 0x6900,
0x7900}; // PLL Num: R45 LS 8 bits

// Note, the first frequency programmed using timer0 is 5824.2 MHz (freq #1)

// The first values in the FrHp integers are for 5745.8 MHz (freq #2)

// The last values in the FrHp integers are for 5824.2 MHz (freq #1)

uint16_t freq_count = 0;

uint16_t change_step = 0;

void main(void)

{

  uint16_t i;

// WARNING: Always ensure you call memcpy before running any functions from RAM

// InitSysCtrl includes a call to a RAM based function and without a call to

// memcpy first, the processor will go "into the weeds"

  #ifdef _FLASH

   memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, (size_t)&RamfuncsLoadSize);

  #endif


// Step 1. Initialize System Control:

// PLL, WatchDog, enable Peripheral Clocks

// This example function is found in the f2802x_SysCtrl.c file.

  InitSysCtrl();


// Step 2. Initialize GPIO:

// Setup the GP I/O only for SPI-A functionality
```

```
   InitSpiaGpio();


// Step 3. Initialize PIE vector table:

// Disable and clear all CPU interrupts

   DINT;

   IER = 0x0000;

   IFR = 0x0000;


// Initialize PIE control registers to their default state:

   InitPieCtrl();


// Initialize the PIE vector table with pointers to the shell Interrupt Service Routines (ISR).

   InitPieVectTable();


// Interrupts that are used in this example are re-mapped to ISR functions found within this file.

   EALLOW;  // This is needed to write to EALLOW protected registers

   PieVectTable.TINT0 = &cpu_timer0_isr;  // CPU-Timer 0 interrupt signal - Initialization

   PieVectTable.TINT1 = &cpu_timer1_isr;  // CPU-Timer 1 interrupt signal - Frequency Hopping

   // CPU-Timer 2 is reserved for DSP/BIOS (if needed).  If DSP/BIOS not needed, can use for applications


// Step 4. Initialize all the Device Peripherals:

   spi_fifo_init();   // Initialize the SPI only

   InitCpuTimers();   // For this example, only initialize the CPU Timers
```

```
#if (CPU_FRQ_60MHZ)
// Configure CPU-Timer 0, 1, and 2 as interrupts
// 60MHz CPU Freq, 1 second Period (in uSeconds)

   ConfigCpuTimer(&CpuTimer0, 60, 1000);  // 1ms delay after step 6 before sending PLL register data
   ConfigCpuTimer(&CpuTimer1, 60, 390000);  // occupy freq for 390ms then hop to next frequency
   ConfigCpuTimer(&CpuTimer2, 60, 10000);
#endif
#if (CPU_FRQ_50MHZ)
// Configure CPU-Timer 0, 1, and 2 to interrupt every second:
// 50MHz CPU Freq, 1 second Period (in uSeconds)

   ConfigCpuTimer(&CpuTimer0, 50, 1000000);
   ConfigCpuTimer(&CpuTimer1, 50, 1000000);
   ConfigCpuTimer(&CpuTimer2, 50, 1000000);
#endif
#if (CPU_FRQ_40MHZ)
// Configure CPU-Timer 0, 1, and 2 to interrupt every second:
// 40MHz CPU Freq, 1 second Period (in uSeconds)

   ConfigCpuTimer(&CpuTimer0, 40, 1000000);
   ConfigCpuTimer(&CpuTimer1, 40, 1000000);
   ConfigCpuTimer(&CpuTimer2, 40, 1000000);
```

```
#endif


   CpuTimer0Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 1, Starts
CpuTimer0
   // 0 = stop timer, 1 = start/restart timer


// Step 5. User specific code, enable interrupts:
// Initialize the send data buffer
   for(i=0; i<2; i++)
   {
     sdata[i] = i;
   }
   rdata_point = 0;


// Enable interrupts required for this example
   PieCtrlRegs.PIECTRL.bit.ENPIE = 1;   // Enable the PIE block
   PieCtrlRegs.PIEIER1.bit.INTx7 = 1;   // Enable TINT0 in the PIE: Group 1 interrupt 7


   IER=0x20;                        // Enable CPU INT6
   // Enable CPU int1 which is connected to CPU-Timer 0,
   //       CPU int13 which is connected to CPU-Timer 1
   //       CPU int 14, which is connected to CPU-Timer 2
   IER |= M_INT1;
   IER |= M_INT13;
   EINT;   // Enable Global Interrupts
   ERTM;   // Enable Global realtime interrupt DBGM
```

```
// Step 6. IDLE loop. Just sit and loop forever (optional):
    for(;;);
}

// Some Useful local functions
void delay_loop()
{
    long      i;
    for (i = 0; i < 1000000; i++) {}
}

void error(void)
{
    __asm("     ESTOP0");    //Test failed!! Stop!
    for (;;);
}

void spi_fifo_init()
{
// Initialize SPI FIFO registers
   SpiaRegs.SPICCR.bit.SPISWRESET=0; // Reset SPI

   SpiaRegs.SPICCR.all=0x0047;       //16-bit character, Loopback mode
   SpiaRegs.SPICTL.all=0x0006;       //Interrupt enabled, Master/Slave XMIT enabled
```

```
  //SpiaRegs.SPISTS.all=0x0000;
  SpiaRegs.SPIBRR=0x0063;          // Baud rate
  SpiaRegs.SPIFFTX.all=0xC022;     // Enable FIFO's, set TX FIFO level to 2
  SpiaRegs.SPIFFRX.all=0x0022;     // Set RX FIFO level to 2
  SpiaRegs.SPIFFCT.all=0x00;
  SpiaRegs.SPIPRI.all=0x0010;

  SpiaRegs.SPICCR.bit.SPISWRESET=1;  // Enable SPI

  SpiaRegs.SPIFFTX.bit.TXFIFO=1;
  SpiaRegs.SPIFFRX.bit.RXFIFORESET=1;
}
__interrupt void cpu_timer0_isr(void)  // PLL Register Programming ISR
{
   if(pll_point<=43)
   {
      sdata[0] = pll_register[pll_point];
      sdata[1] = pll_data1[pll_point];
      sdata[2] = pll_data2[pll_point];
      pll_point++;
   }
 //rdata[0]=0;
 /* if(pll_point<=1)
      {
         sdata[0] = 0x0000;
```

```
            sdata[1] = 0x2200;
            sdata[2] = 0x1C00;
            pll_point++;
        }*/
    else
    {
      PieCtrlRegs.PIEIER1.bit.INTx7 = 0;  // Disable TINT0 in the PIE: Group 1 interrupt 7 (Stop CpuTimer0)
      CpuTimer1Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 1 (Start/Restart CpuTimer1)
    }

    // Transmit data
    uint16_t i;
       for(i=0;i<3;i++)
       {
          SpiaRegs.SPITXBUF=sdata[i];      // Send data
       }

   // Acknowledge this interrupt to receive more interrupts from group 1
   PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

__interrupt void cpu_timer1_isr(void) // Frequency Hopping ISR
{
```

```
   CpuTimer1Regs.TCR.all = 0x4000; // Use write-only instruction to set TSS bit = 0 (Stop
CpuTimer1)
   //
   ////////// FREQUENCY CHANGE - UPDATE REGISTER VALUES //////////
   //
   if(freq_count<76)
   {
      pll_data2[14] = pll_FrHpNdivider[freq_count];   // PLL_N value
      pll_data1[7] = pll_FrHpNumerator1[freq_count];  // PLL-Num value (MS 8 bits)
      pll_data2[7] = pll_FrHpNumerator2[freq_count];  // PLL-Num value (LS 8 bits)
      freq_count++;  // increment freq_count to select the next frequency

   }
   else // else statement programs the last frequency in the hopset
   {
      pll_data2[14] = pll_FrHpNdivider[freq_count];   // PLL_N value
      pll_data1[7] = pll_FrHpNumerator1[freq_count];  // PLL-Num value (MS 8 bits)
      pll_data2[7] = pll_FrHpNumerator2[freq_count];  // PLL-Num value (LS 8 bits)
      freq_count = 0;

   }
   //
   ////////// TRANSMIT NEW REGISTER VALUES TO PLL //////////
   //
   for(change_step=0; change_step<3; change_step++)
   {
```

```
if(change_step==0)  // Write R38 (PLL N divider value)
{
   sdata[0] = pll_register[14];   // write R38=0x2600
   sdata[1] = pll_data1[14];      // write R38 value (MS 8 bits)
   sdata[2] = pll_data2[14];      // write R38 value (LS 8 bits)
}
if(change_step==1)  // Write R45 (numerator value)
{
   sdata[0] = pll_register[7];   // write R45=0x2D00
   sdata[1] = pll_data1[7];      // write R45 value (MS 8 bits)
   sdata[2] = pll_data2[7];      // write R45 value (LS 8 bits)
}
if(change_step==2)  // Write R0 0x00221C
{
   sdata[0] = pll_register[42];   // write R0=0x0000
   sdata[1] = pll_data1[42];      // write R0 value (MS 8 bits)
   sdata[2] = pll_data2[42];      // write R0 value (LS 8 bits)
}
uint16_t i;
for(i=0;i<3;i++)
{
   SpiaRegs.SPITXBUF=sdata[i]; // Send data
}
uint16_t d; // delay after writing each register to allow CSB latch enable
for (d=0;d<870;d++)
```

```
        {

        }

    }

}

//=============================================================
=============

// No more.

//=============================================================
=============
```

REFERENCES

[1] L. Columbus, “2017 roundup of Internet of Things forecasts,” *Forbes*, 10 Dec 2017. [Online], Available: https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/#4e405c501480. Accessed: 5 Mar 2018.

[2] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, West Sussex, U.K.: John Wiley and Sons, 2010.

[3] F. Amato, H. M. Torun and G. Durgin, "RFID Backscattering in Long-Range Scenarios," in *IEEE Trans. Wireless Commun.*, vol. PP, no. 99, pp. 1-1. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8288609&isnumber=4656680 . Accessed: 5 Mar 2018.

[4] D. M. Dobkin, *The RF in RFID: Passive UHF RFID in Practice.* Burlington, MA: Newnes, 2008.

[5] H. Lehpamer. *RFID Design Principles*. 2nd ed. Norwood, MA: Artech House, 2012.

[6] G.D. Durgin. “The Hidden Benefits of Backscatter at 5.8 GHz,“ URSI 2008, Boulder, CO, January 2008.

[7] The Propagation Group (2018). *Backscatter Radio*. Accessed on: Feb. 27, 2018. [Online]. Available: http://www.propagation.gatech.edu/backscatter-radio/.

[8] M. S. Gast, *802.11 Wireless Networks, The Definitive Guide.* 2nd ed. Sebastapol, CA: O’Reilly, 2005.

[9] A. Goldsmith, *Wireless Communications*. New York, NY: Cambridge University Press, 2005.

[10] D. J. Torrieri, *Principles of Spread Spectrum Communication Systems*, 2nd ed. New York, NY: Springer, 2011.

[11] A. B. Carlson, P. B. Crilly, and J. C. Rutledge, *Communication Systems: A Introduction to Signals and Noise in Electrical Communication*. 4th ed. New York, NY: McGraw-Hill, 2002.

[12] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Upper Saddle River, NJ: Prentice Hall, 1996.

[13] M. K. Simon *et al*, *Spread Spectrum Communications Handbook*, New York, NY: McGraw Hill, 2002.

[14] EPCglobal, EPC Radio Frequency Identification Protocol Class 1, Generation 2, UHF RFID, protocol for communications at 860 MHz to 960 MHz, version 2.0.1, 2015. Accessed on Mar 6, 2018. [Online]. Available: https://www.gs1.org/sites/default/files/docs/epc/Gen2_Protocol_Standard.pdf.

[15] M. B. Akbar, D. G. Taylor, and G. D. Durgin, "Hybrid Inertial Microwave Reflectometry for mm-scale Tracking in RFID systems", *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6805-6814, Dec 2015.

[16] M. B. Akbar, "Hybrid inertial microwave reflectometry for mm-scale tracking in RFID systems" Ph.D. dissertation. The Georgia Institute of Technology, Atlanta, 2016.

[17] J.D. Griffin, "High-frequency modulated backscatter communication using multiple antennas" Ph.D. dissertation. The Georgia Institute of Technology, Atlanta, 2009.

[18] G.D. Durgin et al, "Georgia Tech experimental backscatter air interface and hardware specification: GTX1.0," unpublished.

[19] C. Valenta and G. Durgin, "R.E.S.T. – A flexible, semi-passive platform for developing RFID technologies", in Sensors, 2012 IEEE, 2012, pp.1-4.

[20] Texas Instruments, "High Performance, Wideband PLLatinum RF Synthesizer With Integrated VCO," LMX2592 datasheet SNAS646E, Dec. 2015 [Revised Jul. 2017].

[21] Texas Instruments, "TMS320F2802x Piccolo Microcontrollers", datasheet SPRS523K, Nov. 2008 [Revised Jun. 2016].

[22] Texas Instruments, “LMX2592EVM High Performance, Wideband PLLatinum RF Synthesizer Evaluation Board Operating Instructions,” User’s Guide SNAU195, Dec. 2015.

[23] Texas Instruments, “Piccolo F2802x C2000_Launchpad,” C2000_Launchpad schematic, Apr. 2012.

[24] Texas Instruments, “TMS320x2802x, 2803x Piccolo Serial Peripheral Interface (SPI),” SPRUG71B Reference Guide, Feb. 2009 [Revised Oct. 2009].

[25] Texas Instruments, “LAUNCHXL-F28027 C2000 Piccolo LaunchPad Experimenter Kit,” User’s Guide SPRUHH2A, Jul. 2012 [Revised Jan. 2014].

[26] U.S. National Archives and Records Administration. 2018. *Code of Federal Regulations*. Title 47, Chapter 1, Sub-Chapter A, Part 15—Radio Frequency Devices.

[27] Federal Communications Commission (2017, April 4). *Basic Equipment Authorization Guidance For Antennas Used With Part 15 Intentional Radiators*, Publication No. 353028. Office of Engineering and Technology Laboratory Division. Accessed on: Feb. 23, 2018. [Online]. Available: https://apps.fcc.gov/oetcf/kdb/forms/FTSSearchResultPage.cfm?switch=P&id=39060

[28] U.S. National Archives and Records Administration. 2018. *Code of Federal Regulations*. Title 47, Chapter 1, Sub-Chapter A, Part 2. Subpart A—Terminology.

[29] Federal Communications Commission (30 Mar 2000). *Filing and Measurement Guidelines for Frequency Hopping Spread Spectrum Systems,* Publication DA 00-705. Public Notice. Accessed on: Feb 12, 2018. [Online]. Available: https://www.fcc.gov/document/filing-and-measurement-guidelines-frequency-hopping-spread-spectrum-systems.

[30] U.S. National Archives and Records Administration. 2018. *Code of Federal Regulations*. Title 47, Chapter 1, Sub-Chapter A, Part 1. Subpart I. Section 1310—Radiofrequency radiation exposure limits

[31] Federal Communications Commission (16 Apr 2007), *Passive tags used with frequency hopping tag reading systems operating in Section 15.247. Frequently asked questions*. Publication No. 205122. Office of Engineering and Technology Laboratory Division. Accessed on 13 Feb 2018. [Online]. Available: https://apps.fcc.gov/kdb/GetAttachment.html?id=sdQ3PRald1NrhI%2B5Zrgdag%3D%3D&desc=Passive Tag Policy&tracking_number=23378.

[32] S. Dayhoff, "New Policies for Part 15 Devices," Federal Communications Commission, Office of Engineering and Technology, Laboratory Division. May 13, 2005. Accessed on Feb 12, 2018. [Online]. Available: https://transition.fcc.gov/oet/ea/presentations/files/may05/New_Policies_Pt._15_SD.pdf

[33] Semtech. "LoRa and FCC Part 15.247: Measurement Guidance" AN1200.26, May 2015. Accessed Aug 30, 2017. [Online]. Available: https://www.semtech.com/uploads/documents/an1200.26.pdf