REMOTE PROGRAMMING OF IMPLANTABLE CARDIAC DEVICES UTILIZING REAL-TIME ACTIVE TELEMEDICINE

by

Nicholas Griesmer Franconi

BS in Electrical Engineering, University of Pittsburgh, 2010

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2012

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Nicholas Griesmer Franconi

It was defended on

March 30, 2012

and approved by

William Stanchina, PhD, Chairman and Professor, Department of Electrical and Computer Engineering

Zhi-Hong Mao, PhD, Assistant Professor, Department of Electrical and Computer Engineering and Department of Bioengineering

Thesis Advisor: Marlin H. Mickle, PhD, Nickolas A. DeCecco Professor, Department of Electrical and Computer Engineering

REMOTE PROGRAMMING OF IMPLANTABLE CARDIAC DEVICES UTILIZING REAL-TIME ACTIVE TELEMEDICINE

Nicholas Griesmer Franconi, M.S.

University of Pittsburgh, 2012

Telemedicine is the exchange of medical information from one site to another in order to improve patient health and well-being. Telecardiology is the application of telemedicine to cardiology. With over one million pacemakers and implantable cardiac defibrillators (ICDs) implanted every year, the need for a telecardiology system that can remotely interrogate and program pacemakers has increased over the past decade. Current advances in telecardiology can greatly increase the quality of life of a patient with a pacemaker. Interrogation of pacemakers occurs every six to twelve months and involves examining the battery capacity and peak voltages of pulses that are transmitted to the chamber walls inside the heart.

Current telecardiology techniques only allow the cardiologist to remotely monitor a patient. The system of real-time active telecardiology implemented in this thesis allows for remote interrogation and programming capabilities through a Medtronic CareLink 2090 Programmer. The system samples touchscreen location data from a commercial-off-the-shelf touchscreen controller and transmits this information to a remote programmer which registers the data as a touch. Through this method, the cardiologist is able to remotely control a Medtronic 2090 Programmer with little change to the workflow of the cardiologist and at minimal cost to the hospital.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to express my gratitude to Dr. Marlin Mickle, Dr. David Schwartzman and Michael Rothfuss for their invaluable support and guidance. I am forever grateful for the opportunities they have provided for me.

1.0 INTRODUCTION

The demand to remotely monitor implantable cardiac devices has drastically increased over the past decade, as the number of cardiac devices implanted per year has risen from just approximately 500,000 pacemakers in 2005 to over a million pacemakers in 2009[1] [2]. The mean age of 65 years has remained constant from 2005 to 2009[1] [2]. The combination of these two factors has driven the need for a solution to remotely program these implantable cardiac devices. The following subsections will introduce telemedicine and its application to cardiology and implantable cardiac devices. Technologies that are currently being utilized in industry will be examined and a method to actively program an implantable cardiac devices in real-time will be introduced.

1.1 BACKGROUND

An arrhythmia is a condition affecting either the rhythm or the rate of a heartbeat [3]. A pacemaker is implanted in the shoulder area of a patient and sends continuous electrical pulses to a chamber of the heart to control arrhythmias shown in Figure 1 [4]. A pacemaker is composed of a pulse generator, a battery and other electronic circuitry which are joined into an implantable device. Leads are connected to the pacemaker and to the chamber walls inside the heart. The number of leads is dependent on the type of arrhythmia of the patient. The leads transmit the pulses generated by the pacemaker to the heart tissue controlling the heart rate as well as synchronizing the multiple chambers of the heart [4]. An implantable cardiac defibrillator (ICD) performs a similar function as the pacemaker but is implanted for different reasons. Some arrhythmias can periodically cause the heart to stop beating; the ICD sends electrical shocks to the heart until a normal heart rate is achieved [6]. Because programming of ICDs and pacemakers are similar and the number of pacemakers implanted is much greater than the number of ICDs, pacemakers will be the primary discussion in this section.

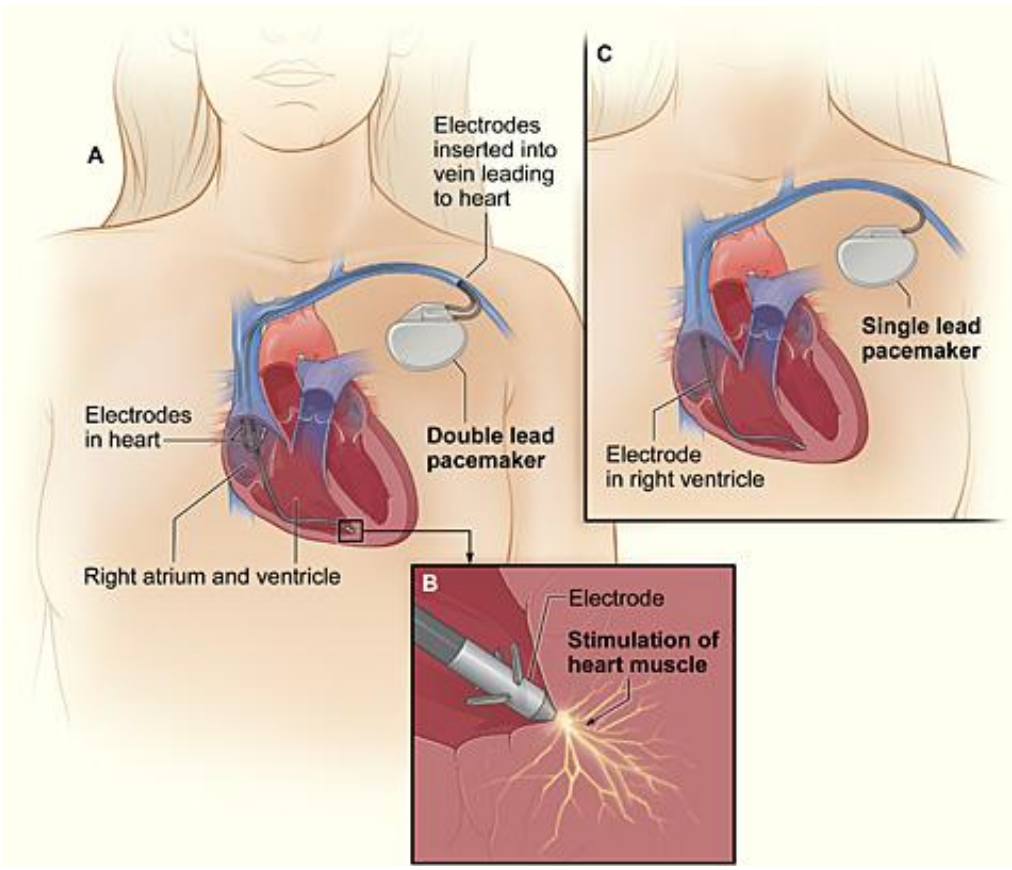

Figure 1: Image of Implanted Pacemaker and Leads to the Heart [4]

In order for the pacemaker to control the heart rate, it must monitor the heart's electrocardiogram (ECG), the heart's electrical activity shown in Figure 2. The ECG is of major importance when the cardiologist is interrogating the pacemaker. Pacemakers and ICDs are both battery operated so the devices must be checked every six to twelve months [7]. These checkups ensure that the battery is functioning properly and the pulses generated are properly maintaining the heart rate as the voltage from the battery begins to decrease. The battery on both a pacemaker and ICD will last anywhere between 5 to 15 years [7]. This battery lifetime varies depending on the device type and the number of leads which are sending electrical pulses to the heart. When the battery reaches a low threshold, the pulse generator and battery unit must be replaced via a surgical procedure. This replacement surgery is much less invasive then the initial surgery. The leads attached to the heart do not need to be replaced and can be disconnected from the pacemaker.

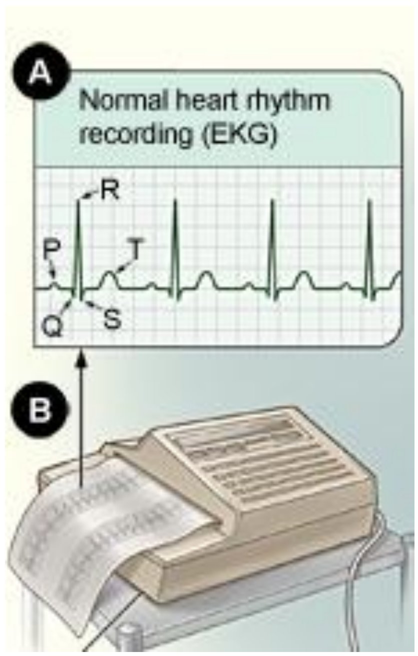

Figure 2: Image of an ECG [5]

The application of telemedicine techniques to cardiology is known as telecardiology. The first record of telecardiology was in London, England in 1901. Willem Einthoven was a professor of physiology at the University of Leiden and the founder of modern electrocardiography [8]. Einthoven transmitted the electrocardiograms of patients from the hospital to his academic research lab for further testing [9]. The theory of this first telecardiology system is very similar to current techniques that are in practice today.

1.2 CURRENT TELECARDIOLOGY TECHNOLOGIES

The initial idea to implement a system of real-time active telecardiology was introduced by a cardiologist, Dr. David Schwartzman, at the University of Pittsburgh Medical Center (UPMC). He performs surgeries on patients from all over North America and needed to do more than the remote monitoring solutions allowed. Current telecardiology systems exist that allow for remote monitoring of patients from their home, although travel into the hospital for even minor adjustments on a shorter interval can be costly to both the hospital and the patient [10] [11]. Since the mean age of the recipient is 65 years, the burden of travel to check up appointments is often placed on the family of the patient. Thus a solution was created to allow for a secure system of real-time active telecardiology.

There are a number of companies that produce pacemakers and the interrogations devices used to program them [12][13][14]. Although the pacing theory is similar among these companies, the programmers, which interrogate the pacemakers, utilize a company specific communication protocol. Only programmers made by each company can interrogate and program their own pacemakers. A decision was made to select the Medtronic hardware due to the availability of the programmers and the large number of Medtronic pacemakers in circulation worldwide. Although the solution, expressed in this thesis, is specific to Medtronic hardware, the underlining theory presented can be applied to any pacemaker manufacturer's hardware that contains a touchscreen.

1.2.1 Types of Telemedicine

Telecardiology technology is broken down into three separate categories: store-and-forward, remote monitoring and real-time active. Each of these categories consists of different telecommunications methods of accessing and storing data over a network. The following sections will introduce these types of telecommunications and their applications to telecardiology.

1.2.1.1 Store-and-Forward System

Store-and-forward is a telecommunications technique that stores data at an intermediate station until the information can be analyzed. In the case of telecardiology, a patient has their pacemaker interrogated at a local hospital. The information is stored locally and then transmitted to a cardiologist at remote location for analysis. If the cardiologist is able to identify an issue with the operation of the pacemaker, he informs the patient that adjustments to the pacemaker are required. This method allows patients to receive care from cardiologists all over the world but presents two major problems. The analysis of the ECG is not done in real-time and the patient must travel to a hospital multiple times for a single evaluation. Also, the hospital must have equipment and staff that can properly interrogate a pacemaker. This method was the first telecardiology system and is still used as a solution. Figure 3 shows the graphical representation of store-and-forward. The Medtronic 2090 Programmer stores information until the cardiologist requests the information and can analyze it.

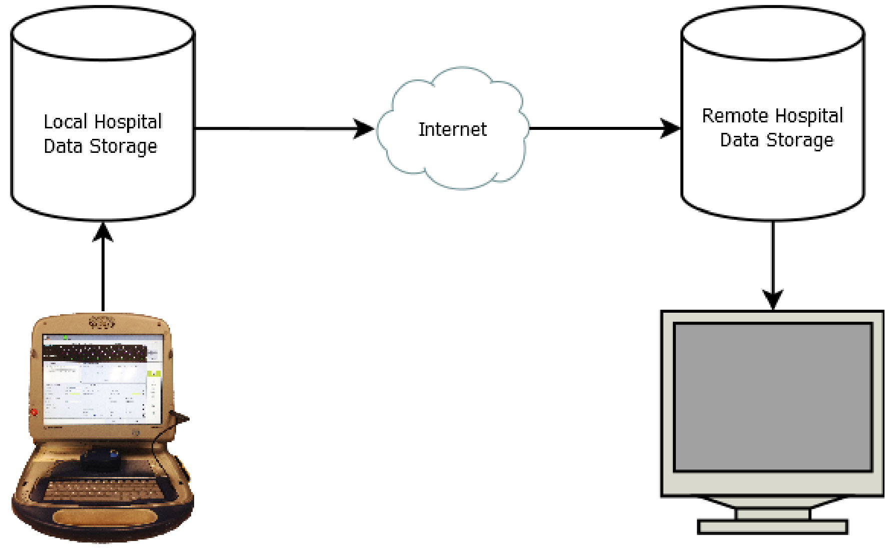

Figure 3: Flowchart of Store-and-Forward

1.2.1.2 Remote-Monitoring System

Remote monitoring is an improvement to store-and-forward due to the analysis of patient's information in real-time. Medtronic has introduced device's that are capable of wirelessly communicating with a pacemaker and transmitting the information to a cardiologist's office in real-time [10]. This CareLink Home Monitor provides a major step in healthcare, because the patient is able to be monitored from their home. This is a large improvement over store-and-forward due to the real-time nature of the system and the initial reduction of travel required by the patient. Figure 4 shows a graphical representation of Remote-Monitoring. Information from the interrogation is transmitted in real-time to an external location for analysis by a cardiologist.

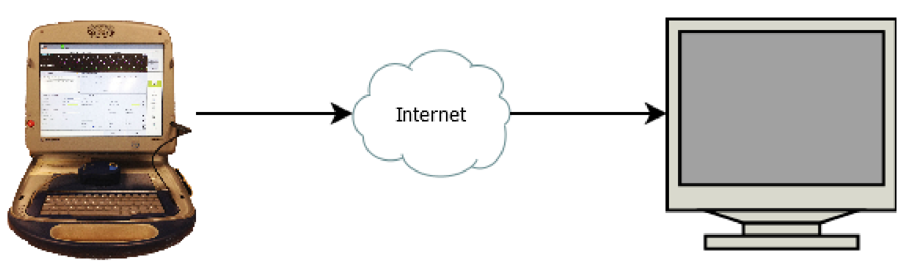

Figure 4: Flowchart of Remote Monitoring

All of telemedicine currently refers to either store-and-forward or remote monitoring of patients with no ability to make adjustments. The remote monitoring is performed in the home of the patient and the remote monitoring equipment connects to the hospital. As the number of pacemakers implanted every year increases as well as the distance patients must travel for interrogations, a system to remotely monitor and program pacemaker's is needed.

1.2.1.3 Real-Time Active System

Real-time active telemedicine is the most robust telemedicine system and has not yet been implemented in telecardiology. The system allows the cardiologist to remotely interrogate and program a pacemaker without the patient traveling to the cardiologist's office. Since most adjustments take less than 10 minutes, a cardiologist is able to make minor adjustments to the pacemaker at a much more frequent interval. The remote programmer is configured at the patient's primary care physician's (PCPs) office. The graphical representation of a real-time active system is shown in Figure 5. This system provides little change to the workflow of the cardiologist while allowing remote diagnosis and adjustments to be made to the pacemaker.

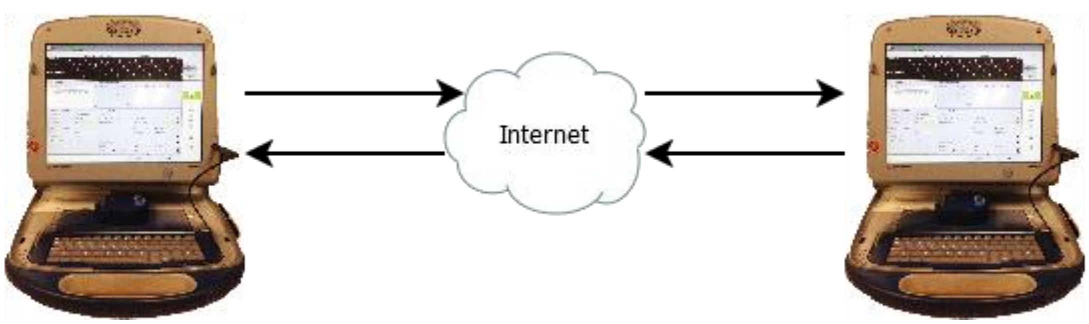

Figure 5: Flowchart of Real-Time Active

2.0 PROBLEM STATEMENT

The real-time active telecardiology system employed in this thesis allows the cardiologist to remotely interrogate a pacemaker with the aid of a nurse. The patient will be located at their PCP's office. The system must be able to program the pacemaker in real-time and must be scalable to multiple locations. The following sections address these main goals and other requirements that affect the design and implementation of a real-time active telecardiology system.

2.1 GOALS AND REQUIREMENTS

The real-time active telemedicine system presented in this thesis was designed with a set of requirements in mind. The following subsections detail the requirements which the real-time active telemedicine system in this thesis was designed to conform to.

2.1.1 Real-time Implementation of Current System

The first requirement of the telemedicine system is the ability to interrogate and program a pacemaker in real-time with minimal changes to the current workflow of the cardiologist. This requirement was set to minimize the training needed during implementation of the system. The interrogation and programming device that Medtronic developed is the Medtronic CareLink Programmer 2090. The cardiologist should be able to turn on the programmer and make adjustments to a pacemaker, as though the patient was in the office. Travel is still required by the patient but this system minimizes the travel time by placing the programmer at the patient's PCP.

2.1.2 Scalability to Multiple Locations

The telemedicine system must be expandable to multiple locations to maximize the care given to the patients and reduce the travel time. Because each programmer can only communicate with its own brand's pacemakers, there is limitation to the expandability of this system. Because the average cost a pacemaker is approximately $30,000, Medtronic provides these CareLink 2090 Programmers to cardiologist's free of charge in exchange for use of their pacemakers [15]. The device described here is an extremely low cost solution that can be scalable to multiple locations with little burden placed on the hospital.

2.1.3 Security and Redundancy of the System

Because patient information is being transferred over a network, security of the transmission is paramount to the wide acceptance of the telecardiology system. In the system presented in this thesis, data transmitted between the host and remote does not contain patient information. The information transferred between the host and remote programmers are the touchscreen data and video data. Because the internal components of the programmer are all commercial-off-the-shelf (COTS) parts, securing the data before transmission is required. Information could be intercepted and decoded with minimal effort if no security is employed. A level of redundancy is also required to ensure that information has not been corrupted during transmission due to noise inherent to the system. This noise would be manifested as incorrect touches and scrambled video.

3.0 THEORY AND OPERATION

The design problems and limitations of a real-time active telemedicine can be understood by examining the governing theory of telecommunications and touchscreens. The following subsections will examine the advantages and disadvantages of capacitive and resistive touchscreens and the basics of telecommunications. The theoretical topics in this section cover the main points needed to understand the design decisions and evolution of the real-time active telemedicine system presented in this thesis.

3.1 TOUCHSCREEN THEORY

Two widely used touchscreens in mass production are capacitive and resistive touch. Resistive touch is a technology that determines the approximate position of a touch by calculating the equivalent resistance between the x-axis and y-axis [16]. This technology is simple and inexpensive to produce but has many limitations that will be discussed. Projected capacitive touchscreens employ the electrical conductivity of the human body to determine the location of a touch. It is broken down into two categories: mutual capacitance and self-capacitance [16]. Both resistive and capacitive touchscreens have complex controllers that determine the location of a touch. Because of the complexity and variety of these controllers, only the basic theory of resistive and capacitive touchscreens will be covered in the following sections.

3.1.1 Resistive Touch

Resistive touchscreens are a type of mechanical sensor that operates by pressure applied by either a finger or stylus. Electrodes are connected to the sides of a thin sheet that contains a transparent resistive coating. When voltage is applied to the electrodes, a resistive network is formed and a touch dependent location can be determined from the calculated resistance. When two of these thin sheets are placed in close proximity and electrodes aligned at 90-degree angles, a resistive touchscreen is created .When pressure is applied to the glass sheets such that contact is made between them, the X and Y locations can be determined by comparing the voltages between the electrodes [16]. An X position is calculated first and then a corresponding Y location is calculated by flipping the switches connected to the electrodes. X and Y locations are measured on the order of milliseconds. This configuration is known as a 4-wire resistive touchscreen and is shown in Figure 6.

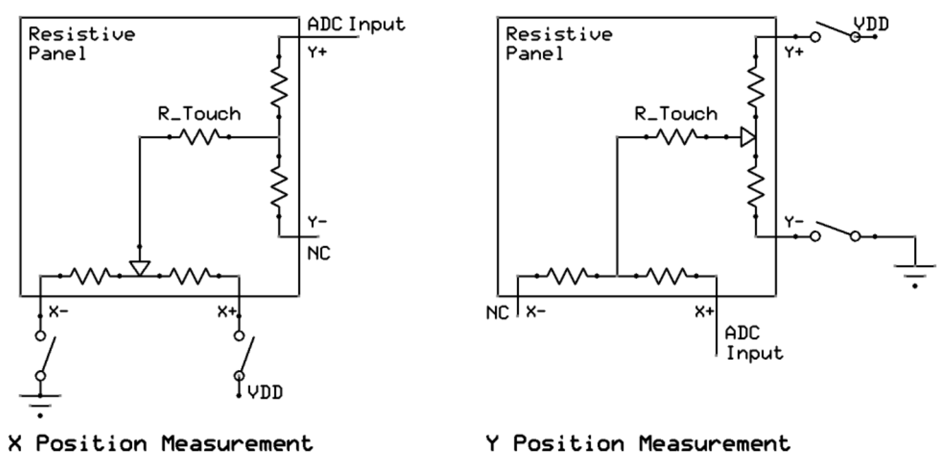

Figure 6: Resistive Touchscreen Diagram

Although inexpensive and easy to interface, the 4-wire resistive touchscreen has many limitations. Because the location of a touch is determined by comparing outputs of a resistive network, only one touch on the screen can by recognized by the controller. The requirement that an air gap exist between the two sheets is vital to the operation of a resistive touchscreen. Excessive pressure over time can cause permanent contact between the glass sheets rendering the touchscreen useless. The possibility of mechanical failure makes resistive touchscreens not viable for mission critical applications.

3.1.2 Projected Capacitive Touch

Projected capacitive touch is a type of capacitive technology that utilizes grids of transparent conductor paths made of indium tin oxide (ITO) [16]. With alternating voltages applied to the X-Y grid of ITO, a charge is induced on the capacitors formed by the ITO diamonds which produces a corresponding electric field. When a conducting object is placed in close proximity of the capacitive touchscreen, the electric field is altered and the internal capacitance of the touchscreen begins to change due to the conduction [18]. If a non-conducting object is brought in close proximity of the touchscreen, there will be no change in the electric field and no touch will be sensed. Because of this, capacitive touchscreens are more robust in application compared to resistive touch but are more prone to external electrostatic interference. Capacitive touchscreens provide greater protection to the liquid crystal display (LCD) due to the glass overlay. The following sections discuss the theory of the two major types of capacitive touchscreens.

3.1.2.1 Self-Capacitance Projected Touch

Self-capacitance implements a diamond pattern of ITO paths, which are etched onto one side of a glass substrate [17]. An alternative voltage is then applied to the rows and columns that induce an electric field. When a conducting object is brought near the touchscreen, the electric field is absorbed and charge redistributes around the touch location which is measured by the touchscreen controller. Figure 7 shows the diamond pattern of a self-capacitance projected touchscreen.

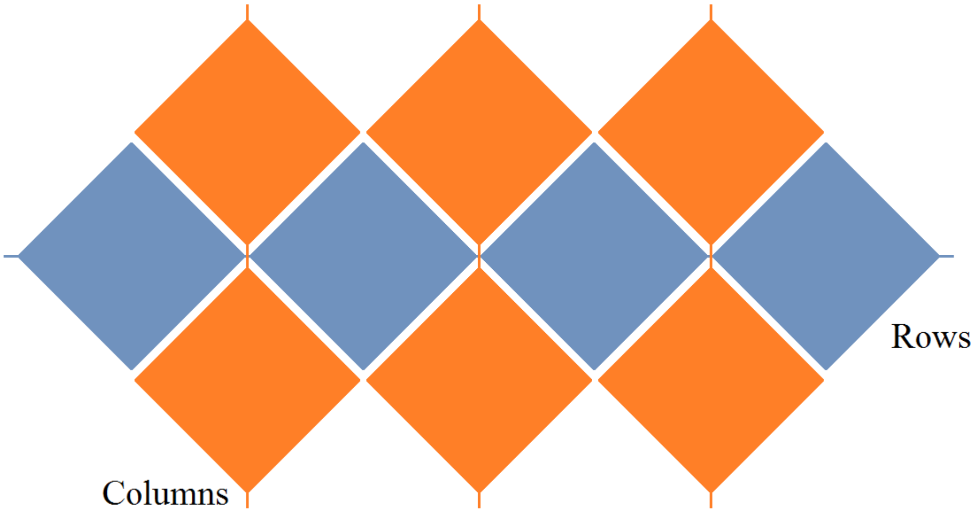

Figure 7: Self-Capacitance ITO Diamond Conductor Paths

This change in electric field measured is converted into a location via interpolation [18]. Interpolation is the mathematical process by which new data points are constructed within the known set of data points [18]. Only single touches can be registered by the touchscreen controller due to limitations of the single layer design.

3.1.2.2 Mutual Capacitance Projected Touch

Mutual capacitance projected touchscreens are more complex and expensive then the self-capacitance touchscreen described in the previous section. Instead of using a single glass substrate, the mutual capacitance system has two layers of glass substrate with vertical and horizontal ITO traces, forming a matrix of capacitors that use the substrate as an insulator [16] [17]. The controller applies an AC voltage to the top layer of driving traces and uses the lower sensing traces to identify a change in capacitance. A conducting object has an internal capacitance and when a touch occurs, the equivalent capacitance seen by touchscreen controller will increase. Because the controller monitors each horizontal and vertical line separately, a multi-touch system can be produced. Figure 8 shows the grid of a mutual capacitance projected touchscreen and the connections to the touchscreen controller.

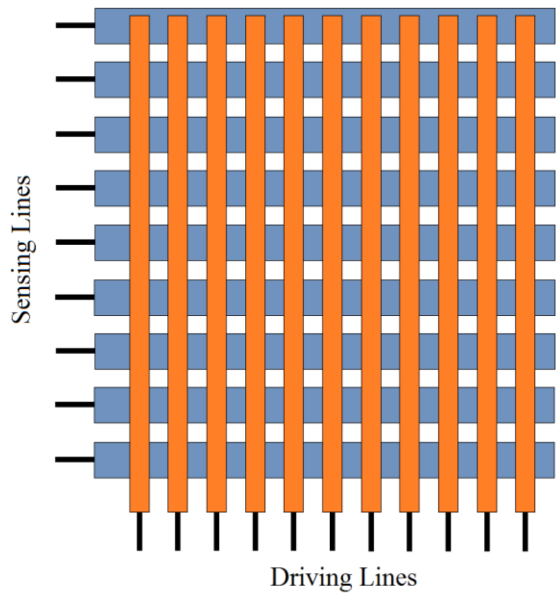

Figure 8: Self-Capacitance Driving and Sensing Lines

3.2 TELECOMMUNICATIONS

The system employed in this thesis relies heavily on the ability to receive and transmit data securely over a variety of transmission protocols. Understanding the underlying theory of serial data transfers, security, and redundancy will aid in the design process.

3.2.1 Universal Synchronous/Asynchronous Receiver Transmitter (USART)

A USART is protocol that receives bytes of data and transmits a bit stream of data at standard TTL voltages [19]. Current microcontroller units (MCU) being produced in the past decade have incorporated an onboard USART for communication between other peripherals. Data can be transferred in either full-duplex mode, both peripherals transmitting at the same time or half-duplex mode, peripherals taking turns transmitting data. The protocol can either be an asynchronous system, referred to as a UART, or a synchronous system, referred to as a USART. Both USARTs and UARTs typically have 4 communication lines: Receive (Rx), Transmit (Tx), Clear to Send (CTS), and Request to Send (RTS). The RTS and CTS control lines are used for flow control to signal that a transmission is about to occur and allow for higher data transmission rates.

When data are transmitted over a USART, no start or stop bits are used and the clock is extracted from data stream. The removal of the start and stop bits means a higher throughput during data transmission but at the cost of complex circuitry. In order to maintain synchronization, the transmitter must send “pad” characters to the receiver when the transmitter is idle [19]. When data are transmitted on a UART, they are framed into packets, which contain one start bit, 8 or 9 data bits, an optional parity bit, and either one or two stop bits. The start and stop bits are always opposite logic values to guarantee that there are at least two separate signal changes between frames. The most commonly used configuration for data transfer is 8 data bits with no parity bit and 1 stop bit, which is referred to as 8N1 and is shown in Figure 9. Data are typically transmitted least significant bit (LSB) transmissions although many MCUs with built in UARTs will allow for most significant bit (MSB) transmissions.

Figure 9: UART 8-N-1 3-Byte Data

3.2.2 Serial Peripheral Interface Bus (SPI)

The SPI bus is a 4-wire synchronous data link that was created by Motorola and operates in a full duplex mode with a master device and a slave device [20]. The 4-wire bus includes a serial clock (SCLK), a master output/slave input (MOSI), a master input/slave output (MISO), and a slave select (SS). The serial clock is always transmitted from the master unit to the slave units so only a single synchronization needs to be performed as multiple slaves are added to the system. When there is one master and one slave, the SS, an active low pin, is grounded on the slave unit and left unconnected on the master unit. This is done to allow for data to be transferred in half-duplex. The SS is used when the SPI has multiple slaves attached to a single master. It controls data flow between the multiple slaves and the master. The ability to communicate between multiple peripherals on the same bus is one of the major advantages of the SPI over a UART. Figure 10 is a graphical implementation of SPI configuration with a master unit and multiple slave units.

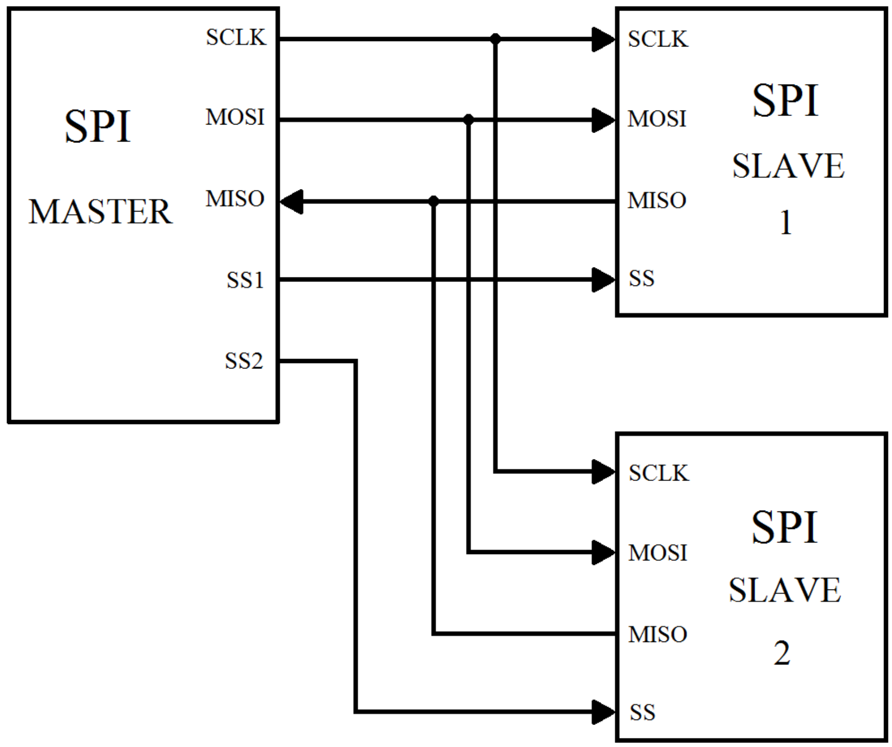

Figure 10: SPI Interface with Multiple Slaves

3.2.3 Interleaving

Interleaving is a telecommunications method of arranging data in a packet as to reduce the effect of noise bursts during transmissions [21]. Because wireless communication systems transmit data over radio frequencies (RF), earlier transmissions can be reflected off surfaces, and couple into current packets of data. These bursts of noise usually affect only a few contiguous bits [21]. This is referred to as multipath interference and can drastically affect the throughput of a wireless network. Interleaving helps reduce the effect of coupled noise bursts into packets but adds latency to the system. The latency arises from the need for the entire packet must be received before it can be decoded. This latency is not considered an issue in the system due to the relatively slow data rate between the touchscreen controller and the programmer. The touchscreen data are copied three times into the same packet before transmission. If data are changed during transmission, the receiver should be able to identify the corrupt data and recreate the original packet.

4.0 SOLUTION EVOLUTION AND ITERATIONS

The real-time active telecardiology system presented in this thesis has gone through several design changes and device iterations. Medtronic outsourced the building of the Medtronic 2090 Programmer to IBM which was released to the public in 2005 [22]. This programmer had a resistive touchscreen with a pen input. The next revision released had a capacitive touchscreen with an updated capacitive touchscreen controller. Both the resistive and capacitive touchscreens have been configured to only accept input from an attached pen. This allows for a level of verification that only touches from the attached pen will be registered by the touchscreen, which is extremely important in a hospital environment.

Each iteration of the telecardiology system in this thesis experimented with transmission interfaces and telecommunications techniques to ensure data integrity while still observing the requirements of the telecardiology system. Because the design requirements remained the same, the following descriptions detail the differences and improvements in the iterations. The results of the Capacitive Touch - Revision II are detailed in the Section 9.0 of the thesis.

4.1 RESISTIVE MEDTRONIC 2090 PROGRAMMER

The Resistive Medtronic 2090 Programmer was the first programmer received from Dr. Schwartzman. Initial interrogation of the touchscreen controller revealed the touchscreen controller was extremely complex. The controller was designed with multiple microcontroller units (MCUs) and complex programmable logic devices (CPLDs). No documentation was found detailing the operation of the resistive controller. A printed circuit board (PCB) was designed to sample data transmitted via the Molex SlimStack connector. The connection provides a communication link between the touchscreen controller and the programmer.

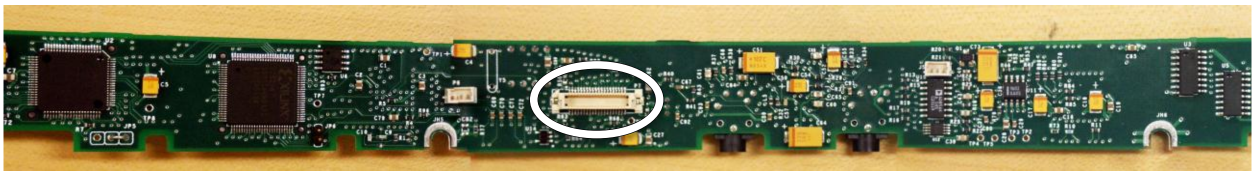

Figure 11: Resistive Touchscreen Controller

Figure 11 shows an image of the resistive touchscreen controller on the Medtronic 2090 Programmer. The Molex SlimStack connector is circled in white. As testing continued, it was discovered that a new revision of the Medtronic 2090 Programmer had been released with a capacitive touchscreen. The resistive touchscreen variant is no longer in production. This led to the decision to abandon the resistive Medtronic 2090 Programmer and focus on the capacitive Medtronic 2090 Programmer.

4.2 CAPACITIVE MEDTRONIC 2090 PROGRAMMER

The capacitive Medtronic 2090 Programmer is the focus of this thesis. The capacitive touchscreen controller has been updated with the 3M MicroTouch EX-II series controller, an application specific integrated circuit (ASIC). This ASIC handles all touch processing and transmits a touch location to the programmer over UART at 9600 baud [23]. The resistive and capacitive controllers are the same size and have the same Molex SlimStack connector. This allows for the same SlimStack PCB to be used for analysis on both the resistive and capacitive variants. Figure 12 shows the capacitive controller PCB with the 3M ASIC circled in white.

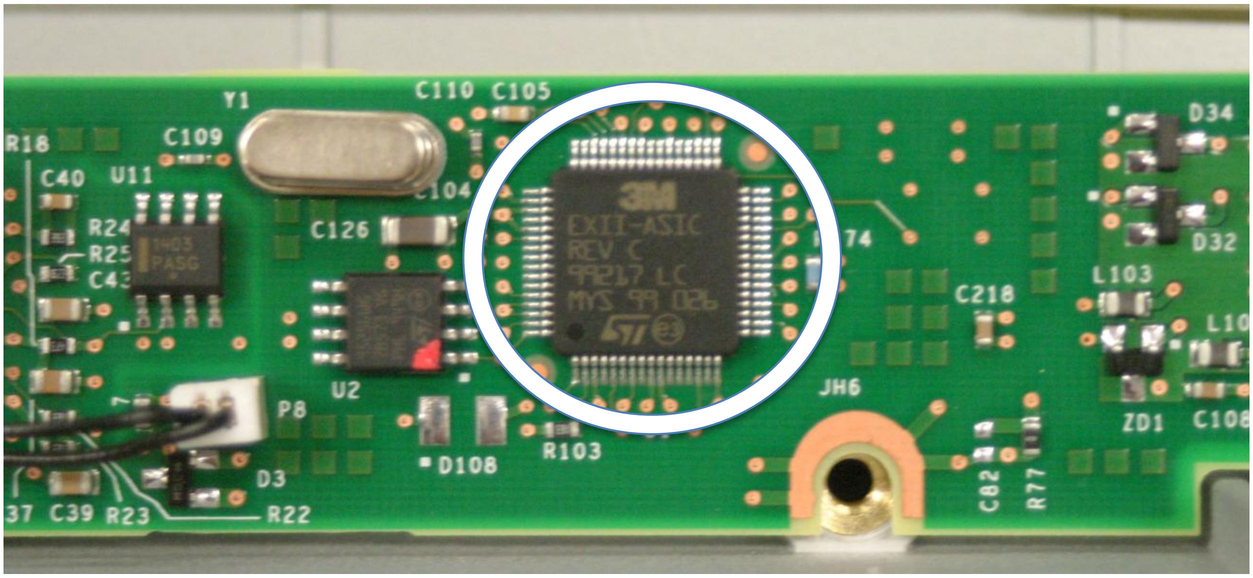

Figure 12: Capacitive Touchscreen ASIC

4.2.1 Revision I – Texas Instruments CC2510

Revision I was a proof-of-concept design attempting to implement a real-time active telemedicine system over a short distance. The system was designed around the Texas Instruments CC2510 mini-development board. The CC2510 is an ASIC that combines an 8051 MCU and a radio frequency (RF) transceiver operating at 2.4GHz [24]. The RF transceiver is capable of transmitting data at 500k baud. The mini-development board met the requirements for a short-range wireless proof-of-concept system while allowing for future expansion due to the widely used 8051 core. Figure 13 shows the proof-of-concept real-time active telecardiology system developed.

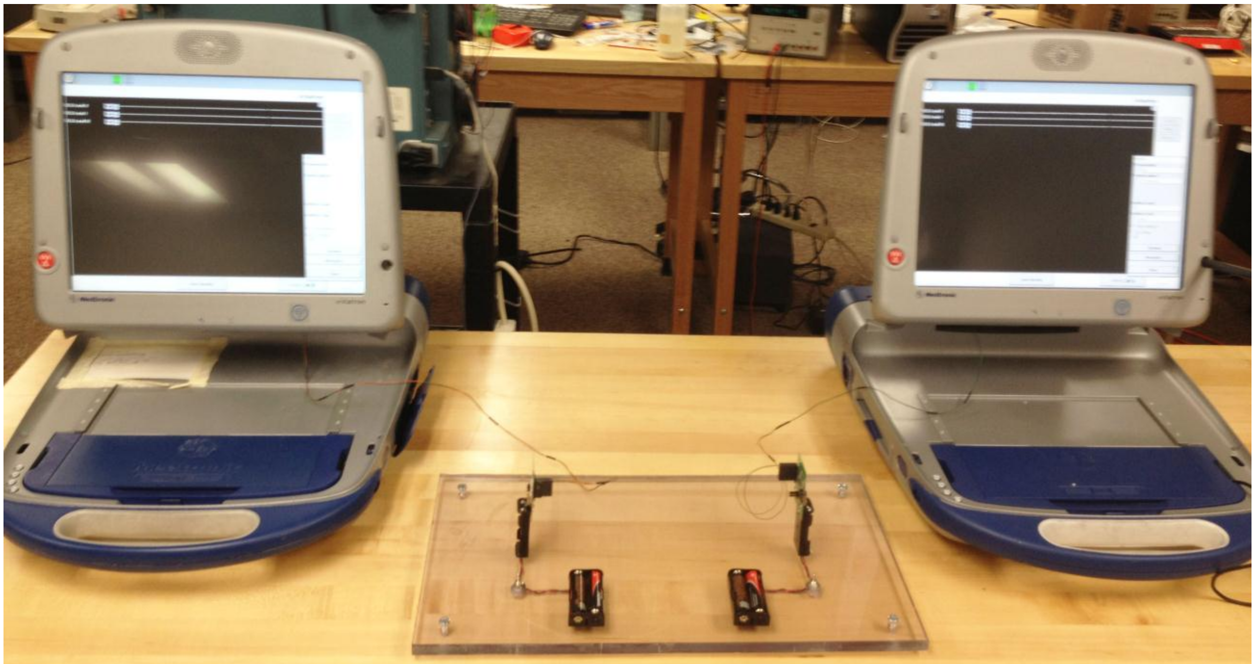

Figure 13: Revision I - TI CC2510

Revision I implemented a proof-of-concept design with a major focus on the real-time aspect of the system. The proof-of-concept was plagued by wireless interference originating from the 802.11b/g wireless routers present in the testing environment. The interference occurred during long transmissions of data when the pen was continuously dragged across the screen.

4.2.2 Revision II – Lantronix MatchPort

Revision II was the final revision of the real-time active telecardiology project, which moved from the proof-of-concept system to implementable system. Revision II replaces the 2.4 GHz transceiver with a Lantronix Serial-to-ethernet controller that allows for communication over a local area network (LAN) or wide area network (WAN). The remaining sections in this thesis explore the detailed design of Revision II – Lantronix MatchPort real-time active telecardiology system.

5.0 SYSTEM SPECIFICATIONS

The system consists of a host Medtronic 2090 Programmer and a remote Medtronic 2090 Programmer. The host unit continuously samples data transmitted between the touchscreen controller and the programmer. The hose unit routes only the touchscreen location data to the remote unit. The remote unit has two functions it must perform. The remote unit acts as a touchscreen controller for the remote Medtronic 2090 Programmer. The remote unit continuously waits for touchscreen location transmissions from the host unit and routes the data to the remote Medtronic 2090 Programmer. Figure 14 shows the high level view and the data flow throughout the telecardiology system.

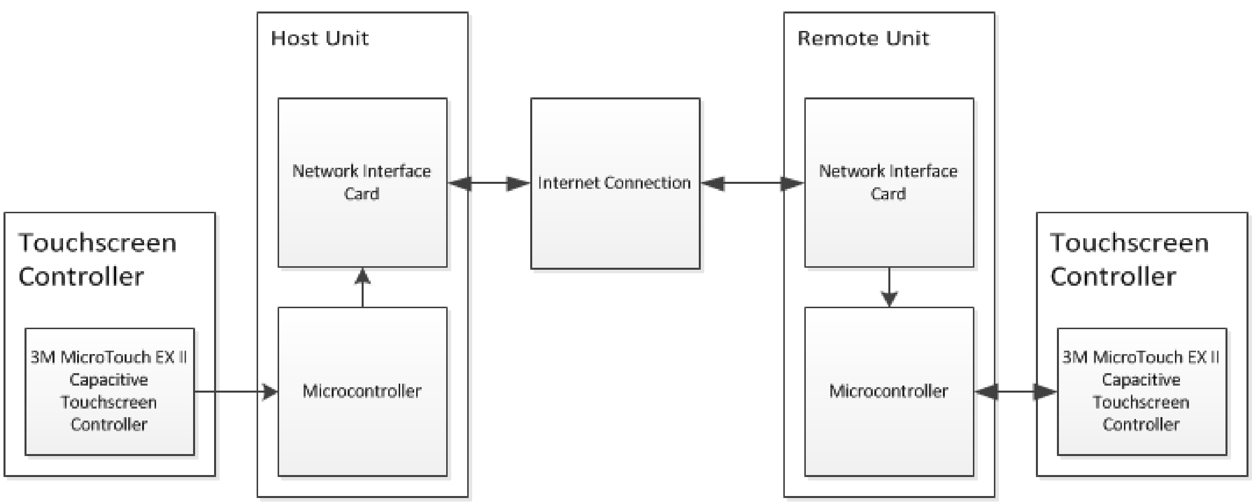

Figure 14: High Level View of Real-Time Active Telemedicine System

5.1 HOST UNIT SPECITIFICATIONS

The following subsections present the specifications of the host unit connected to the Host Medtronic 2090 Programmer in the real-time active telecardiology system. The subsections will explore the hardware and software architectures of the host unit at a high level. The host unit samples and filters data transmitted from the touchscreen controller to the programmer. The data are interleaved before transmission as discussed in Section 3.2.3.

5.1.1 Host Unit Hardware Architecture

This subsection details the hardware architecture of the host unit used in the real-time active telecardiology system. The hardware architecture is presented as a high-level view of the host unit shown in Figure 15.

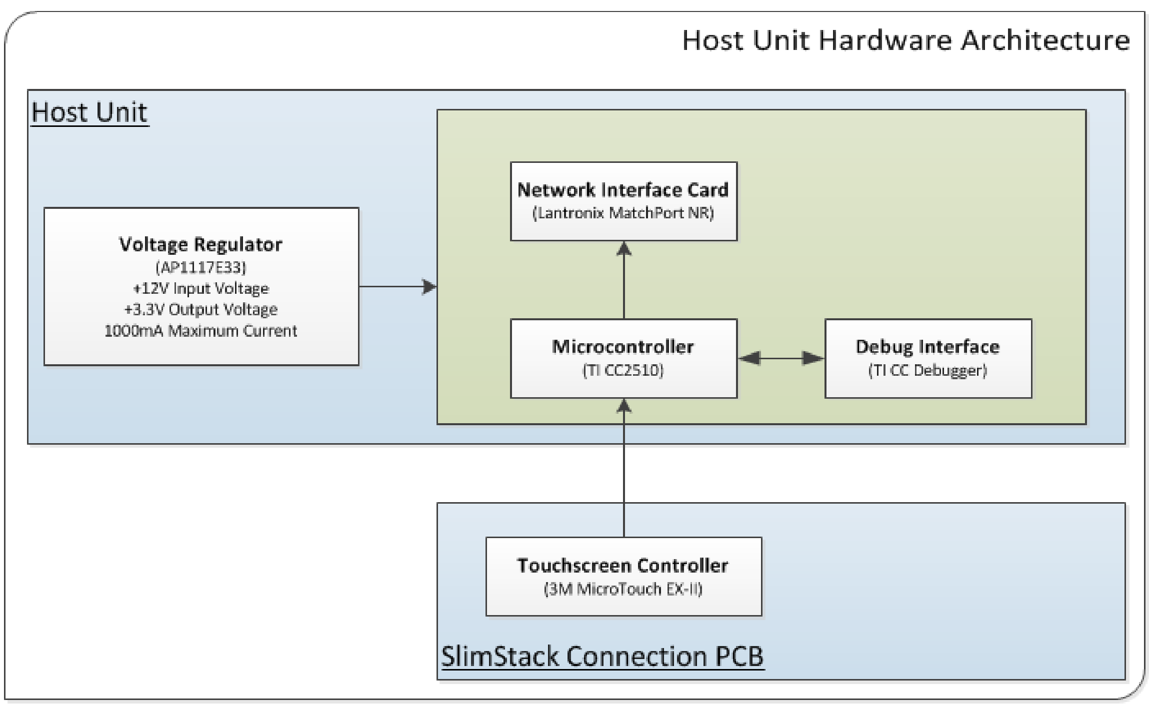

Figure 15: Host Unit Hardware Architecture

The host unit's MCU filters and routes data that is transmitted between then touchscreen controller and the programmer. The debug interface is required for both programming and debugging of the MCU. It allows code to be stepped through to ensure general execution of the code and external interrupts are occurring properly. The voltage regulator provides a constant voltage supply to the MCU and the Serial-to-ethernet controller. Because the Medtronic 2090 Programmer requires a standard wall outlet of 120V at 60Hz, the host unit was designed to operate from an AC to DC power supply. The touchscreen PCB connects the data lines between the touchscreen controller and the host unit's MCU. The serial-to-ethernet interface provides network connectivity to the host unit and interfaces directly with the MCU.

5.1.2 Host Unit Software Architecture

The host unit software architecture subsection details a high-level view of the software programmed onto the MCU through the debug interface. The software controlling the host unit initially waits for the Medtronic 2090 Programmer to be turned on before configuring the UART and DMA registers. Once the UARTs and DMA channels are properly configured, the MCU filters startup data transmitted from the touchscreen controller. When touchscreen location data are detected, the MCU packets the data and transmits it to the serial-to-ethernet controller. Figure 16 illustrates the host unit's software architecture from a high-level.

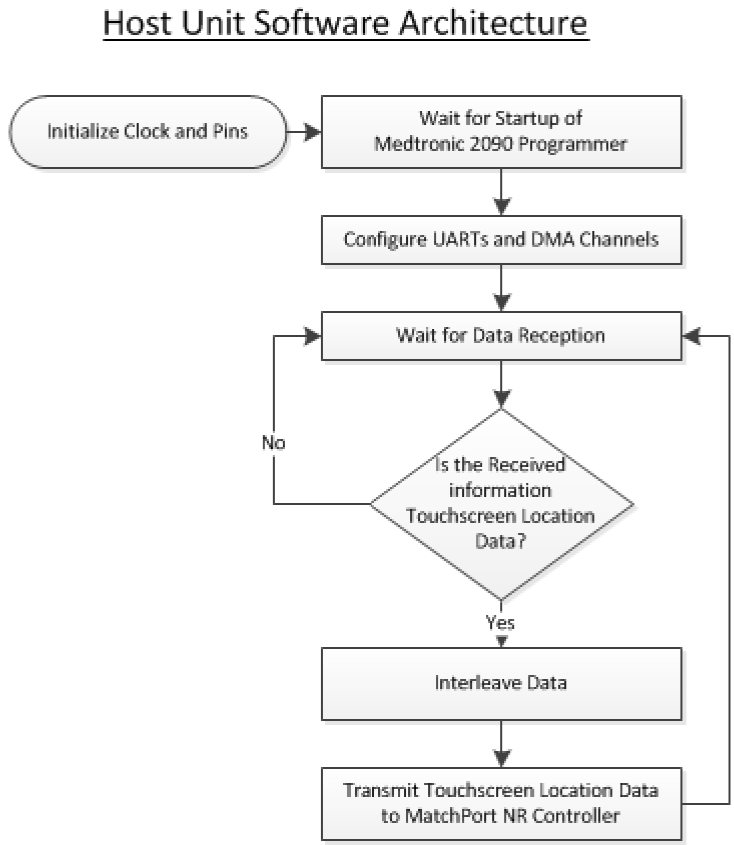

Figure 16: Host Unit Software Architecture

5.2 REMOTE UNIT SPECITIFICATIONS

The following subsections present the specifications of the remote unit connected to the Remote Medtronic 2090 Programmer in the real-time active telecardiology system. These subsections will explore the hardware and software architectures of the remote unit at a high level. The remote unit must imitate the touchscreen controller during startup while simultaneously waiting for a data transmission from the serial-to-ethernet controller. Once data are received, it is processed and transmitted to the programmer.

5.2.1 Remote Unit Hardware Architecture

This subsection details the hardware architecture of the remote unit used in the real-time active telecardiology system. The hardware architecture presented is a high-level view of the remote unit as seen in Figure 17.

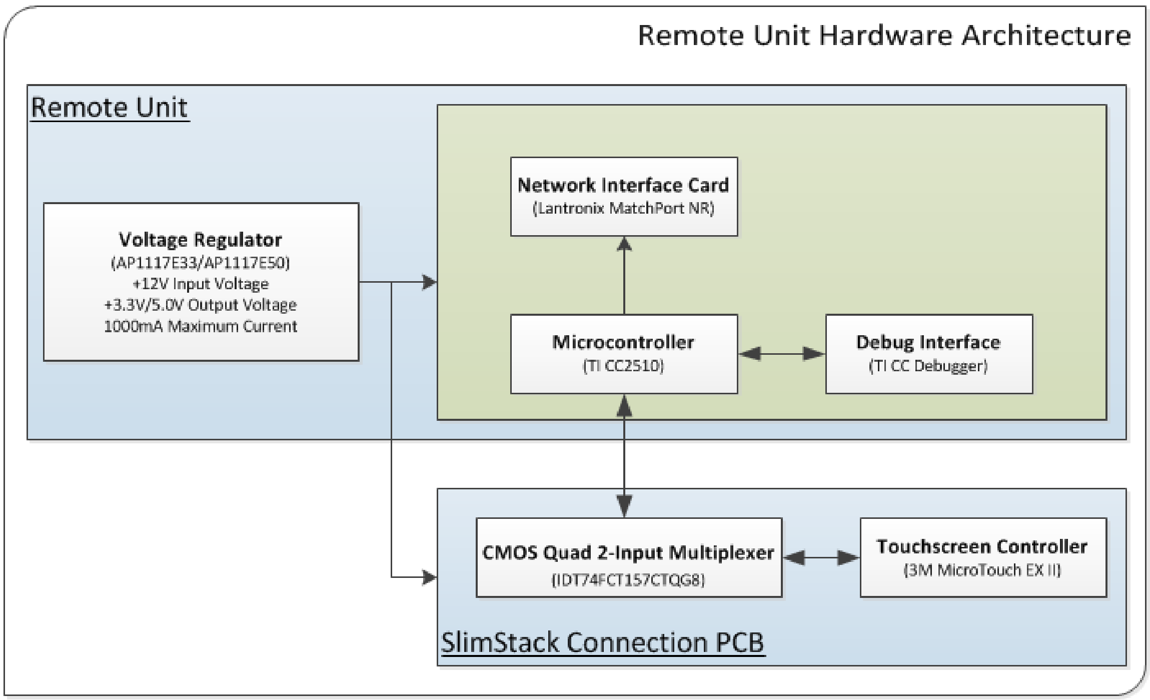

Figure 17: Remote Unit Hardware Architecture

The remote unit's MCU acts as a pseudo touchscreen controller to the programmer and waits to receive data from the serial-to-ethernet controller. The debug interface is required for both programming and debugging of the MCU. It allows code to be stepped through to ensure proper execution of the code and external interrupts. The voltage regulator provides a constant voltage supply to the MCU, the Serial-to-ethernet controller, and the touchscreen PCB. Because the Medtronic 2090 Programmer requires a standard wall outlet of 120V at 60Hz, the remote unit was designed to operate from an AC to DC power supply. The touchscreen PCB routes the flow of data from the touchscreen controller to either the remote unit's MCU or the programmer. This switching allows the remote control of a Medtronic 2090 Programmer to be switched on or off. The serial-to-ethernet interface provides network connectivity to the remote unit and interfaces directly with the MCU.

5.2.2 Remote Unit Software Architecture

The remote unit software architecture subsection details a high-level view of the flow of software programmed onto the MCU through the debug interface. The software controlling the remote unit initially waits for the Medtronic 2090 Programmer to be turned on before configuring the UART and DMA registers. Once the UARTs and DMA channels are properly configured, the MCU acts as a pseudo touchscreen controller responding to the requests of the programmer while waiting for touchscreen location data to be transmitted from the serial-to-ethernet controller. Once data are received from the serial-to-ethernet controller, they are checked for errors before transmission to the programmer. Figure 18 illustrates the remote unit's software architecture from a high-level.

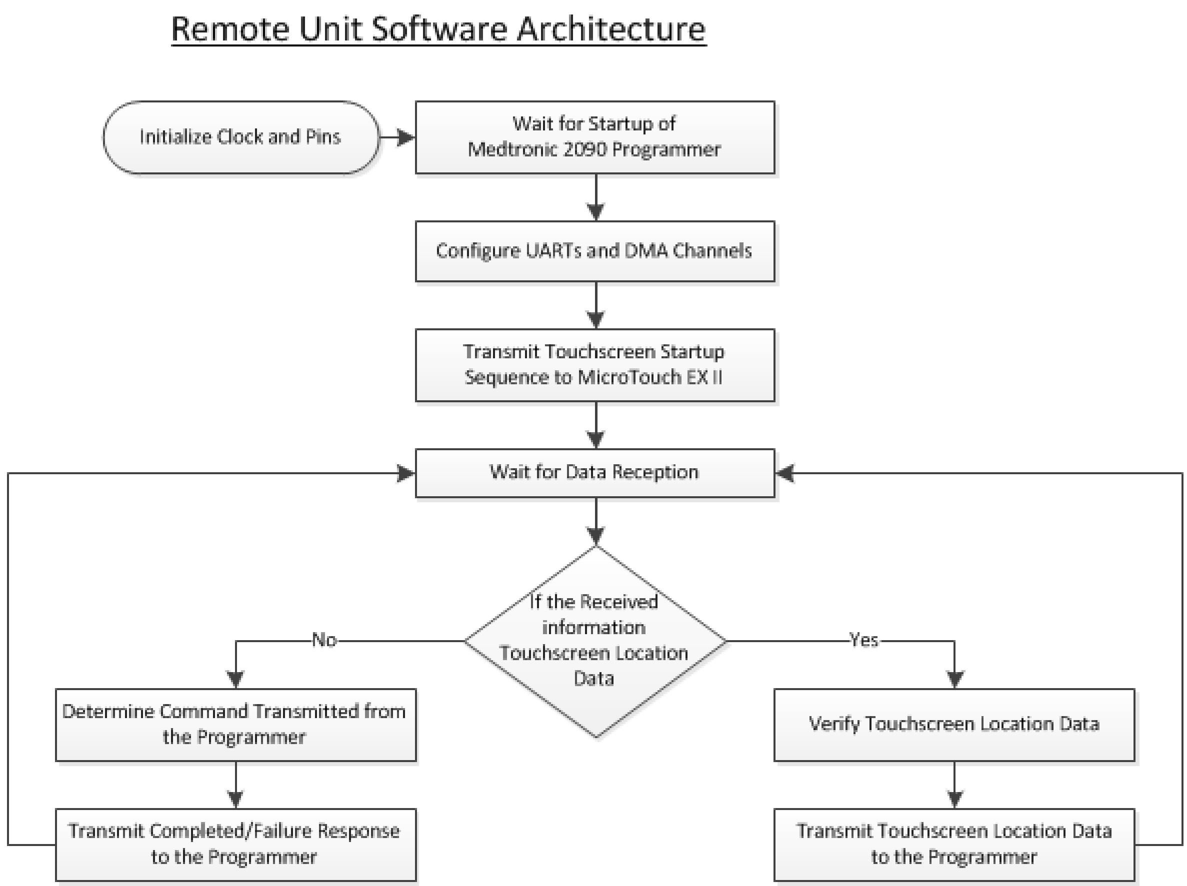

Figure 18: Remote Unit Software Architecture

6.0 HARDWARE DESIGN

The hardware design section of this thesis provides an in-depth analysis of the hardware design of the real-time active telemedicine host and remote units. This section will examine each hardware component of the host and remote unit. The in-depth analysis will include schematic layouts and pin connections.

6.1 HOST UNIT HARDWARE DESIGN

The following subsections perform an in-depth analysis of the host unit hardware and the specific detailed descriptions each component as detailed in Section 5.1.1.

6.1.1 Microcontroller

The microcontroller on the host unit filters, packs and transmits data to the serial-to-ethernet controller. The application specific integrated circuit (ASIC) used was the Texas Instruments CC2510, which incorporates an 8051 microprocessor with a 2.4GHz wireless transceiver [25]. The CC2510 runs on a 16-bit architecture with 4KB of RAM and 32KB of programmable flash memory and uses a 26MHz crystal oscillator. The CC2510 was used during Revision I, when a proof of concept telecardiology system was implemented and is used in

Revision II due to the familiarity of the microcontroller, even though the 2.4GHz transceiver is not being utilized. The CC2510 is programmed and debugged through the Texas Instruments CC Debugger.

Two pins are configured as peripheral pins to allow for UART transmissions at separate baud rates. UART0 is configured to receive data from the touchscreen controller through the direct memory access controller (DMA) at 9600 baud. UART1 is configured to transmit data to through a separate DMA channel to the serial-to-ethernet controller at 460,800 baud. Two pins are configured as GPIO pins to control status LEDs that signal correct startup and data transfers on UART0 and UART1. One 8-bit timer is configured at startup to delay execution of the program. An additional seven pins are reserved for programming and debugging of the CC2510 through the debug interface. Figure 19 shows the schematic of the CC2510 and the pin connections to the other hardware components on the host unit.

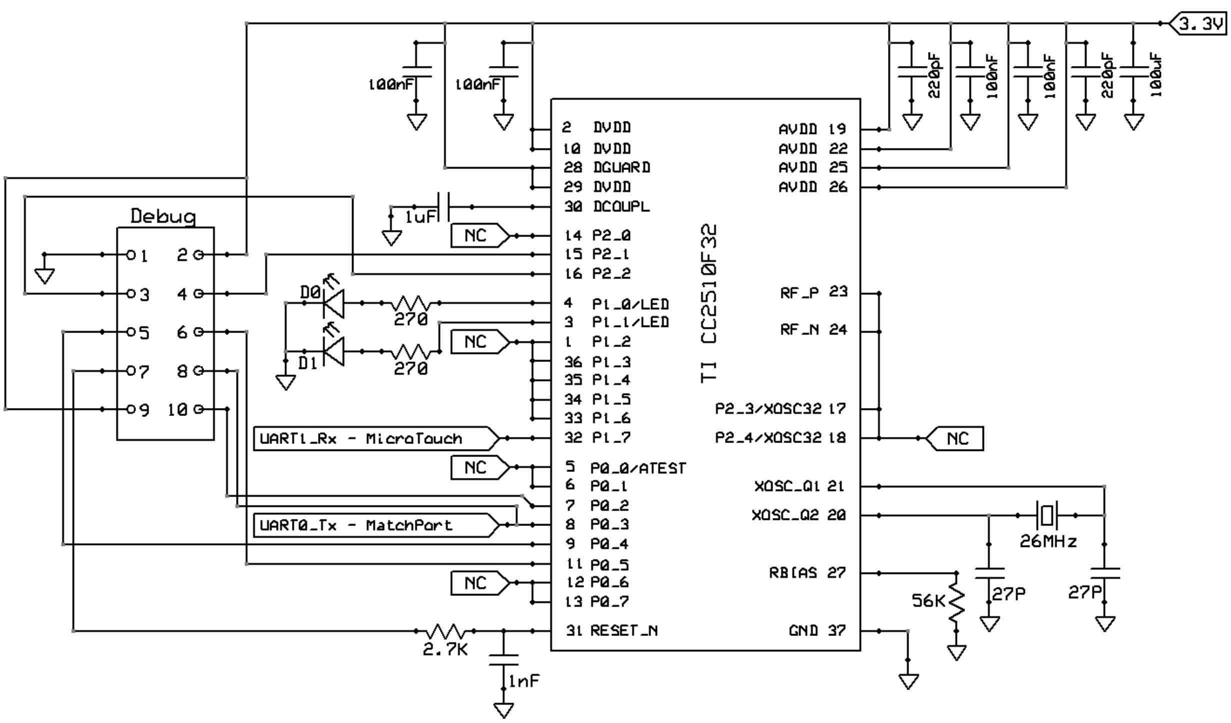

Figure 19: Schematic of CC2510F32 Microcontroller on Host Unit

6.1.2 Network Connectivity

Every host unit must communicate to the remote unit via a network. The Lantronix MatchPort NR serial-to-ethernet controller provides this function on the host unit [26]. The MatchPort NR provides two (2) CMOS 3.3V level UART communications lines that can be configured in a variety of parities, flow control and serial line formats. The maximum serial transmission rate that can be achieved is 921,000 baud.

The MatchPort provides a full TCP/IP stack that can be configured via a web interface or a command line interface. The onboard CPU provides variable levels of AES encryption. The MatchPort NR connects to the hardwired network via a standard RJ-45 connector. The MatchPort b/g provides a serial-to-wireless interface that is pin and software compatible with the MatchPort NR. This allows for future revisions to have a variety of connections with few changes to the current configuration.

The MatchPort NR comes in a standard 40-pin through-hole package and runs on a +3.3V supply. There are 8 GPIO pins that control status LEDs and require LED drivers because the MatchPort pins cannot source enough current to the LED. The UART interface connects to the microcontroller through one data line and a common ground. The link and activity LEDs require the Fairchild Semiconductor MMBT3906 BJT to drive them. The UART interface connects to the microcontroller through one data line and a common ground. Figure 20 shows the schematic of the Lantronix MatchPort NR serial-to-ethernet controller and the functionality of each pin on the host unit. Table 1 elaborates on each pin and provides an explanation of the pin interfaces from the MatchPort NR controller to the host unit.

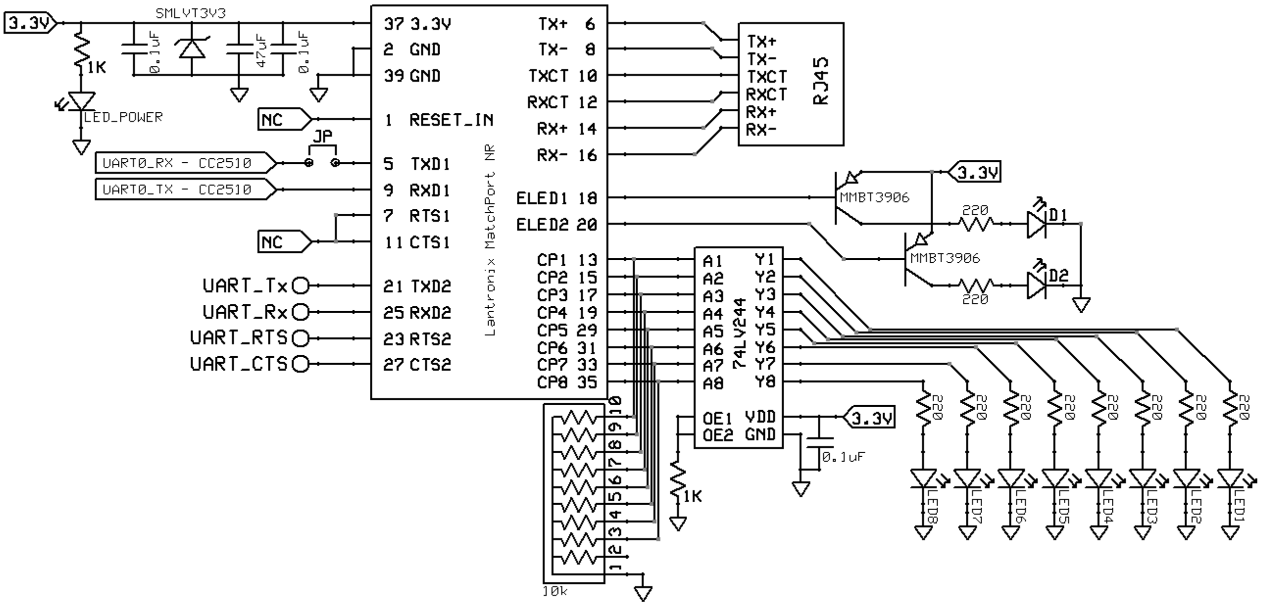

Figure 20: Schematic of Host Lantronix MatchPort NR Interface

Table 1: Elaboration of Pins on MatchPort NR on Host Unit

Pin #	Name	Function
1	Reset In#	Left Floating as no reset of controller is required
5	TXD1	Transmit Data Port 0 – Not used but connected in schematic
7	RTS1	Request to Send Port 0 – Not connected as Flow Control isn't used
9	RXD1	Receive Data Port 0 – Connected to UART1 Tx on TI CC2510
11	CTS1	Clear to Send Port 0 – Not connected as Flow Control isn't used
13	CP1	Configurable Pin – Not used but connected in schematic
15	CP2	Configurable Pin – Not used but connected in schematic
17	CP3	Configurable Pin – Configured as status LED
19	CP4	Configurable Pin – Configured as status LED
21	TXD2	Transmit Data Port 1 – Not used but left to program controller
23	RTS2	Request to Send Port 1 – Not used but left to program controller
25	RXD2	Receive Data Port 1 – Not used but left to program controller
27	CTS2	Clear to Send Port 1 – Not used but left to program controller
29	CP5	Configurable Pin – Configured as status LED
31	CP6	Configurable Pin – Configured as status LED
33	CP7	Configurable Pin – Configured as status LED

Table 1 (continued).

35	CP8	Configurable Pin – Configured as status LED
37	3.3V	3.3V Power Input – Requires bypass capacitors as close to pin
39	GND	Ground
2	GND	Ground
6	TX+	Ethernet TX+
8	TX-	Ethernet TX-
10	TXCT	Ethernet TX Center Tap
12	RXCT	Ethernet RX Center Tap
14	RX+	Ethernet RX+
16	RX-	Ethernet RX-
18	ELED1	Ethernet Link LED – Requires LED Driver
20	ELED2	Ethernet Activity LED – Requires LED Driver

6.1.3 Power Considerations and Voltage Regulation

Because the Medtronic 2090 Programmer requires a 120V 60Hz AC supply voltage, the host unit was designed to run on the same 120V 60Hz AC supply voltage. A wall outlet transformer from V-Infinity was chosen to rectify the AC signal. The EPSA120100U rectifies the 120V at 60Hz AC to a 12V DC voltage and can source up to 1A of current [27]. Both the CC2510 and the MatchPort NR require 3.3V. The Diodes Incorporated AP1117E33G regulates a 12V DC down to 3.3V DC and can source a maximum of 1A [28]. The AP1117E33G is housed in a standard SOT-223 surface mount package to enable a compact design. Figure 21 shows the schematic of the voltage regulator circuitry.

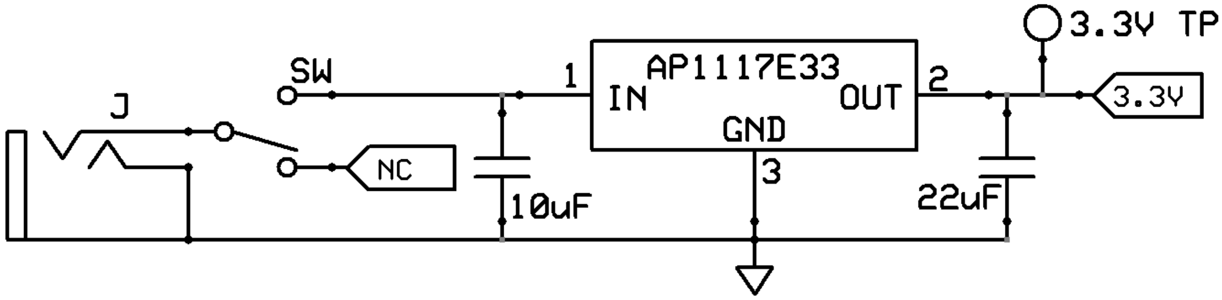

Figure 21: Schematic of Voltage Regulation Circuitry on Host Unit

6.1.4 Touchscreen Connection PCB

The touchscreen connection board was designed to sit in between the touchscreen controller and the programmer without disrupting data flow. The connection allows the microcontroller to sample touchscreen location data that are being transmitted. The touchscreen connection PCB has a male and female 40-pin Molex SlimStack connector on a 2-layer PCB [29]. Figure 22 shows the schematic of the touchscreen connection PCB and the functionality of each pin.

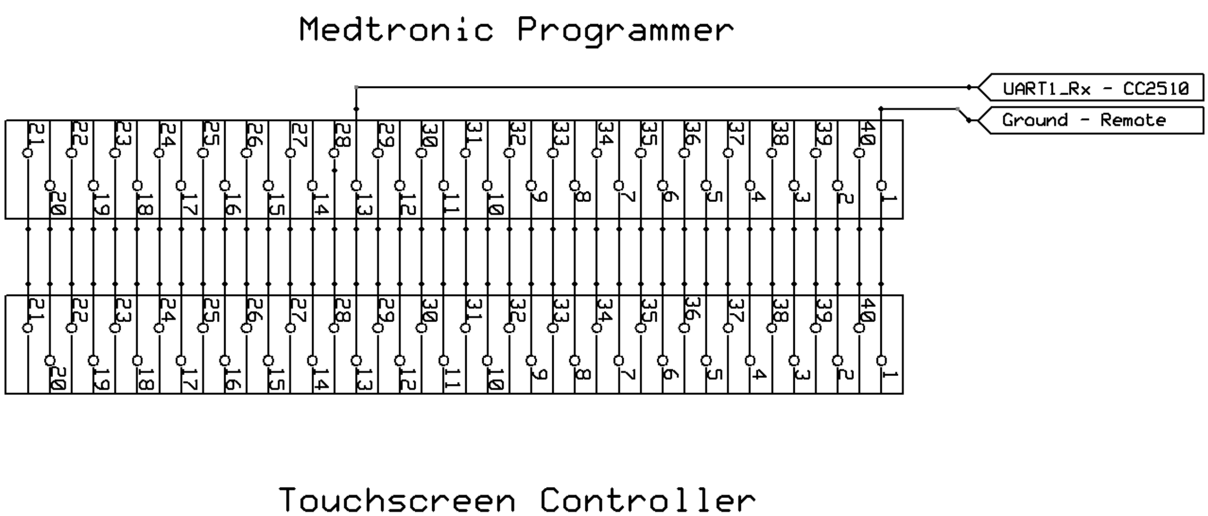

Figure 22: Schematic of Touchscreen Connection PCB with SlimStack Connector

6.2 REMOTE UNIT HARDWARE DESIGN

The following subsections perform an in-depth analysis of the remote unit hardware and the specific detailed descriptions each component as detailed in Section 5.2.1.

6.2.1 Microcontroller

The microcontroller on the remote unit has two main functions. It must be able to act as a pseudo touchscreen controller for the remote programmer during startup. The MCU waits for touchscreen location data from the ethernet controller. The application specific integrated circuit (ASIC) used was the Texas Instruments CC2510, which incorporates an 8051 microprocessor with a 2.4GHz wireless transceiver [25]. The TI CC2510 is the same microprocessor implemented on the host unit and described in Section 6.1.1.

Three pins are configured as peripheral pins for UART transmissions at separate baud rates. UART0 is configured to transmit and receive data from the touchscreen controller through separate channels of the direct memory access controller (DMA) at 9600 baud. UART1 is configured to transmit data through a third DMA channel to the serial-to-ethernet controller at 460,800 baud. Two pins are configured as GPIO pins to control status LEDs that are used to signal correct startup and data transfers. One 8-bit timer is configured at startup to delay execution of the program. An additional seven pins are reserved for programming and debugging of the CC2510 through the debug interface. Figure 23 shows the schematic of the CC2510 and the pin connections to other hardware components on the remote unit.

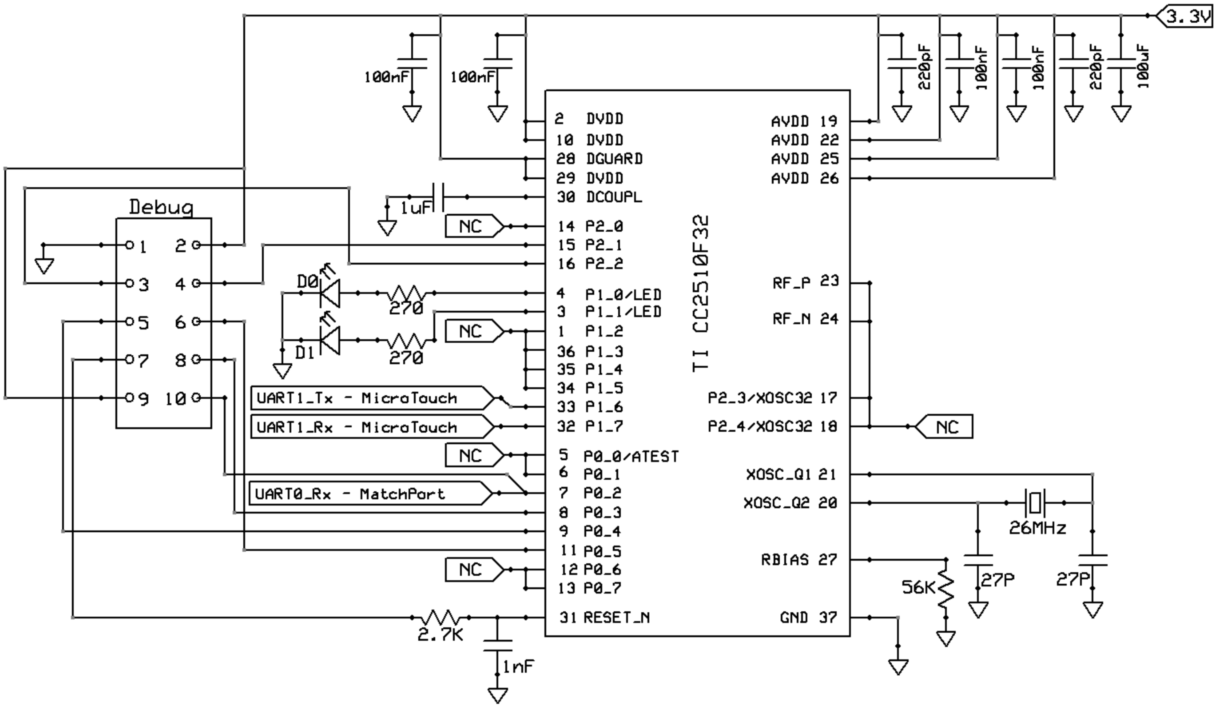

Figure 23: Schematic of CC2510F32 Microcontroller on Remote Unit

6.2.2 Network Connectivity

Each remote unit must wait for transmissions from the host unit via the network interface. The Lantronix MatchPort NR serial-to-ethernet controller provides this function on the remote unit [26]. The Lantronix MatchPort NR is the same serial-to-ethernet device implemented on the host unit and described in Section 6.1.2. A jumper was added between the UART0 Rx pin to allow for future revisions and debug purposes. Figure 24 shows the schematic of the Lantronix MatchPort NR serial-to-ethernet controller and the functionality of each pin on the remote unit.

Table 2 elaborates on each pin and provides an explanation of the pin interfaces from the MatchPort NR controller to the remote unit.

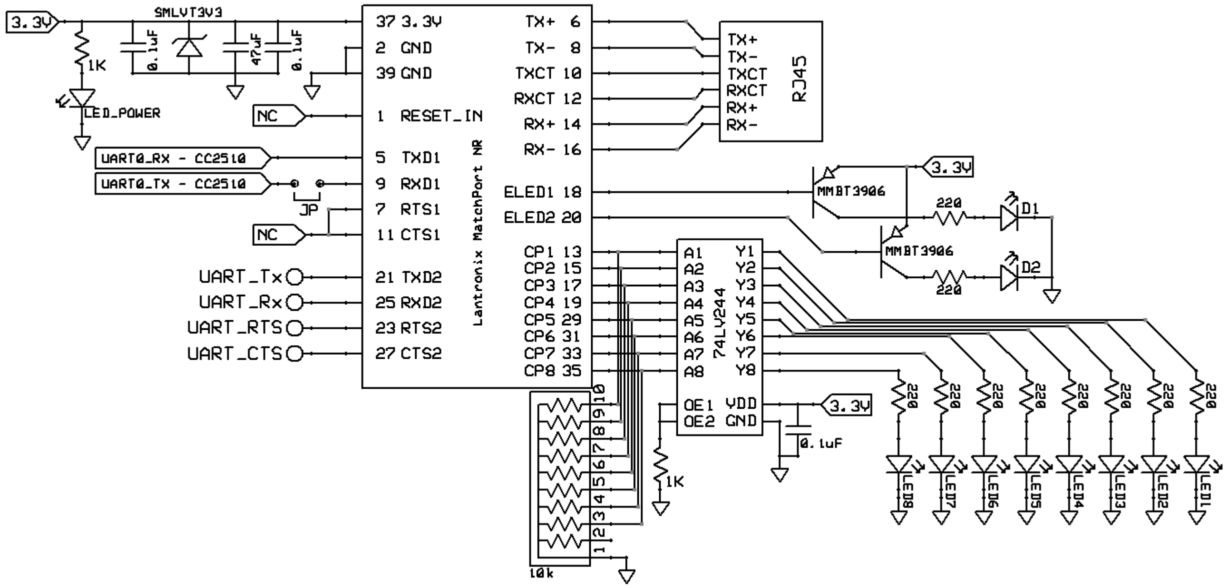

Figure 24: Schematic of Remote Lantronix MatchPort NR Interface

Table 2: Elaboration of Pins on MatchPort NR on Remote Unit

Pin #	Name	Function
1	Reset In#	Left Floating as no reset of controller is required
5	TXD1	Transmit Data – Port 0 – Connected to UART1 Rx on TI CC2510
7	RTS1	Request to Send Port 0 – Not connected as Flow Control isn't used
9	RXD1	Receive Data – Port 0 – Not used but connected in schematic
11	CTS1	Clear to Send Port 0 – Not connected as Flow Control isn't used
13	CP1	Configurable Pin – Not used but connected in schematic
15	CP2	Configurable Pin – Not used but connected in schematic
17	CP3	Configurable Pin – Configured as status LED
19	CP4	Configurable Pin – Configured as status LED
21	TXD2	Transmit Data Port 1 – Not used but left to program controller
23	RTS2	Request to Send Port 1 – Not used but left to program controller
25	RXD2	Receive Data Port 1 – Not used but left to program controller
27	CTS2	Clear to Send Port 1 – Not used but left to program controller

Table 2 (continued).

29	CP5	Configurable Pin – Configured as status LED
31	CP6	Configurable Pin – Configured as status LED
33	CP7	Configurable Pin – Configured as status LED
35	CP8	Configurable Pin – Configured as status LED
37	3.3V	3.3V Power Input – Requires bypass capacitors as close to pin
39	GND	Ground
2	GND	Ground
6	TX+	Ethernet TX+
8	TX-	Ethernet TX-
10	TXCT	Ethernet TX Center Tap
12	RXCT	Ethernet RX Center Tap
14	RX+	Ethernet RX+
16	RX-	Ethernet RX-
18	ELED1	Ethernet Link LED – Requires LED Driver
20	ELED2	Ethernet Activity LED – Requires LED Driver

6.2.3 Power Considerations and Voltage Regulation

Because the Medtronic 2090 Programmer requires a 120V 60Hz AC supply voltage, the remote unit was designed to run on the same 120V 60Hz AC supply voltage. A wall outlet transformer from V-Infinity was chosen to rectify the AC signal. The EPSA120100U rectifies the 120V at 60Hz AC to a 12V DC voltage and can source up to 1A of current [27]. Both the CC2510 and the MatchPort NR require 3.3V. The Diodes Incorporated AP1117E33G regulates a 12V DC down to 3.3V DC and can source a maximum of 1A [28]. The AP1117E33G is housed in a standard SOT-223 surface mount package to enable a compact design. The multiplexer on the touchscreen connection PCB requires a 5V DC supply. A Diodes Incorporated AP1117E50G regulates the 12V DC down to 5V and source a maximum of 1A [28]. The AP1117E50G is housed in a standard SOT-223 surface mount package. Both the AP1117E33G and AP1117E50G are linear voltage regulators.

Capacitors are placed from the input voltage to ground and the output voltage to ground. These are added to decouple noise at the input and output that can produce unreliable IC performance. Figure 25 shows the schematic of the voltage regulator circuitry.

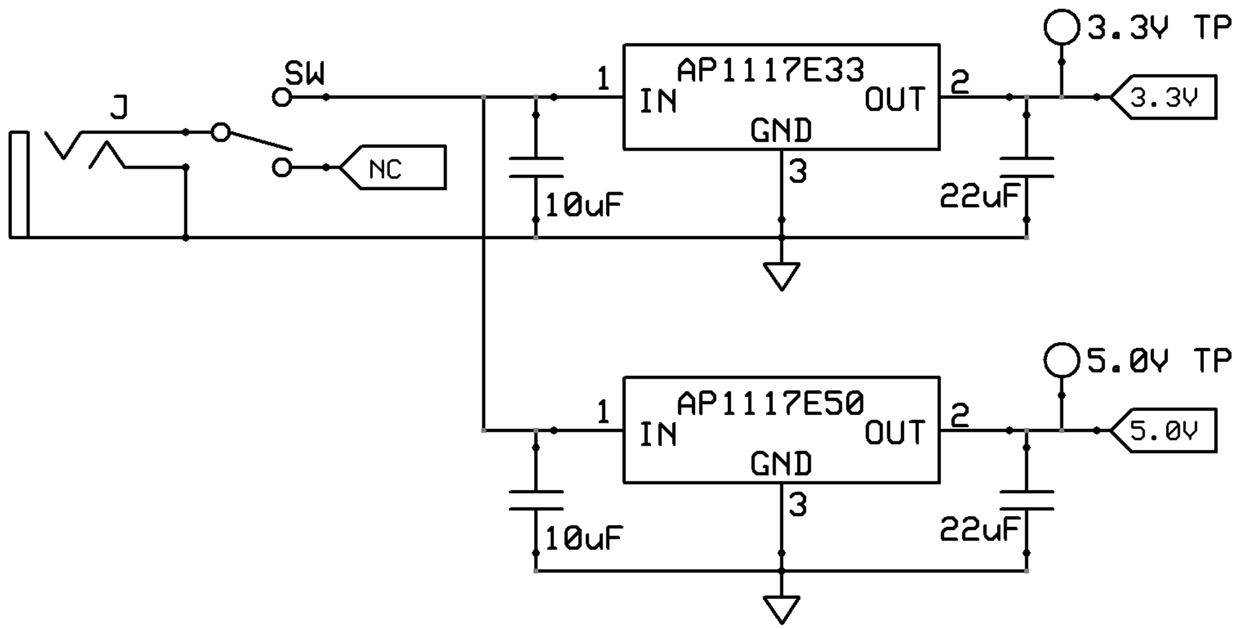

Figure 25: Schematic of Voltage Regulator on Remote Unit

6.2.4 Touchscreen Connection PCB

The touchscreen connection board was designed to sit in between the touchscreen controller and the programmer. The board was designed to control the flow of data to the programmer from either the touchscreen controller or the MCU. In order to control the flow of data, an NXP single 2-input multiplexer was designed onto the PCB [30]. The 74LVC1G157 requires a 5V for power and a enable line. The connection allows the microcontroller to sample touchscreen location data that is being transmitted. The touchscreen connection PCB has a male and female 40-pin Molex SlimStack connector on a 2-layer PCB [29]. Figure 26 shows the schematic of the touchscreen connection PCB and the functionality of each pin.

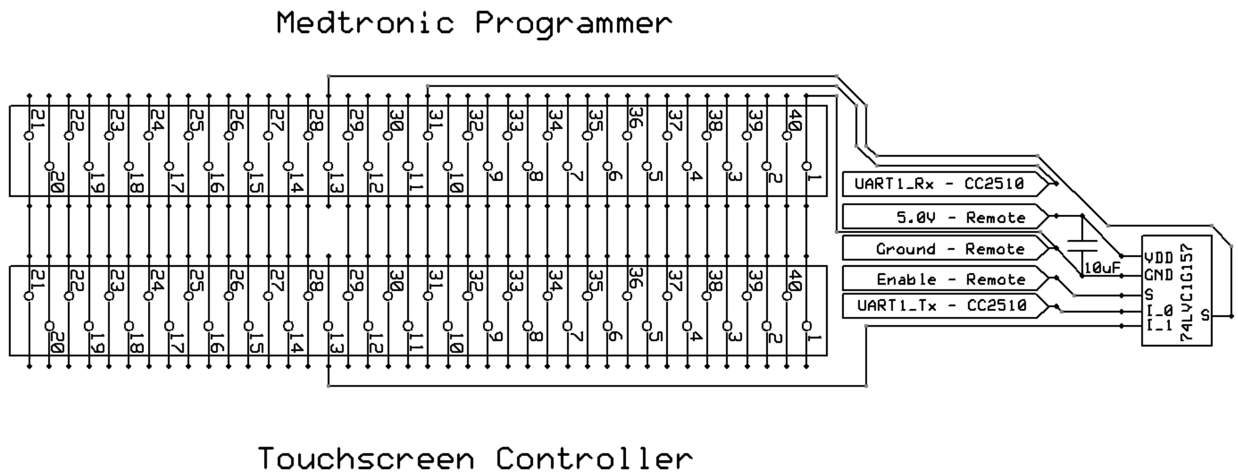

Figure 26: Schematic of Touchscreen Connection PCB on Remote Unit

6.3 ADDITIONAL HARDWARE ASPECTS

6.3.1 Debug Connection

Programming and debugging the CC2510 microcontroller are accomplished through a 10-pin debug interface, which is connected to the Texas Instruments CC Debugger [31]. This debugger uses a standard USB interface. The debugger communicates to the MCU through USART0 which is configured as an SPI. A conflict can occur between the debugger and the MCU if the USART0 interface is configured and the debugger halts the code at an interrupt from the IAR software. The data seen by the debugger on the USART0 buffers will not be correct because of this conflict. Figure 27 shows the schematic of the pin out for the debug interface to the CC2510 microcontroller. Table 3 elaborates on each pin and provides an explanation of each pin on the debug connection.

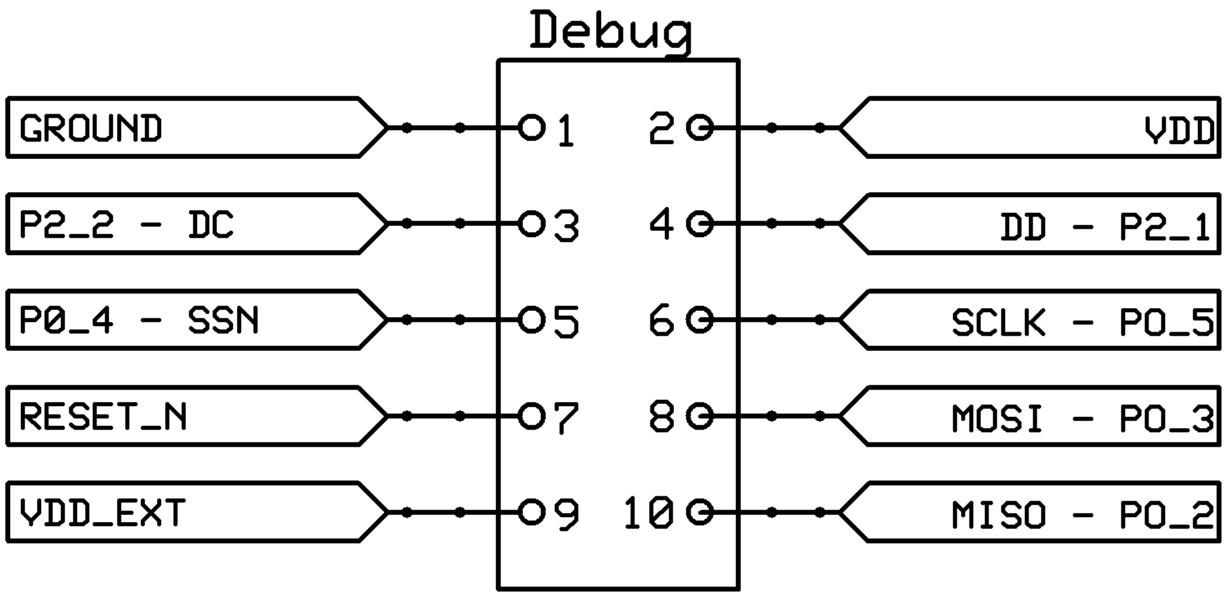

Figure 27: Schematic of Debug Interface to TI CC2510

Table 3: Elaboration of Debugger Pins Interfacing with TI CC2510

Pin #	Name	Description
1	GND	The pin provides a common ground between the microcontroller and the debugger
2	Target Voltage	This pin is used to provide a common VDD between the debugger and the CC2510. This can be connecter with pin 10 to power the CC2510 during programming and debugging.
3	Debug Clock (DC)	This pin provides a clock for programming of the CC251.
4	Debug Data (DD)	This pin is the data line that programs the CC2510 when the debugger is connected.
5	Chip Select (CS)	This pin is the standard Chip Select for an SPI interface. The CC Debugger is used as a Master SPI Device.
6	SPI Clock (SCLK)	This pin provides the clock for debugging of the CC2510 and is the standard SCLK for an SPI Interface.
7	RESETn	This pin is used during the programming of the CC2510 and is used to halt operation of the microcontroller.
8	SPI Data Out (MOSI)	This pin is used to configure breaks in code. It can interfere with the internal SPI/UART controller onboard the CC2510.

Table 3 (continued).

9	SPI Data In (MISO)	This pin is used to return values of registers to the CC2510 once a break point has occurred in the code.
10	3.3V from Debugger	This pin can be used to power the CC2510 during programming and debugging.

6.3.2 3M MicroTouch EX II Capacitive Touchscreen Controller

The MicroTouch EX II capacitive touchscreen controller is a commercial off the shelf (COTS) manufactured by 3M [23]. The EX II ASIC integrated into the touchscreen controller on the Medtronic 2090 Programmer communicates through two serial UART data lines at a 9600 baud rate. Data are transmitted through the 8N1 LSB communication standard discussed in Section 3.2.1. The standard format which commands are transmitted to the EX II controller is <Header> Command <Terminator>. The header byte is the ASCII character for start-of-header, 0x01. The terminator byte is the ASII character for carriage return, 0x0D. The command is a one byte sequence that allows for configuring the EX II controller.

The standard format in which commands are received from the EX II controller is <Header> Response <Terminator>. The header byte is the ASCII character for start-of-header, 0x01. The terminator byte is the ASII character for carriage return, 0x0D. Two responses from the EX II controller can be received. The ASCII character for 0, 0x30, indicates that the command transmitted was successfully completed. The ASCII character for 1, 0x31, indicates that the command transmitted has failed.

When a touch is registered by the controller, touchscreen location data are transmitted in a 5 byte packet which includes X and Y location data and a proximity bit. The proximity bit determines whether the pen is touching or just lifted off of the capacitive touchscreen. Table 4 gives an explanation of the 5 byte touchscreen location data sequence.

Table 4: 5-Byte Touchscreen Location Data from EX II Controller

Data Sequence	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
S – Byte 1	1	RESERVED						
X – Byte 2	0	X3	X2	X1	X0	RESERVED		
x – Byte 3	0	Xs	X9	X8	X7	X6	X5	X4
Y – Byte 4	0	Y3	Y2	Y1	Y0	RESERVED		
y – Byte 5	0	Ys	Y9	Y8	Y7	Y6	Y5	Y4

6.3.3 Printed Circuit Board

The printed circuit board (PCB) is the medium in which all components, ICs and interconnects are placed. The PCBs used in this thesis are made of 0.062 inch thick FR-4 epoxy glass substrate and were produced by ExpressPCB [32]. Boards were designed on both 2-layer boards and 4-layer boards. 2-layer boards have a top and bottom layer of copper in which IC's, components and interconnects are routed. The 4-layer boards have the top and bottom layer and include two internal layers for ground and power planes. These planes are layers of copper that allow for short connections to VDD and GND. The connection of components and integrated circuits is accomplished by metalized traces of copper that have a height of 0.0017 inches and varying widths depending on current requirements.

7.0 SOFTWARE DESIGN

The software design for the real-time active telecardiology system was programmed in IAR Embedded Workbench for 8051 KickStart Edition [33]. Separate software was designed for both the host unit and remote unit. The following sections provide an in-depth explanation of the major aspects and flow of the software designed for the host unit and the remote unit as specified in Section 5.0

7.1 HOST UNIT SOFTWARE DESIGN

The following subsections detail the specifics of the host unit microcontroller's software providing an in-depth analysis of the startup procedures and data filtering performed by the host unit.

7.1.1 Host Unit Software Design and Flowchart

Figure 28 illustrates the flow of the host unit's software and Figure 29 illustrates the flow of the DMA interrupt service routine (ISR). When the software begins execution, the 26 MHz clock is configured because the UART communication is based on this clock. Because the microcontroller is turned on before the Medtronic 2090 Programmer, the Rx UART pin is configured as a GPIO input and the Port 1 ISR is enabled to trigger on the rising edge. Because the UART has a logic high idle state, when the Medtronic 2090 Programmer is initially turned on, the idle state will cause a false trigger on the on the microcontroller's UART and disrupt the flow of the data. Once the Port 1 ISR is reached, a port semaphore is cleared and an 8-bit timer is configured to run for 322 milliseconds. This was done to prevent false triggers on the UART caused by noise on the idle data line during the startup.

Once the timer is finished running, a timer semaphore is cleared and UART0, UART1 and the three DMA channels are configured. The program waits for data to be received from the touchscreen controller. Each DMA channel is configured to transfer data either to or from the UART buffers. DMA Channel 0 and Channel 1 are configured to receive a block data transfer from UART1 to a buffer. Both DMA channels are triggered on the reception of data from UART1 but the size of the transfer differs. A start up response from the touchscreen controller to the programmer is a 3 byte packet while the touchscreen location data are transferred in a 5 byte packet.

The ISR for the DMA Channel 0 will occur after 3 bytes of data are received. If the data are determined to be a standard response from the touchscreen controller, the transfer on DMA channel 0 and channel 1 are aborted. If the data are not a standard response, nothing is done and the ISR for DMA channel 1 is then triggered. The data are interleaved and added to the buffer. A 2-byte match sequence is added to the end of the buffer to signal the Lantronix MatchPort controller to transmit the data immediately. DMA Channel 2 is then armed for transmission on UART0 and a DMA transfer is initiated by writing the first byte of the data to the UART0 buffer.

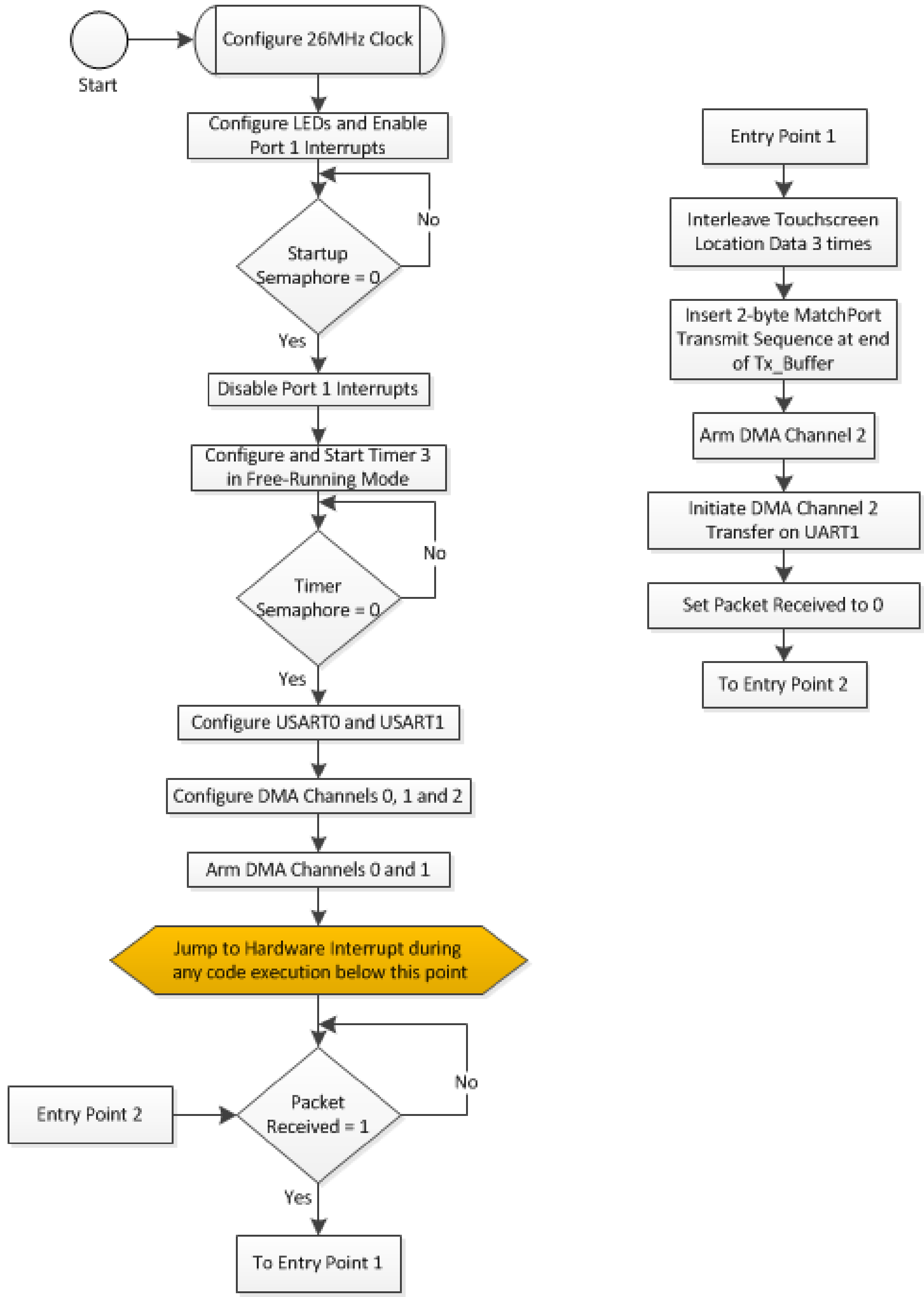

Figure 28: Flowchart of Main on Host Unit

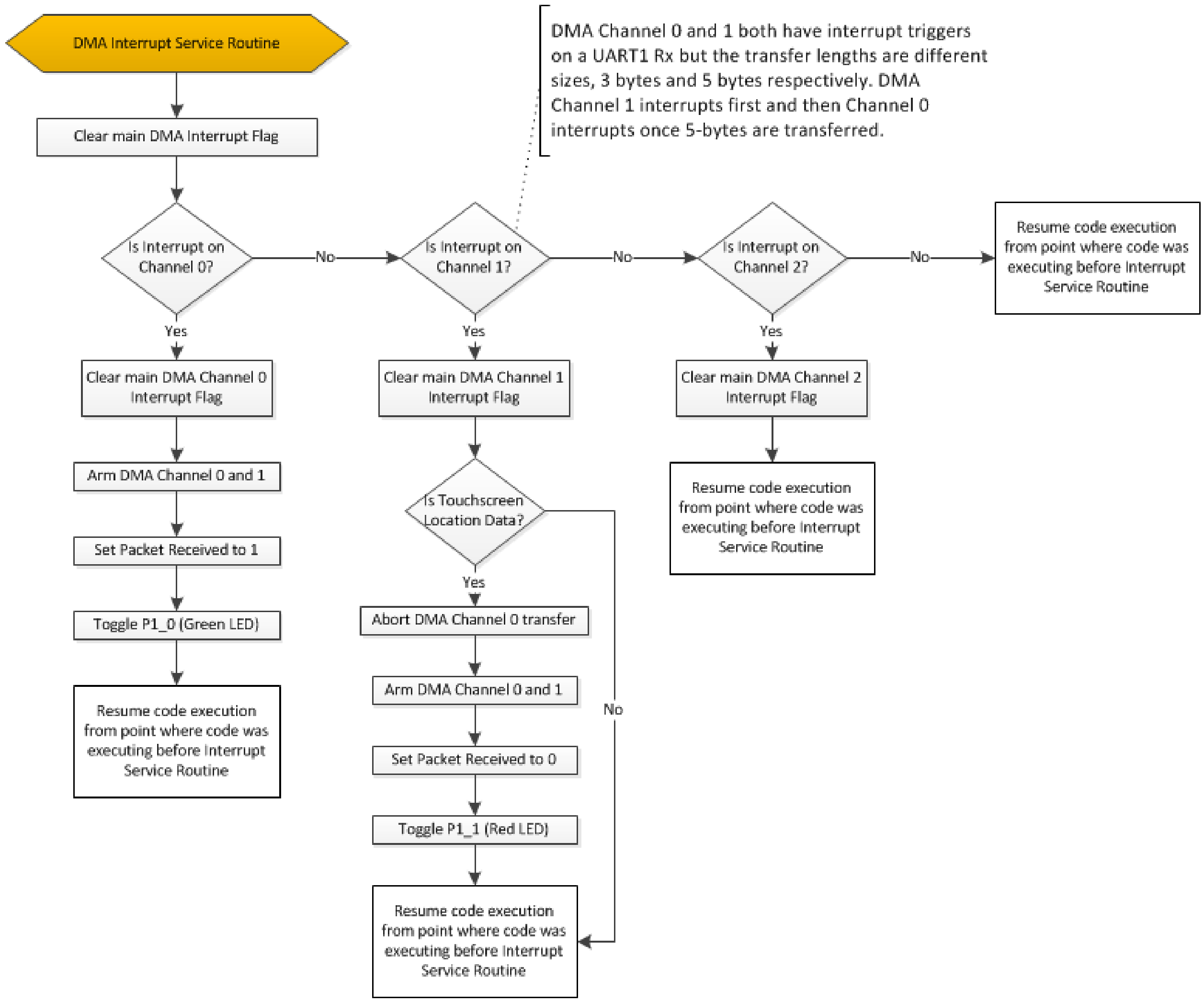

Figure 29: Flowchart of DMA ISR on Host Unit

7.2 REMOTE UNIT SOFTWARE DESIGN

The following subsections detail the specifics of the remote unit microcontroller's software providing an in-depth analysis of the startup procedures, data flow and interrupts performed by the remote unit.

7.2.1 Remote Unit Software Design and Flowchart

Figure 30 illustrates the flow of the remote unit's software and Figure 31 illustrates the flow of the DMA interrupt service routine (ISR). The startup procedure for the remote unit's microcontroller is the same as the host unit's microcontroller's startup procedure described in Section 7.1.1.

Once the timer is finished running, a timer semaphore is cleared and UART0, UART1 and the three DMA channels are configured. The remote unit's microcontroller acts as a pseudo touchscreen controller to the programmer and must properly interpret requests. DMA Channel 1 is configured to receive data on UART1, which are requests from the programmer. DMA Channel 2 is configured to transmit data on UART1, which are responses to the programmer. Once a request is received, a response is generated and a DMA Channel 2 transfer is initiated all inside of the DMA Channel 1 ISR.

DMA Channel 0 is configured to receive data on UART0 which is the touchscreen location data transmitted from the Lantronix MatchPort. When data are received on UART0, a packet received semaphore is set and the touchscreen location data are de-interleaved. In order to transmit the 5 byte touchscreen location data packet, DMA Channel 2 must be reconfigured and armed before transmission can occur.

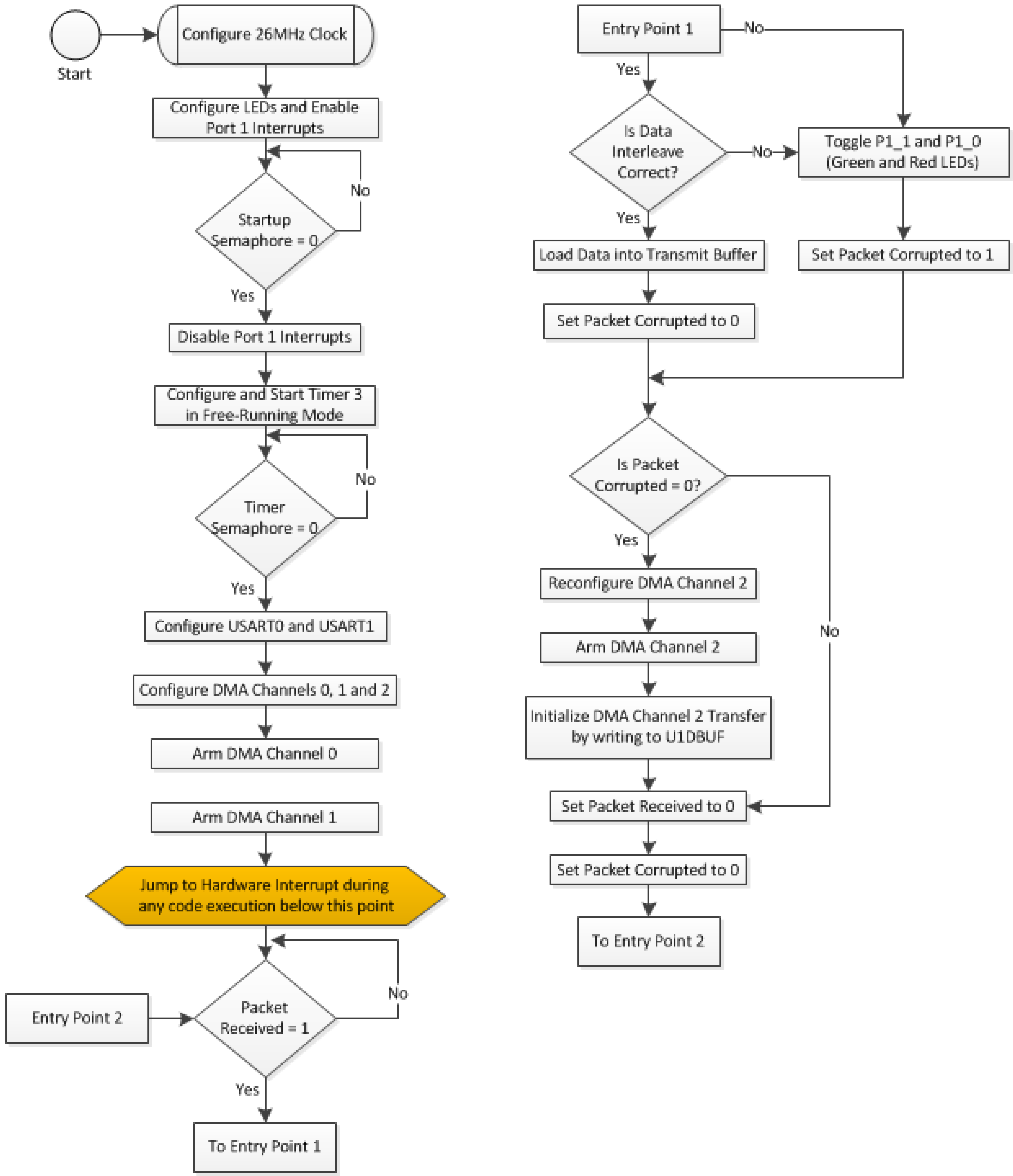

Figure 30: Flowchart of Main on Remote Unit

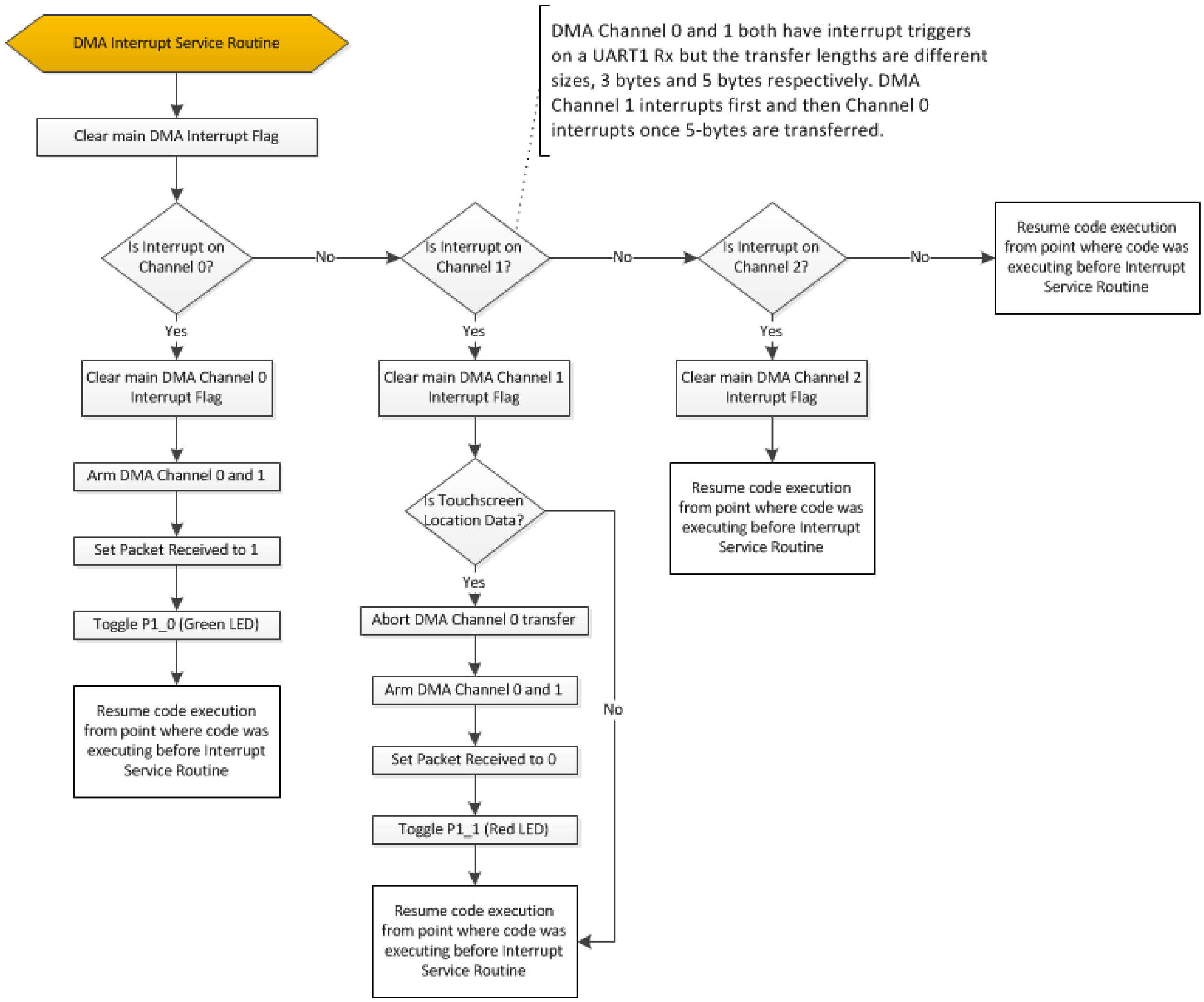

Figure 31: Flowchart for DMA ISR on Remote Unit

8.0 UNIT SCHEMATICS, PROGRAMMING AND SET-UP

Proper set-up and programming of the host unit and the remote unit are essential to the operation of the real-time active telecardiology system detailed in Section 2.0 . Figure 32 show the fully configured host and remote units. Due to circumstances at the end of the project, a final PCB for the host and remote units was designed but never manufactured. The following subsections explore how the host and remote units are programmed.

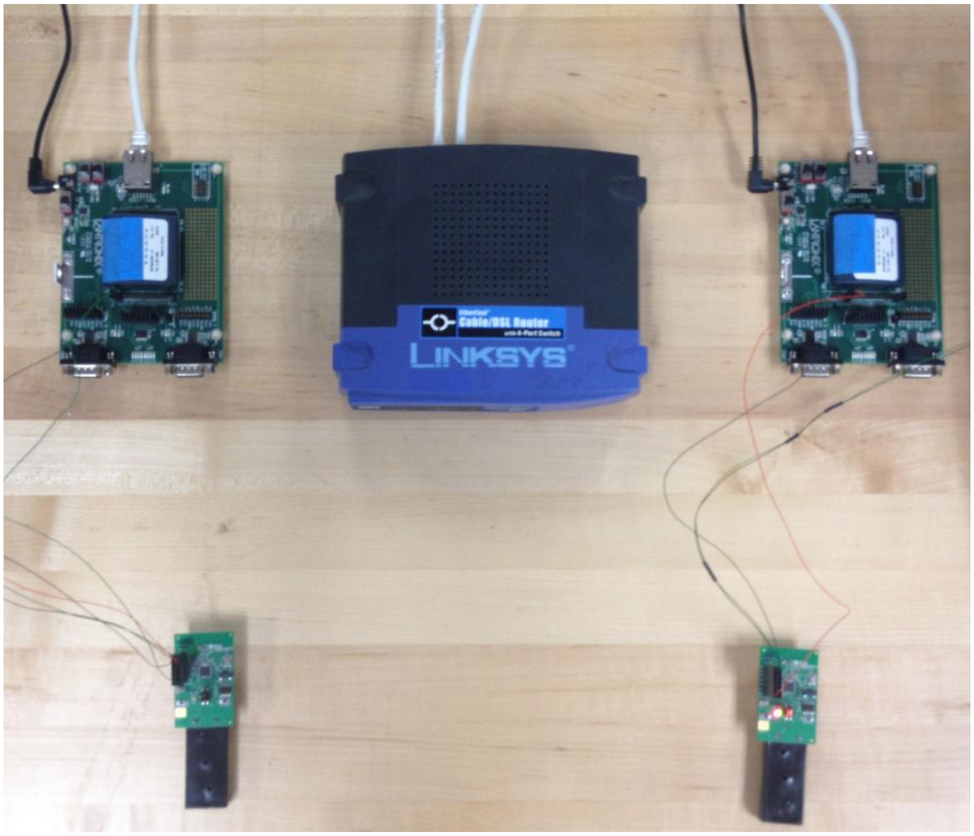

Figure 32: Proof-of-Concept Real-Time Active Telecardiology System

8.1 PROGRAMMING THE CC2510 MICROCONTROLLERS

The code programmed to the host and remote units was written in C using the IAR Embedded Workbench for 8051 KickStart Edition software. The units were programmed using the Texas Instruments CC Debugger through a 10-pin interface and a USB port on a computer, shown in Figure 33.

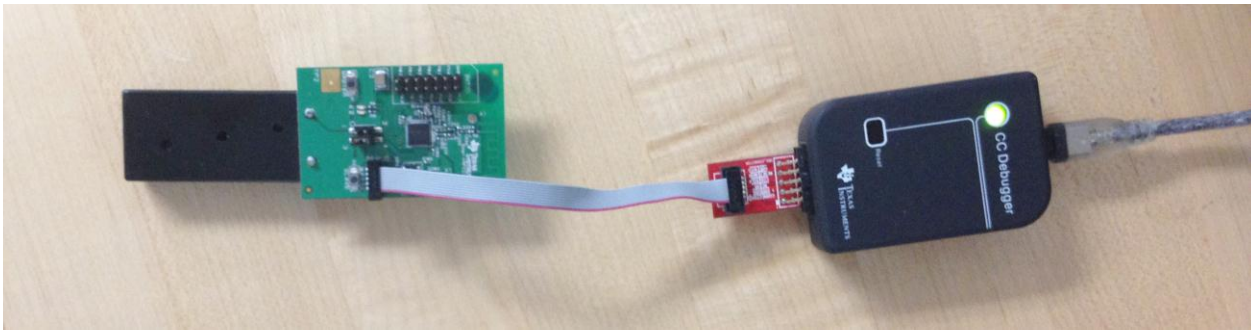

Figure 33: TI CC2510 Connection to CC Debugger

8.2 PRINTED CIRCUIT BOARD AND SCHEMATICS

The printed circuit boards (PCB) allow for components to be connected in an organized fashion. The PCBs are first designed in a schematic representation. The schematics are then linked to the PCB layout to ensure all components on the PCB are properly connected. The PCBs schematics and layouts were designed using ExpressPCB software. ExpressSCH version 7.02 was used to design the schematics for the host and remote units. ExpressPCB version 7.02 was used to design the board layouts of the host and remote units. The following subsections detail the schematics and board layouts for the host and remote unit and the touchscreen connection

PCBs that interface with the touchscreen controller on both the host and remote Medtronic 2090 Programmer.

8.2.1 Host Unit Schematic and Board Layout

Figure 33 and Figure 34 show the host unit schematic and PCB layout as designed with the ExpressPCB software. Figure 35 shows the touchscreen connection board layout as designed with the ExpressPCB software.

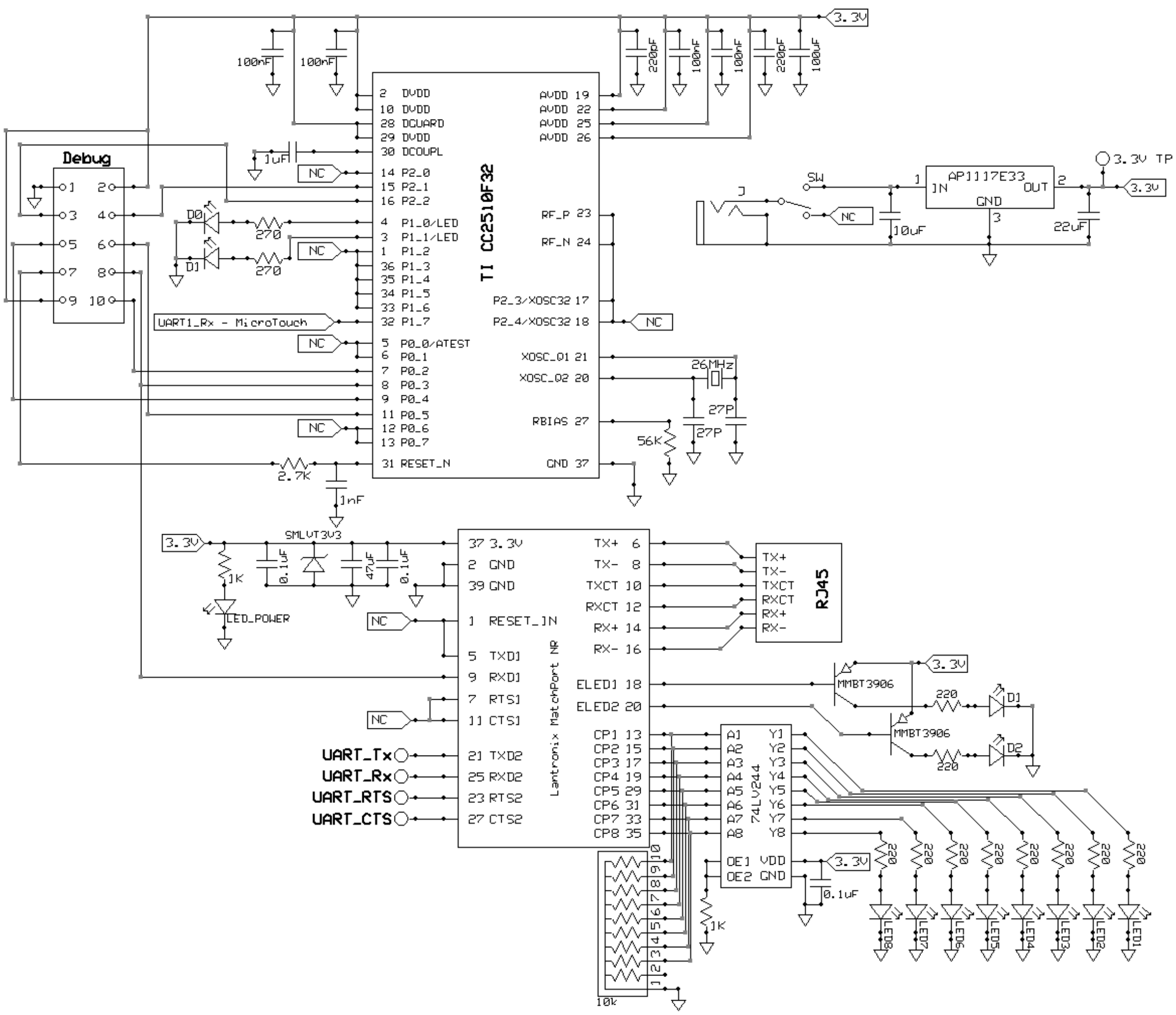

Figure 33:Schematic of Assembled Host Unit

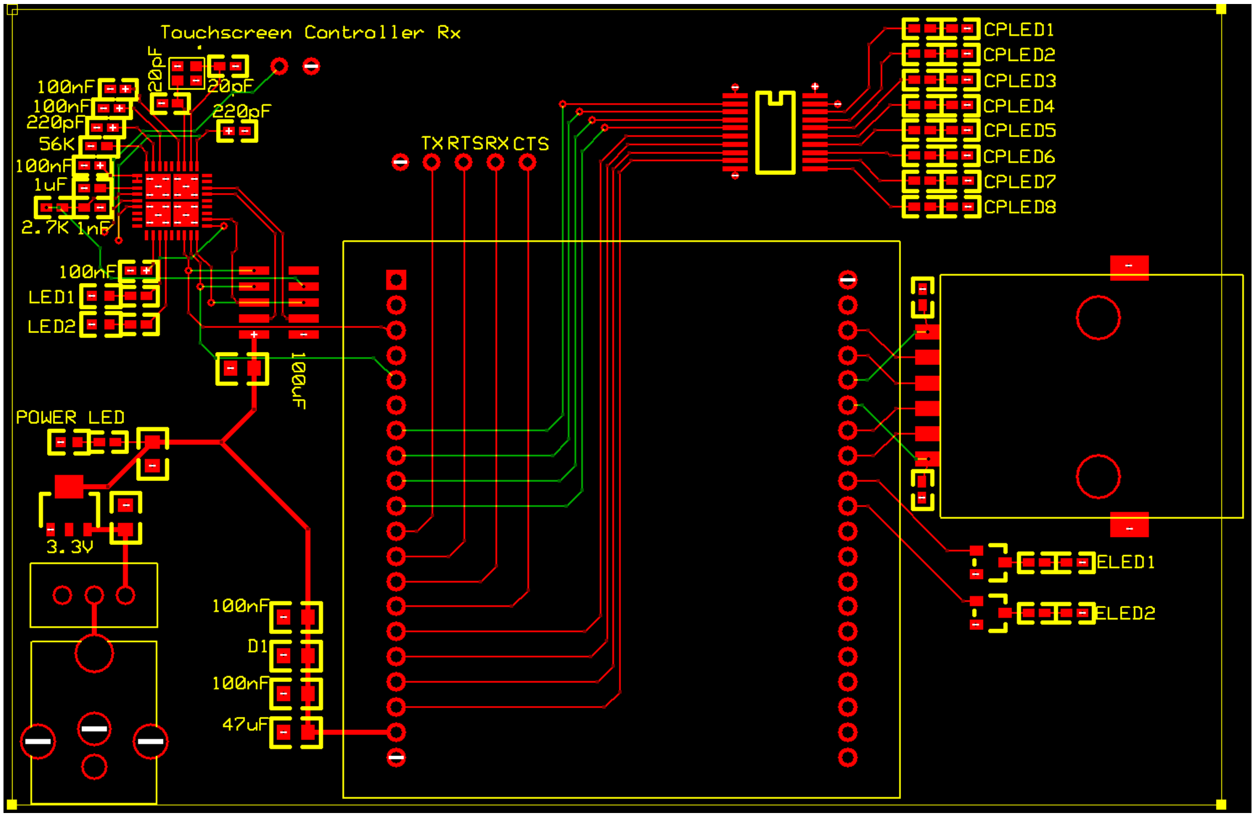

Figure 34: PCB Layout of Host Unit

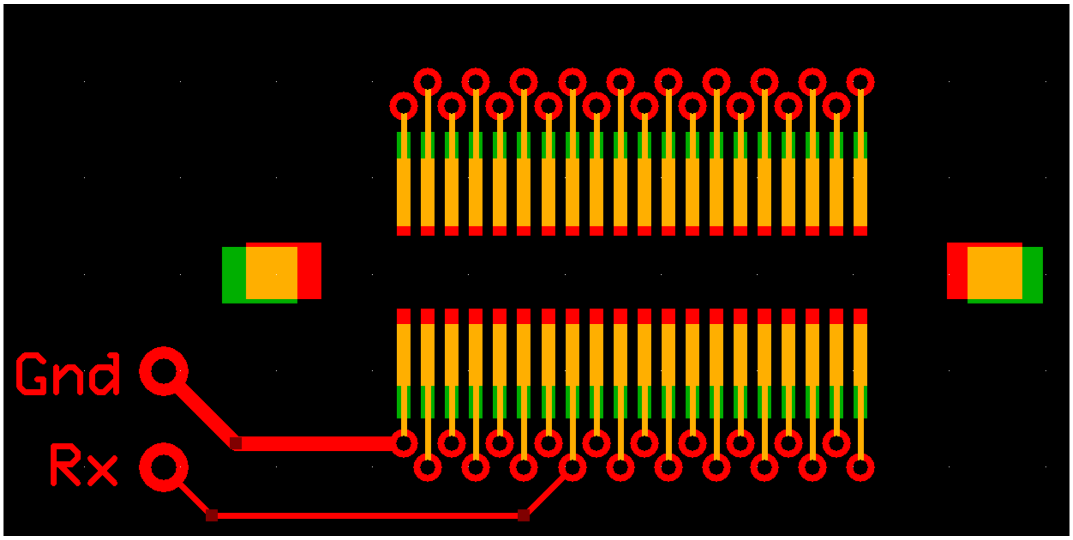

Figure 35: PCB Layout of Touchscreen Connection Board on Host Unit

8.2.2 Remote Unit Schematic and Board Layout

Figure 36 and Figure 38 show the remote unit schematic and PCB layout as designed with the ExpressPCB software. Figure 38 shows the touchscreen connection board layout as designed with the ExpressPCB software.

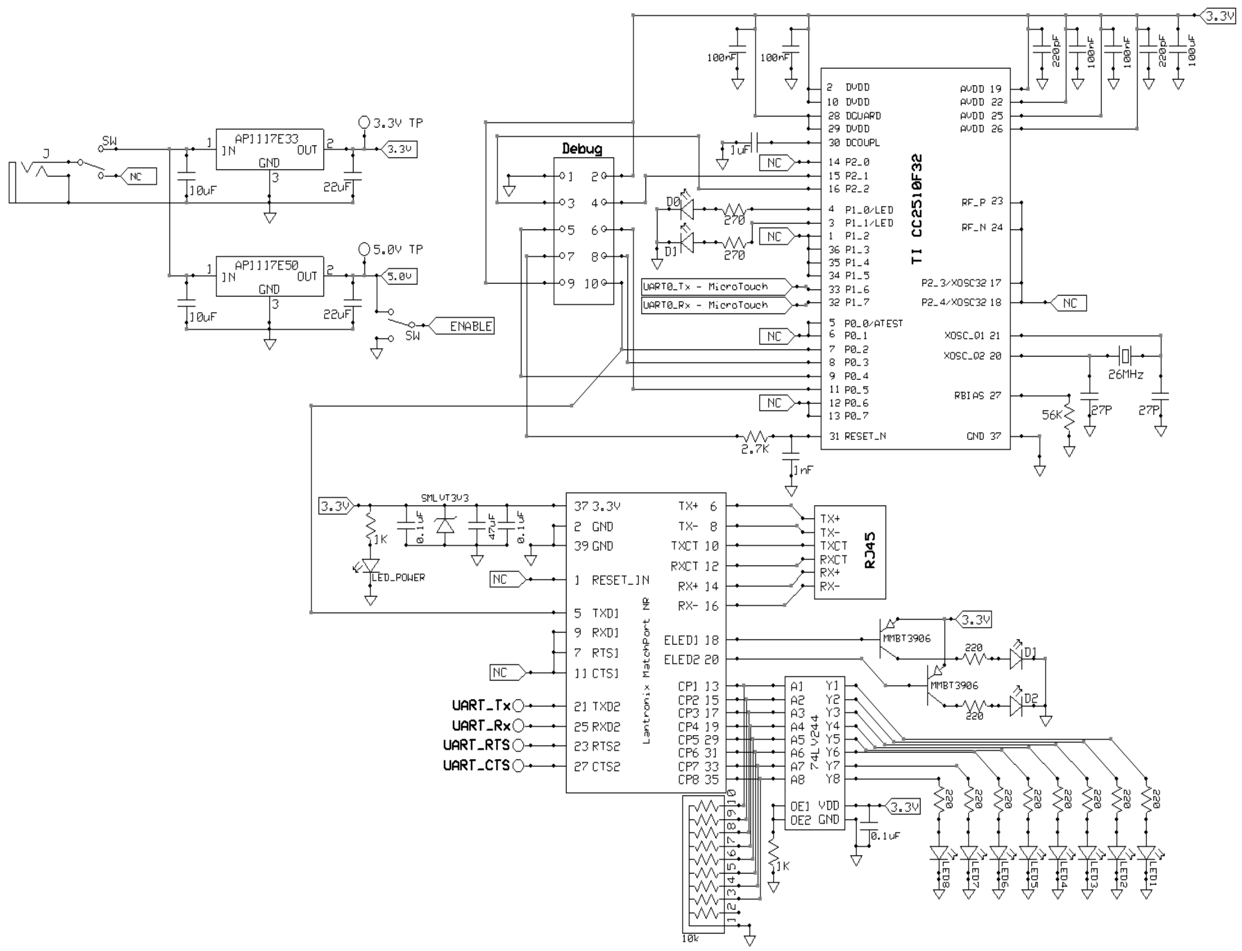

Figure 36: Schematic of Assembled Remote Unit

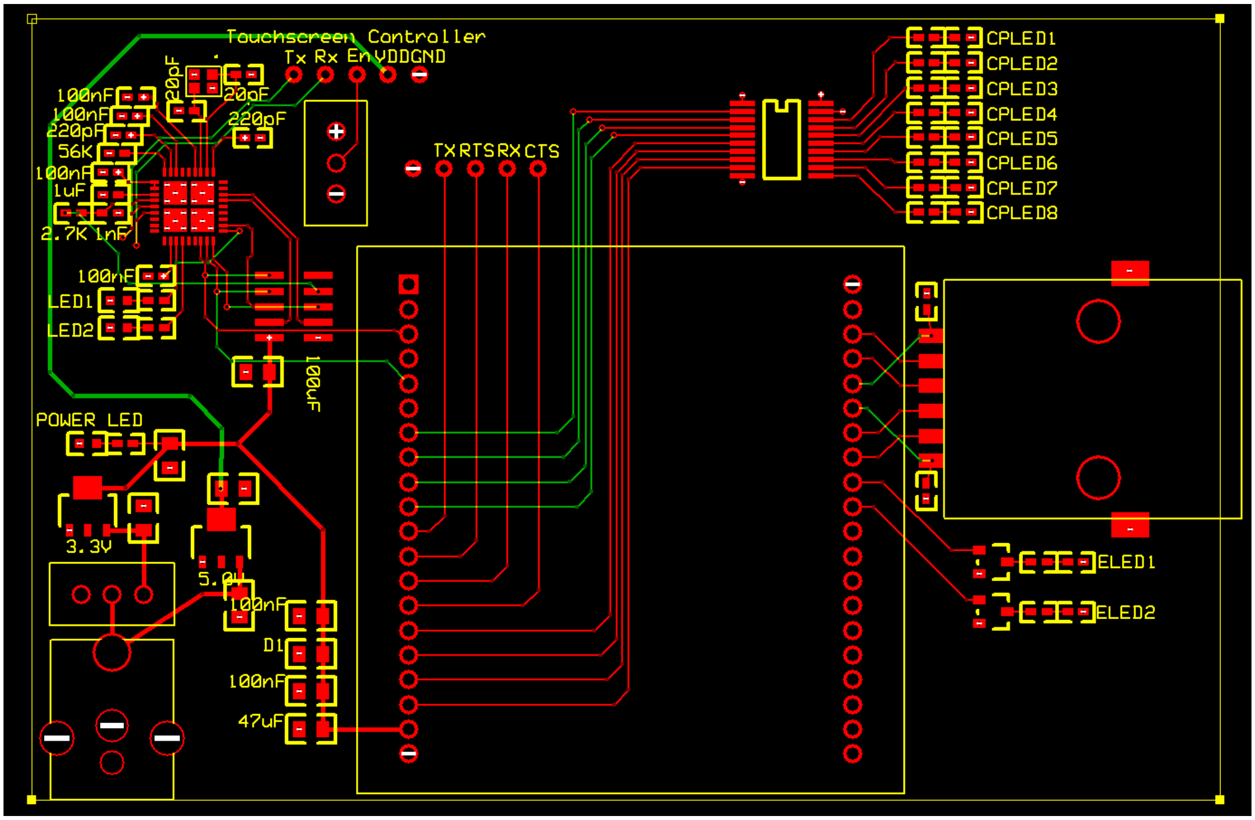

Figure 37: PCB Layout of Remote Unit

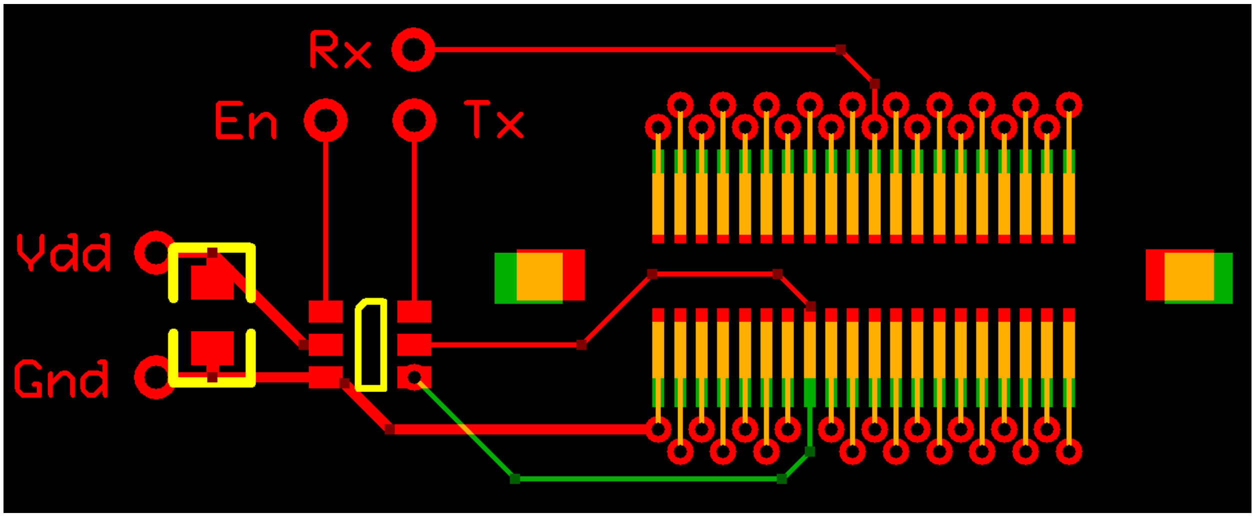

Figure 38: PCB Layout of Touchscreen Connection Board on Remote Unit

8.3 LOCAL AREA NETWORK CONFIGURATION

Proper configuration of the LAN and Lantronix MatchPort controllers is vital to the operation of the real-time active telecardiology system presented in this thesis. The following subsections will detail the required configurations for the LAN router and the Lantronix MatchPort devices.

8.3.1 LAN Router Configuration

The LAN router used in this thesis is the Linksys BEFSR41 EtherFast 4-port router. The BEFSR41 version 4.0 was chosen due to the availability of the router but any router will suffice. No configuration changes need to be made on the router itself although information concerning the operation of the router is required. All routers can dynamically assign IP addresses to a network interface card (NIC). This dynamic host configuration protocol (DHCP) temporarily assigns an IP address to a NIC for a given lease time. Once the lease expires, the NIC must request an extension or another IP address. This configuration can cause problems when two NICs are attempting to communicate with each other. The IP addresses of each NIC are not constant which will create problems when the telecardiology system is implemented.

To overcome this problem, IP addresses were statically assigned inside of the Lantronix MatchPort controllers. The only knowledge of the router required is the range of the DHCP server which is usually 50 addresses out of a possible 255 on a given subnet. This information is required as to reduce network conflicts when the DHCP server assigns the same IP address to a NIC with a static address. The DHCP server on the Linksys delivered IP addresses between

192.168.1.100 to 192.168.1.149. The IP addresses assigned to the host and remote Lantronix MatchPort controllers were 192.168.1.10 and 192.168.1.20, respectively.

8.3.2 Lantronix MatchPort Controller Configuration

Configuration of the host and remote MatchPorts can be processed through either the web interface or a serial communication link. Appendix A shows a printout from the serial interface of the configurations that are loaded onto the host unit's MatchPort controller. Appendix B shows a printout from the serial interface of the configurations that are loaded onto the remote unit's MatchPort controller.

8.4 SYSTEM DEMONSTRATION

In order to demonstrate the system, the Lantronix MatchPort controllers must connected to the LAN router, turned on and given a short time to open a connection. Once this connection is made, the TI CC2510 mini development board can be turned on at both the host unit and the remote unit. The red and green LEDs will signify that the microcontroller is waiting for the Medtronic Programmer to be turned on. By turning on the Medtronic Programmer, both LEDs will toggle to the off state and the Medtronic Programmers will begin to startup. Once startup has completed on both programmers, the host Medtronic 2090 Programmer can be used as though the remote programmer were being operated on.

9.0 RESULTS

The real-time active telecardiology system described in this thesis is capable of transmitting the touchscreen location data from the host Medtronic 2090 Programmer to the remote Medtronic 2090 Programmer in real-time. The logic analyzer used for verification is the Saleae Logic16. It was chosen due to the high sampling rate, up to 100MHz, and the large number of samples able to be recorded, up to 1 trillion samples. The Logic16 was configured to sample 3 channels at 100MHz to ensure that each waveform was properly sampled.

Figure 39 shows the Logic16 reading of the noise which is inherent in the data line connected to the microcontroller during startup of the Medtronic 2090 Programmer. The noise lasts for approximately 32 milliseconds at which point the idle logic high is attained. The source of the noise is most likely due to the mismatched impedance of the microcontroller UART. Figure 40 shows the processing time required by the microcontroller once the touchscreen location packet is received to the transmission of the interleaved data to the serial-to-ethernet controller. Because little to no processing is required and the high clock speed of the microcontroller, this time is negligible at 27.8 microseconds.

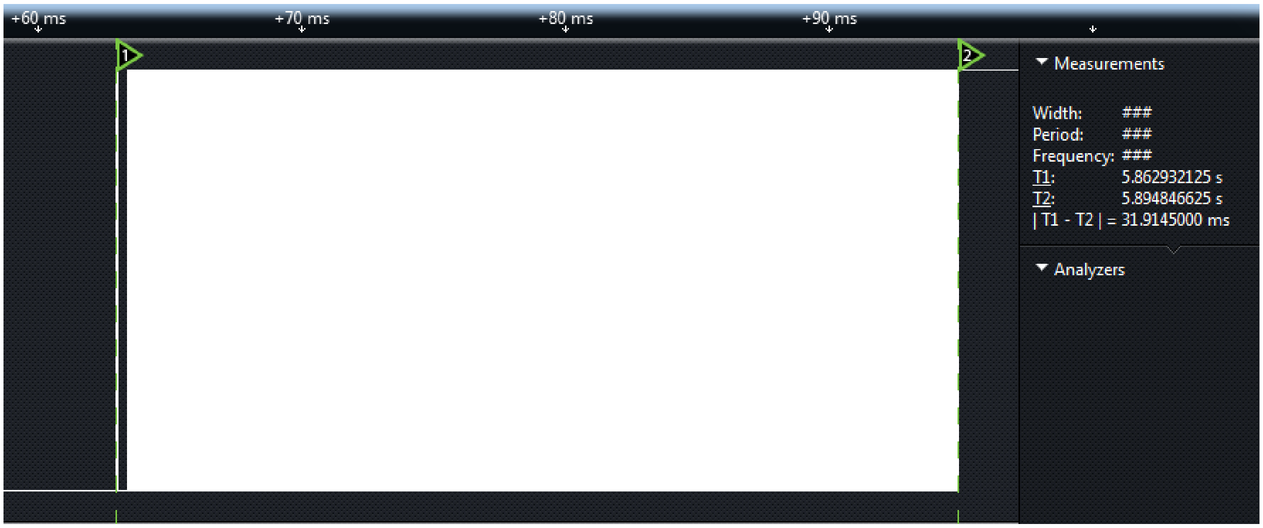

Figure 39: Startup Noise on UART Data Lines

Figure 40: Processing Time of Microcontroller

Figure 41 shows the transmission time of the packet through the local area network. This will be the largest latency in the system and will vary depending on the speed of the network connection and current network load. The latency of 1.4 milliseconds seen should be considered the fastest transmission time because the data were transmitted through a local area network (LAN) with no network load. These tests were performed with no encryption enabled on the Lantronix MatchPort. Depending on the level of encryption enabled on the MatchPort, this latency will increase by an unknown amount of time. The total latency of the system from the transmission of the first bit at the host Medtronic 2090 Programmer to the last bit received by the remote Medtronic 2090 Programmer is approximately 12.21 milliseconds. This latency does not include the processing time required by the remote programmer to register the touchscreen location data. In testing, this 12.21 millisecond delay is not perceivable by the user and is considered a real-time solution.

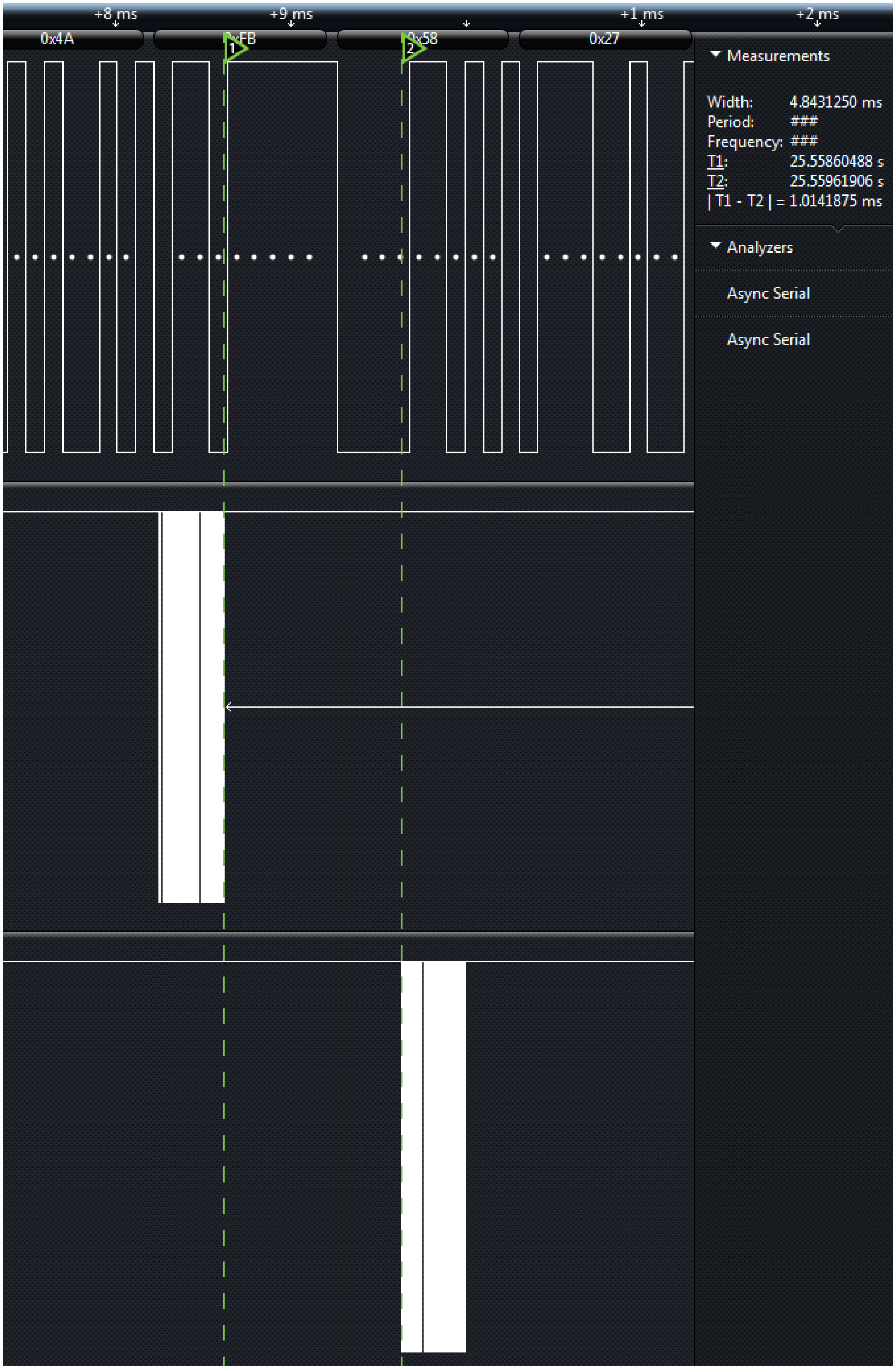

Figure 41: Latency of a Packet through the LAN

10.0 CONCLUSION

The real-time active telemedicine system presented in this thesis is a viable solution to the requirements of the system detailed in Section 2.1. First, the real-time nature of the system has been met. This is the most important requirement of the system and was achieved with 12 millisecond delay allowing full control of a remote Medtronic 2090 Programmer. Second, the redundancy of the system was met. The redundancy requirement was imposed to ensure that errant touches were reported to the remote Medtronic 2090 Programmer and was solved by interleaving the data at the host unit. Third, the security of the system was not completely met. Although encryption is available on the MatchPort, the encryption was not configured due to time constraints. Finally, the requirement that the system be scalable was met and is discussed in Section 11.0 If further development is followed with the details presented in the Section 11.0 a real-time active telemedicine system presented in this thesis can be achieved and implemented in the current hospital system.

11.0 FUTURE WORK

A major requirement that this thesis does not cover is the transmission of the video feed from the remote Medtronic 2090 Programmer to the host Medtronic 2090 Programmer. This is of considerable importance but was not covered in this thesis. For this system to be widely implemented, a solution must be created. The resolution of the display is 1024 by 768 pixels with 8-bit RGB resolution and a refresh rate of 60Hz. Without a form of video compression, transmission over a wide area network is not a viable solution due to throughput required.

The host Medtronic 2090 Programmer simply acts as a touchscreen with an LCD. By replacing the host Medtronic 2090 Programmer with an external monitor and a touchscreen overlay, the cost of the system is reduced while increasing the portability. The remote Medtronic 2090 Programmer could also be controlled by a smartphone or tablet. This further removes costs inherent to the system and allows the cardiologist to program a pacemaker from any location.

Because the MatchPort controllers connect to each other through a programmable IP address, the scalability of the system only depends on the ability of the host microcontroller to program the host unit's MatchPort controller with a specific server IP address. The scalability can only be limited by the internet connection of the remote Medtronic 2090 Programmer.

APPENDIX A

CONFIGURATION OF HOST LANTRONIX MATCHPORT NR

Appendix A details the configuration of the Host Lantronix MatchPort NR serial-to-ethernet controller mentioned in Section 6.1.2.The profile was accessed by entering into the Setup Mode through the Lantronix DeviceInstaller software. For more information on programming the MatchPorts, reference the MatchPort User Guide [26].

```
*** basic parameters
Hardware: Ethernet TPI
IP addr 192.168.1.10, gateway 192.168.1.1,netmask 255.255.255.0
DNS Server 192.168.1.1

*** Security
SNMP is                    enabled
SNMP Community Name:       public
Telnet Setup is            enabled
TFTP Download is           enabled
Port 77FEh is              enabled
Web Server is              enabled
Web Setup is               enabled
ECHO is                    disabled
Encryption is              disabled
Enhanced Password is       disabled
Port 77F0h is              enabled

*** Channel 1
Baudrate 460800, I/F Mode 4C, Flow 00
Port 02000
Connect Mode : 25
Send '+++' in Modem Mode disabled
```

Show IP addr after 'RING' disabled
Auto increment source port disabled
Hostlist :
01. IP : 192.168.001.020 Port : 02000
02. IP : 192.168.001.020 Port : 02000
03. IP : 192.168.001.020 Port : 02000
04. IP : 192.168.001.020 Port : 02000
Hostlist Retrycounter : 3
Hostlist Retrytimeout : 250
Disconn Mode : 00
Flush   Mode : F7
Pack Cntrl   : 30
SendChars   : A4 A5

*** Channel 2
Channel 2: disabled

*** Expert
TCP Keepalive   : 45s
ARP cache timeout: 600s
CPU performance: High
Monitor Mode @ bootup : enabled
HTTP Port Number : 80
SMTP Port Number : 25
MTU Size: 1400
Alternate MAC: disabled
Ethernet connection type: auto-negotiate

*** E-mail
Mail server: 0.0.0.0
Unit   :
Domain   :
Recipient 1:
Recipient 2:
- Trigger 1
    Serial trigger input: disabled
    Channel: 1
    Match: 00,00
    Trigger input1: X
    Trigger input2: X
    Trigger input3: X
    Message :
    Priority: L
    Min. notification interval: 1 s
    Re-notification interval : 0 s
- Trigger 2

Serial trigger input: disabled
Channel: 1
Match: 00,00
Trigger input1: X
Trigger input2: X
Trigger input3: X
Message :
Priority: L
Min. notification interval: 1 s
Re-notification interval : 0 s

- Trigger 3

Serial trigger input: disabled
Channel: 1
Match: 00,00
Trigger input1: X
Trigger input2: X
Trigger input3: X
Message :
Priority: L
Min. notification interval: 1 s
Re-notification interval : 0 s

APPENDIX B

CONFIGURATION OF REMOTE LANTRONIX MATCHPORT NR

Appendix B details the configuration of the Remote Lantronix MatchPort NR serial-to-ethernet controller mentioned in Section 6.2.2. The profile was accessed by entering into the Setup Mode through the Lantronix DeviceInstaller software. For more information on programming the MatchPorts, reference the MatchPort User Guide [26].

```
*** basic parameters
Hardware: Ethernet TPI
IP addr 192.168.1.20, gateway 192.168.1.1,netmask 255.255.255.0
DNS Server 192.168.1.1

*** Security
SNMP is                    enabled
SNMP Community Name:       public
Telnet Setup is            enabled
TFTP Download Is           enabled
Port 77FEh is              enabled
Web Server is              enabled
Web Setup is               enabled
ECHO is                    disabled
Encryption is              disabled
Enhanced Password is       disabled
Port 77F0h is              enabled

*** Channel 1
Baudrate 460800, I/F Mode 4C, Flow 00
Port 02000
Connect Mode : C0
Send '+++' in Modem Mode disabled
```

Show IP addr after 'RING' disabled
Auto increment source port disabled
Remote IP Adr: 192.168.1.10, Port 02000
Disconn Mode : 00
Flush   Mode : 77
*** Channel 2
Channel 2: disabled

*** Expert
TCP Keepalive   : 45s
ARP cache timeout: 600s
CPU performance: High
Monitor Mode @ bootup : enabled
HTTP Port Number : 80
SMTP Port Number : 25
MTU Size: 1400
Alternate MAC: disabled
Ethernet connection type: auto-negotiate

*** E-mail
Mail server: 0.0.0.0
Unit   :
Domain   :
Recipient 1:
Recipient 2:
- Trigger 1
    Serial trigger input: disabled
    Channel: 1
    Match: 00,00
    Trigger input1: X
    Trigger input2: X
    Trigger input3: X
    Message :
    Priority: L
    Min. notification interval: 1 s
    Re-notification interval : 0 s
- Trigger 2
    Serial trigger input: disabled
    Channel: 1
    Match: 00,00
    Trigger input1: X
    Trigger input2: X
    Trigger input3: X
    Message :
    Priority: L
    Min. notification interval: 1 s

Re-notification interval : 0 s

- Trigger 3

    Serial trigger input: disabled
    Channel: 1
    Match: 00,00
    Trigger input1: X
    Trigger input2: X
    Trigger input3: X
    Message :
    Priority: L
    Min. notification interval: 1 s
    Re-notification interval : 0 s

BIBLIOGRAPHY

[1] Mond, Harry G., and Alessandro Proclemer. "The 11th World Survey of Cardiac Pacing and Implantable Cardioverter-Defibrillators: Calendar Year 2009-A World Society of Arrhythmia's Project." *Pacing and Clinical Electrophysiology* 34.8 (2011): 1013-027. Print.

[2] Mond, Harry G., Marleen Irwin, Hugo Ector, and Alessandro Proclemer. "The World Survey of Cardiac Pacingand Cardioverter-Defibrillators: Calendar Year 2005." *Pacing and Clinical Electrophysiology* 31.9 (2008): 1202-212. Print.

[3] "What Is an Arrhythmia?" *National Heart Lung and Bood Institute*. National Institute of Health, 1 July 2011. Web. 2 Mar. 2012. <http://www.nhlbi.nih.gov/health/health-topics/topics/arr/>.

[4] "How Does a Pacemaker Work?" *National Heart Lung and Bood Institute*. National Institute of Health, 28 Feb. 2012. Web. 2 Mar. 2012. <http://www.nhlbi.nih.gov/health/health-topics/topics/pace/howdoes.html>.

[5] "What To Expect During an Electrocardiogram." *National Heart Lung and Bood Institute*. National Institute of Health, 1 Oct. 2010. Web. 2 Mar. 2012. <http://www.nhlbi.nih.gov/health/health-topics/topics/ekg/during.html>.

[6] "How Does an Implantable Cardioverter Defibrillator Work?" *National Heart Lung and Bood Institute*. National Institute of Health, 9 Nov. 2011. Web. 19 Mar. 2012. <http://www.nhlbi.nih.gov/health/health-topics/topics/icd/howdoes.html>.

[7] "How Will a Pacemaker Affect My Lifestyle?" *National Heart Lung and Bood Institute*. National Institute of Health, 28 Feb. 2011. Web. 2 Mar. 2012. <http://www.nhlbi.nih.gov/health/health-topics/topics/pace/lifestyle.html>.

[8] "Willem Einthoven - Biography." Nobelprize.org. Web. 2 Mar. 2012. <http://www.nobelprize.org/nobel_prizes/medicine/laureates/1924/einthoven-bio.html>.

[9] Barold, SS. "Willem Einthoven and the Birth of Clinical Electrocardiography a Hundred Years Ago." *Card Electrophysiol Rev* 7.1 (2003): 99-104. Print.

[10] Medtronic Inc. *Remote Monitoring of Pacemaker Patients on Medtronic Carelink® Network Detects Clinically Actionable Events More Quickly Than Combination of Transtelephonic Monitoring and Routine Office Visits*. 15 May 2008. Web. <http://wwwp.medtronic.com/Newsroom/NewsReleaseDetails.do?itemId=1210875085539&lang=en_US>.

[11] Fauchier, Laurent, Nicolas Sadoul, Claude Kouakam, Florent Briand, Michel Chauvin, Dominique Babuty, and Jacques Clementy. "Potential Cost Savings by Telemedicine-Assisted Long-Term Care of Implantable Cardioverter Defibrillator Recipients." *Pacing and Clinical Electrophysiology* 28.S1 (2005): S255-259. Print.

[12] "Adapta Pacemaker." *Adapta Family of Pacemakers for Bradycardia*. Medtronic Inc. Web. <http://www.medtronic.com/patients/bradycardia/device/our-pacemakers/adapta/index.htm>.

[13] "Unify Quadra CRT-D." *Unify Quadra CRT-D*. St. Jude's Medical. Web. <http://www.sjmprofessional.com/Products/US/CRT-Systems/Unify-Quadra-CRT-D.aspx>.

[14] "ALTRUA Pacemaker Family." *Boston Scientific Pacemakers*. Boston Scientific. Web. <http://www.bostonscientific.com/Device.bsci?navRelId=1000.1003>.

[15] "Dual-Chamber Pacemakers Worth Extra Cost By Reducing Risk Of Hospitalization And Disability." *ScienceDaily*. ScienceDaily, 04 Jan. 2005. Web. <http://www.sciencedaily.com/releases/2005/01/050104112300.htm>.

[16] Sandfuchs, Roland. "Touch-screen Technologies Enable Greater User/device Interaction." *EETimes*. EETimes, 24 July 2011. Web. <http://www.eetimes.com/design/analog-design/4218131/Touch-screen-technologies-enable-greater-user-device-interaction-?cid=NL_PlanetAnalog>.

[17] Blankenship, Tim. "Projected-Capacitive Touch Systems from the Controller Point of View." *Maxim Application Notes*. Maxim, 10 June 2011. Web. <http://www.maxim-ic.com/app-notes/index.mvp/id/5047>.

[18] Kolokowsky, Steve, and Trevor Davis. "Touchscreens 101: Understanding Touchscreen Technology and Design." *EETimes*. EETimes, 29 June 2009. Web. <http://www.eetimes.com/design/analog-design/4010387/Touchscreens-101-Understanding-touchscreen-technology-and-design>.

[19] Triebel, Walter A., and Singh Avtar. *The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware, and Applications : Including the 80286, 80386, 80486, and Pentium Processor Families*. Upper Saddle River, NJ: Prentice Hall, 2003. 536-58. Print.

[20] "Daisy-Chaining SPI Devices." *Maxim Application Notes*. Maxim, 15 Dec. 2006. Web. <http://www.maxim-ic.com/app-notes/index.mvp/id/3947>.

[21] Haykin, Simon S., and Michael Moher. *Modern Wireless Communications*. Upper Saddle River, NJ: Pearson Prentice Hall, 2003. 207-12. Print.

[22] "Medtronic." *Newsroom*. Medtronic Inc., 5 May 2005. Web. <http://wwwp.medtronic.com/Newsroom/NewsReleaseDetails.do?itemId=1115244764089>.

[23] "3M MicroTouch EX II Series Serial Controller." *3M Touch Systems*. 3M. Web. <http://multimedia.3m.com/mws/mediawebserver?mwsId=66666UuZjcFSLXTtMxfclxT2EVuQEcuZgVs6EVs6E666666--&fn=EXII_Serial-Cntrls_RefGuide.pdf>.

[24] "CC1110 & CC2510 Mini Development Kit User's Guide." Texas Instruments. Web. <http://www.ti.com/lit/ug/swru236a/swru236a.pdf>.

[25] "Low-Power SoC (System-on-Chip) with MCU, Memory, 2.4 GHz RF Transceiver, and USB Controller." Texas Instruments. Web. <http://www.ti.com/lit/ds/symlink/cc2510f32.pdf>.

[26] "MatchPort User Guide." Lantronix. Web. <http://www.lantronix.com/pdf/MatchPort-NR_UG.pdf>.

[27] "V-Infinity 12W Wall Plug." Web. <http://products.cui.com/CUI_EPSA120100U_Datasheet.pdf?fileID=2995>.

[28] "1A LOW DROPOUT POSITIVE ADJUSTABLE OR FIXED-MODE REGULATOR." Web. <http://www.diodes.com/datasheets/AP1117.pdf>.

[29] "Molex SlimStack 0.635mm Pitch." *Molex SlimStack*. Molex. Web. <http://www.molex.com/molex/products/family?key=slimstack_0635mm_025_pitch>.

[30] "74LVC1G157 Datasheet." *Single 2-input Multiplexer*. NXP. Web. <http://www.nxp.com/documents/data_sheet/74LVC1G157.pdf>.

[31] "CC Debugger User's Guide." Texas Instruments. Web. <http://www.ti.com/lit/ug/swru197e/swru197e.pdf>.

[32] "Manufacturing Specifications." ExpressPCB. Web. <http://www.expresspcb.com/ExpressPCBHtm/SpecsStandard.htm>.

[33] "IAR Embedded Workbench for 8051." *Embedded Compiler 8051*. IAR. Web. <http://www.iar.com/en/Products/IAR-Embedded-Workbench/8051/>.