

Microcontroller Design and Bluetooth Signal Transmission for the Non-Invasive Health Monitoring System (NIHMS)

By: Omer Waseem

EE 4BI6

Electrical and Biomedical Engineering Design Project
Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada

Microcontroller Design and Bluetooth Signal Transmission
for the Non-Invasive Health Monitoring System (NIHMS)

By: Omer Waseem

Electrical and Biomedical Engineering
Faculty Advisor: Prof. T. Doyle

EE 4BI6 Design Project Report submitted
in partial fulfilment of the degree of
Bachelor of Engineering

McMaster University
Hamilton, Ontario, Canada
April 27th, 2009

Copyright © April, 2009 by Omer Waseem

Abstract

The NIHMS is intended to be a multipurpose health monitoring device with current transducers for measuring photoplethysmography (PPG), electro-oculography (EOG), and temperature. All three of these transducers can provide useful data for diagnosis or experiments and EOG can provide an effective method for HCI. Through the use of Bluetooth technology the device can be paired with a wide variety of pre-existing devices (cell phones, PDAs and computers to name a few). This makes the device portable, cheaper and versatile. Data is collected from the transducers by a microcontroller with an attached Bluetooth module, which transmits the data to a host computer for processing. Costs are reduced by making use of pre-existing processing power and display on the host computer. Depending on the application the data can be processed in any way required on the host computer. For the purpose of this project the host computer application acts as a TCP/IP server to make the data available to clients over the internet. The data obtained, though noisy, is sufficient for measuring heart rate, breathing rate, blood pressure, oxygen saturation and EOG. Details on the hardware and software design for the microcontroller and Bluetooth data transmission for the NIHMS are presented in this report.

Keywords: photoplethysmography, PPG, electro-oculography, EOG, temperature, Bluetooth, wireless, human-computer interface, HCI, cost effective, pulse-oximeters, health monitoring

Acknowledgements

The author acknowledges his team members Aiyush Bansal, Yousuf Jawahar and Mastan Kalsi for their aid and support, Prof. T. Doyle for his guidance and oversight, Prof. M. Noteworthy for his help and encouragement and the IEEE department for their technical support.

Table of Contents

Abstract	ii
Acknowledgements	iii
Table of Content	iv
List of Tables	v
List of Figures	vi
Nomenclature	vii
1 Introduction	1
1.1 Objective	1
1.2 Methodology	2
1.3 Scope	3
2 Literature Review	5
2.1 Microcontroller Design	5
2.2 Bluetooth Technology	7
2.3 Wireless Technology in Medicine	9
3 Statement of Problem and Methodology of Solution	11
3.1 Problem Statement	11
3.2 Solution Methodology	12
4 Experiment and Design Procedures	15
4.1 Microcontroller Support Circuitry	15
4.2 Analog-to-Digital Converter	17
4.3 Thermometer	18
4.4 LCD Module	19
4.5 USART Setup and Bluetooth Connection	20
4.6 Transducer Integration	23
4.7 Host Computer Application	24
5 Results and Discussion	25
5.1 Transmission at 5.92 kbps	25
5.2 Transmission at 20.8 kbps	26
5.3 Transmitted Signal Characteristics	27
6 Conclusion and Recommendations	30

Appendix A: Supplementary Figures	32
Appendix B: Supplementary Tables	37
Appendix C: PIC Programs	39
References	42
Vitae	43

List of Tables

Table 4.1: Temperatures measured from different areas on the skin	18
Table B.1: Complete list of components used for the project	37
Table B.2: Data from the for-loop delay experiment	38

List of Figures

Figure 1.1: Conceptual block diagram of the NIHMS	2
Figure 2.1: Cosmanescu A. et al. design block diagram	6
Figure 2.2: Core Bluetooth protocol groups	8
Figure 3.1: PICkit 2.0 programmer pin connections	13
Figure 4.1: Support circuitry for the PIC	16
Figure 4.2: Delay produced by varying X values in a for-loop	20
Figure 4.3: Serial connection between the PIC and the Bluetooth module	21
Figure 4.4: Ideal (a) and corrupted (b) data during a serial transmission	21
Figure 5.1: 1 Hz sine wave sampled at 37 Hz	26
Figure 5.2: 1 Hz sine wave sampled at 130 Hz	27
Figure 5.3: Fourier Transform of a 1 Hz sine wave sampled at 130 Hz	28
Figure 5.4: PPG waveform sampled at 37 Hz	29
Figure A.1: PACS compared to the conventional method of processing images	32
Figure A.2: Block diagram of the Edward Barnes G. et al. monitoring system	33
Figure A.3: Pin structure of the PIC18F2620 microcontroller	33
Figure A.4: Pin structure of the PIC18F4620 microcontroller	34
Figure A.5: PICkit 2.0 microcontroller programmer	34
Figure A.6: 7-Segment LED display setup	35
Figure A.7: eb301 Bluetooth module used for the NIHMS	35
Figure A.8: Final Design Prototype	36

Nomenclature

NIHMS: Non-Invasive Health Monitoring System.

Microcontroller: standalone programmable microprocessor.

PIC: Refers specifically to microcontrollers developed by Microchip Technology Inc.

Bluetooth: 2.4 GHz wireless technology with standardized implementation and protocol.

Host computer: Computer which receives and processes the signal from the NIHMS.

Chapter 1

Introduction

1.1 Objective

In today's hospitals and research laboratories there are a large number of biomedical devices that have become integral to daily practice. This includes, but is in no way limited to, pulse-oximeters, imaging devices, blood pressure monitors, and various forms of human-computer interfaces (HCI). Generally these devices are very expensive and consequently only available to institutions with adequate resources. With the NIMHS we attempt to provide a cost effective alternative to various commonly used devices in such facilities.

The growing implementation of technology in the field of medicine has had significant impact on its methods and practice. This integration has changed the face of medicine in many ways. One technology with such an impact is the ever growing use of wireless systems in hospitals and other biomedical institutions. Local Area Networks (LANs) are becoming common in hospitals and integrative hand held devices such as PDAs are now a necessity for doctors and other medical personnel. Keeping that in mind, the core objective for this device is to acquire various biological signals and make them available wirelessly over Bluetooth. From there the signal can easily be transmitted to other wireless systems or LANs. This would allow for convenient and round-the-clock monitoring of patients or test subjects from essentially anywhere internet access is possible. Furthermore, due to the device's wireless capabilities, certain costs are significantly reduced by making use of pre-existing displays and processing power on other wireless devices.

1.2 Methodology

The purpose of this device is to measure various biological signals and transmit them via Bluetooth to a computer for processing. Figure 1.1 illustrates the overall concept of this project. Transducers for determining heart rate, oxygen saturation, blood pressure, breathing rate and EOG (electro-oculography) are designed by other members of the group. This report focuses on the remaining parts of the project. This can be subdivided into three major design categories: microcontroller design, Bluetooth link with computer, and host computer application. These categories have both hardware and software components depending on their individual function which are detailed in latter parts of the report.

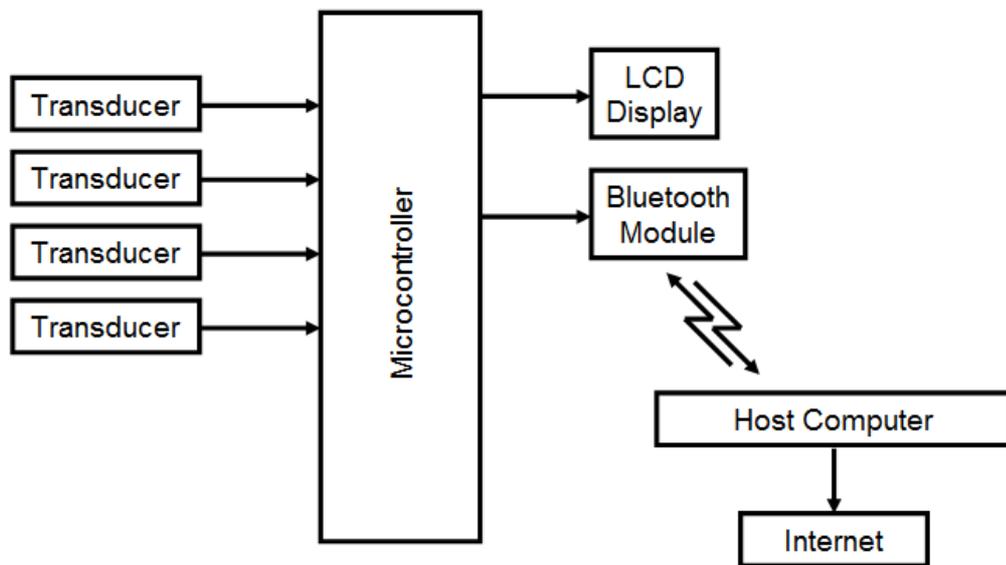


Figure 1.1: Conceptual block diagram of the NIHMS.

The microcontroller directly receives an analog signal from each of the transducers which are assigned their own separate input ports. These ports are configured to direct the signal into the analog-to-digital converter. From there the digital signal is transmitted to the Bluetooth module, which is an independent circuit from the microcontroller but is controlled by it through ASCII commands. Through the Bluetooth module, a connection with the host computer is established and then information is

transmitted over the wireless channel. The host computer, which is Bluetooth enabled, receives the information through an onboard application which transmits the data over TCP/IP. The data can then be accessed by any computer with available internet access.

Along with diagnostic applications, human-computer interface (HCI) was one of the applications considered when designing the project. Though it did not materialize in the final design (due to unforeseen issues and time limitations) it remains an area of interest for this project. The EOG signal received by the device can be processed to determine the motion of the eyes (horizontal and vertical). This can be translated into cursor motion for a computer, or control of other devices such as powered wheelchairs for individuals suffering from severe paralysis (with the exception of their eyes).

1.3 Scope

Since the NIHMS is meant to be a multipurpose device, it has a broad scope when it comes to design criteria as well as application. NIHMS collects physiological data from the subject, which can be used in hospitals for monitoring or diagnosis. Alternatively the device can also come in handy in laboratories where experiments require real-time monitoring of certain subject characteristics (such as heart rate, oxygen saturation and blood pressure). The device can even be used as a personal monitoring system for individuals interested in monitoring their condition, or that of a loved one. Its wireless capabilities allow the subject a great degree of mobility. This is very useful for experiments that require the subject to move or perform certain actions and convenient for patients that require long term monitoring.

In order to ensure that the device was versatile enough for the described applications, a number of design criteria were considered. In some cases the device could be used to measure the critical health of a patient, which requires it to maintain a stable connection over its wireless range. This is one of the reasons Bluetooth was the method of choice for the NIHMS. Its operating frequency and protocol, details of which are discussed in latter chapters, make it a very reliable choice. Through this we achieved a

stable signal over a 9 meter range from the device (indoors). However, that can be easily increased to a much greater range by simply replacing the current Bluetooth module with its external antenna equivalent (sold by the same manufacturer).

By using Bluetooth technology, the choice for what can be classified a host computer for this device is broadened multiple times over. Theoretically any device that is Bluetooth enabled can be used as a host computer. This includes cells phones, PDAs, laptops even devices such as Bluetooth headsets. The only limiting factor is that depending on the application, the host device might not have sufficient processing power or display capabilities. Using a host device eliminates the need for the NIHMS to have an onboard display system (reducing costs and complexity) and all its remaining processing power can be directed towards receiving a higher resolution signal from the transducers. Furthermore, since Bluetooth devices are readily found in homes, industries and hospitals alike, the NIHMS has an edge in compatibility when compared to other radio frequency device (such as the BioCapture¹ from Cleveland Medical Devices Inc.).

¹ http://www.clevemed.com/products_research/prd_res_biocapture_overview.shtml

Chapter 2

Literature Review

The three major areas of research investigated for the purpose of this project are microcontroller design, Bluetooth technology and wireless technology as it applies to medicine. This research played a significant role in determining which parts were eventually chosen for the NIHMS design, since there are a wide range of microcontrollers and Bluetooth modules available. The following subsections outline the relevant literature that played a determining role in the final design.

2.1 Microcontroller Design

The basic idea for how this device would function was adopted from Cosmanescu A. et al. who demonstrated the integration of a PIC microcontroller (PIC18LF2220) with a WBT42 Bluetooth module [1]. Their device (shown in Figure 2.1) was designed to obtain and transmit signals from electromyography (EMG), electroencephalography (EEG) and electrocardiography (ECG). Though the NIHMS measures none of these signals, the nature and frequency range of signals obtained from our transducers is very similar. Therefore picking a microcontroller that could at the very least match the capabilities of the PIC18LF2220 (used by Cosmanescu A. et al.) ensured that the NIHMS would have sufficient processing power. They outlined the following characteristics that were important in their choice for the microcontroller: operates on a 3.3V power level, in-circuit programmable firmware, up to 40 MHz clock speed, 10-bit analog-to-digital converter (ADC) channels and an onboard Universal Synchronous/Asynchronous Receiver/Transmitter (USART) module [1].

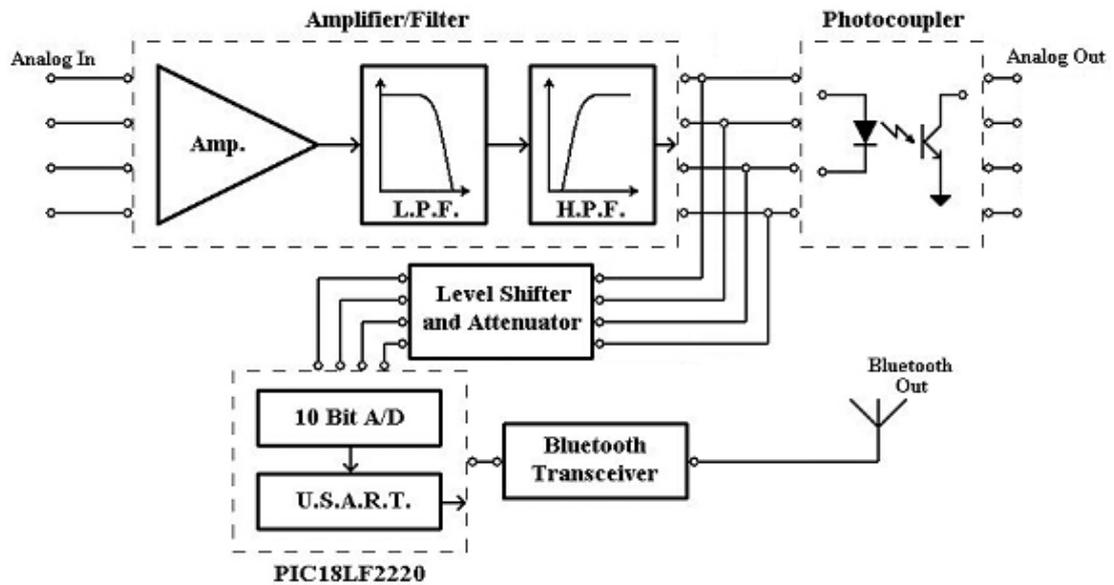


Figure 2.1: Cosmanescu A. et al. design block diagram [1].

In their design, the signal is first filtered, attenuated and level shifted to ensure that it is within a 3V range (the operating voltage for their device is 3.3V) and biased positive with respect to ground [1]. Although this report is not concerned with these functions (they are covered by other group members for their respective transducers) they are a determining factor in proper signal acquisition at the microcontroller. Even though the signal can be filtered digitally, analog filtering and attenuation minimize the error in the converted digital signal. Furthermore if the signal contains negative components, it can produce large errors in the signal when converted by the microcontroller. This phenomenon is discussed in detail in the design chapter of this report (chapter 4.6). Once the signal is filtered, it is converted by the 10-bit ADC channels and passed to the USART as a 10-bit binary number. The signal is then sent serially to the Bluetooth module for wireless transmission. The overall sampling rate of the instrument is governed largely by the bit rate of the USART. The bit rate is the number of bits that are transmitted serially by the USART. An increase in the bit rate of the serial transmission directly corresponds to an increase in the sampling rate, and consequently a better signal resolution.

Along with the hardware design and functionality, Cosmanescu A. et al. also detail the software algorithm used by the microcontroller. Their firmware is written in Assembly. Although more difficult to program in when compared to other programming options (such as C), Assembly gives the programmer a large degree of control over the microcontroller. As per their algorithm, the microcontroller first configures the ADC characteristics on power on. It then waits for the instruction to begin data acquisition. Once that instruction is given, the process of receiving, converting and transmitting the data begin. The microcontroller samples four ADC channels, one after another. After each channel is sampled, its 2 byte binary equivalent is transmitted over Bluetooth, while the next channel is being sampled. The process repeats in this fashion continuously (after the fourth channel is sampled the program goes back to channel one) until the stop instruction is given. Their Bluetooth module, upon connecting with the host computer, appears as a virtual RS232 port, from which the transmitted data is read. The NIHMS uses the same method for receiving data on the host device. Cosmanescu A. et al. chose MATLAB to receive and process their data. For our purposes a stand alone application is more desirable since it gives us more complete control over how the data is utilized. Furthermore it eliminates the need to have MATLAB on the host device.

Edward Barnes G. et al. has also developed a device that is similar to the NIHMS in functionality [3]. Their implementation also revolves around a Bluetooth enabled device. Their article lacks significant technical details regarding their device; however, the idea to make the information provide by the NIHMS available over the internet is credited partially to them. Figure A.2 in Appendix A illustrates their design.

2.2 *Bluetooth Technology*

Over the years, Bluetooth has become the standard for short range communication. It is preferred over other forms of short range wireless transmission methods due to its lower power consumption, lower costs and greater signal stability [2]. The technology was originally developed by Ericsson (a cell phone manufacturer), who later joined with Intel Corporation, International Business Machines Corporation (IBM),

Nokia Corporation, Toshiba Corporation, 3Com Corporation, Lucent/Agere Technologies Inc., Microsoft Corporation and Motorola Inc [2]. Together they formed the Special Interest Group (SIG). Due to this overwhelming association and collaboration, Bluetooth quickly became an open standard and allowed for acceptance and compatibility between devices made by entirely unrelated corporations. It is now supported by over 2100 companies worldwide [2]. In this regard the NIHMS gains a great deal of versatility and compatibility with third party devices.

SIG developed certain specifications to govern the communication between Bluetooth devices. Protocols for information transfer were divided into three logical categories: transport protocol group, the middleware protocol group, and the application group [2]. Figure 2.2 illustrates these groups and their relation with each other. The transport protocol group is responsible for locating other Bluetooth devices and managing the physical synchronous or asynchronous link between them. The NIHMS is concerned mainly with establishing an asynchronous Bluetooth link in this protocol group. The middleware protocol group contains standardized and specific third part protocols to allow applications to receive and send data. With regards to the NIHMS, this protocol group will create a virtual serial port on the host computer through which the transmitted data will be accessed (as if the device was physically connected through a RS232 connection). Lastly the application group consists of the actual application that processes the acquired information, which will be developed specifically for the NIHMS.

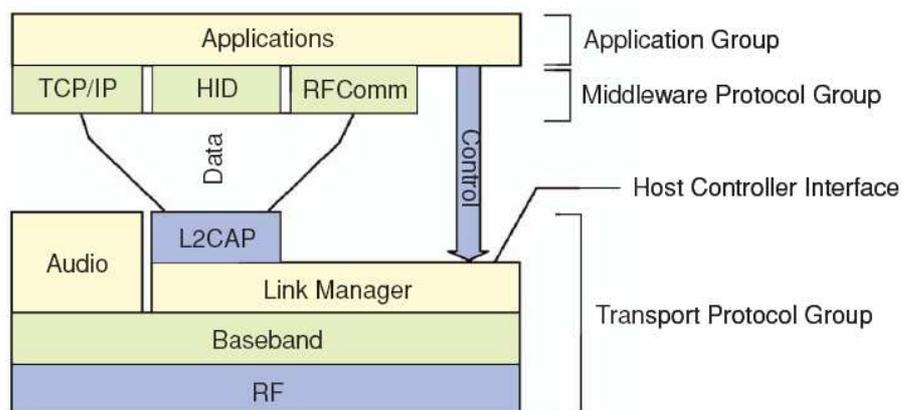


Figure 2.2: Core Bluetooth protocol groups [2].

2.3 *Wireless Technology in Medicine*

In the past two decades, the field of medicine has undergone many changes due to advancements in computer technology. With the introduction of new devices and techniques patient care has improved significantly and that is reflected in the growing life expectancy in the developed nations (life expectancy in Canada now stands at 80.4 years²). Wireless technology is one of the major contributors to the increasing efficiency and reliability of the healthcare system. One such example is the remote monitoring of blood sugar levels for patients suffering from diabetes. Patients can now use their cell phones to enter their measured blood sugar levels, food intake, physical activity levels and medications. The cell phone then informs the patients about their health and transmits that information to a doctor for review³. This not only makes it easier for doctors to access patient information but also provides the patient with the assurance that they are being looked after. Devices such as the NIHMS are also being used for continuous health monitoring and diagnosis. An example of this is the Polysomnographic Diagnostic System which was successfully used to examine causes of sudden infant death syndrome (SIDS) [4].

Wireless technology also expands the network of information systems present in hospitals. Picture Archiving and Communication Systems (PACS) is an example of such a system. With PACS, diagnostic images taken from MRI, CT Scan, PET/CT, Ultrasound or a number of other imaging modalities, can be uploaded to universal servers, from where it can be accessed from anyone in any hospital connected to the network⁴. This increases the ease and speed with which a diagnosis can be made, and also eliminates the problems associated with previous methods of storing images (hard copies which could be lost or misplaced). Though PACS is not necessarily a wireless system (certain parts of it are), it is a great example of how networks can improve medical practice. Figure A.1 in

² Life expectancy hits 80.4 years: Statistics Canada, CBC News, Jan. 14th 2008, Retrieved: Apr. 24th 2009 from: <http://www.cbc.ca/health/story/2008/01/14/death-stats.html>

³ Cell Phones Manage Diabetes, ScienceDaily, Sep. 1st 2008, Retrieved: Apr. 24th 2009 from: http://www.sciencedaily.com/videos/2008/0902-cell_phones_manage_diabetes.htm

⁴ Diagnostic Imaging Services, St. Joseph Hospital, Retrieved: Apr. 24th 2009 from: <http://www.stjosephhospital.com/Diagnostic-Imaging-Services>

Appendix A illustrates how PACS has improved on the conventional method of storing patient images.

Hospitals are now looking to increase the use of wireless technology and networks in their facilities. A 2004 survey of 34 health organizations in the United States of America showed that over 80 percent of hospitals either already have, or plan on implementing wireless Local Area Networks (LANs) in the next 12 months. Another survey published in February of 2005 showed that 79 percent of 253 healthcare executives will use wireless systems and 54 percent will use handheld devices in their daily practice⁵. Since the NIHMS is compatible with both LANs (through a TCP/IP server) and handheld devices (through Bluetooth), it can be a useful tool in this growing market of wireless medical technology.

⁵ Havenstein H. Industry Focus: Wireless in Health Care, TECHWORLD, June 21st 2005, Retrieved Apr 24th 2009, from: <http://www.techworld.com/mobility/features/index.cfm?featureid=1469>

Chapter 3

Statement of Problem and Methodology of Solution

This chapter will outline the technical objective of the project and provide the theoretical and technical knowledge related to the development of the final design for the NINMS. Tools and programs essential in developing the final solution are also discussed to give the reader practical knowledge regarding the NIHMS design process (detailed in the following chapters).

3.1 Problem Statement

For simplicity the design objective for this report can be broken down into three categories: microcontroller design, Bluetooth link and host computer application. The microcontroller design consists of both hardware and software components. Though microcontrollers are designed to operate with minimal circuitry, it is still required to ensure proper functionality, and in certain cases, to acquire specific operational characteristics (such as higher clock speeds). The main hardware challenge is to develop the support circuitry to minimize interruptions from voltage source fluctuations and to achieve the maximum possible clock speed (to ensure that enough processing power is available). The software component is a much larger part of the microcontroller design than the hardware. The software that will be installed on the microcontroller will have two tasks. First, it must configure the analog-to-digital converter and then carry out the conversion process for all the input channels. This will convert the data from an analog signal into a 10-bit binary number. Secondly, the software will be responsible for communicating with the Bluetooth module. Like the ADC, the Bluetooth module will also have to be configured through the software. Once a wireless link is established with

the host computer, the software will need to transmit the acquire data from multiple channels (at least 4). Additionally the data will also need to be time-stamped so that information such as heart rate and breathing rate can be calculated.

The second part of the problem is establishing the Bluetooth link between the computer and the NIHMS. This problem is more complicated than it might originally sound. The Bluetooth module communicates with the microcontroller through the USART port, which enables a serial connection between the two. The microcontroller can then control the Bluetooth module by sending it ASCII commands. In case the module does not perform the desired action, it is difficult to measure or determine the exact response of the Bluetooth module (due to the limited display capabilities of the microcontroller). This problem is further complicated by the fact that the user is limited to the amount of control provided by the module manufacturer.

The last part of the problem is development of a stand alone application that will receive the transmitted information. The application, at the very least, should be capable of receiving the data and then splitting it into multiple data files (one for each transducer), corresponding to the number of input channels. The application might also be required to initiate the Bluetooth connection, or control the module. Beyond that, the potential features that can be incorporated into the application are endless. The data can be processed to provide the subject with a health status or it can be transmitted to a healthcare personnel for analysis. The application can even implement a HCI as discussed earlier in the report.

3.2 Solution Methodology

This section will outline certain concepts and tools that are important for a complete understanding of the design process. The section also introduces the specific components used in the NIHMS's microcontroller and Bluetooth module design for the first time. The main components used in the design processes are as follows: Microchip's PIC18F2620 and PIC18F4620 microcontrollers, Microchip's PICkit 2.0 programmer and

software, and Microchip's MPLAB development environment. The Bluetooth module (A7 Engineering's eb301) is also a major component; however its specifications will be discussed in the following chapter.

The PIC18F2620 and PIC18F4620 are both flash programmable microcontrollers. PIC18F2620 has 28 pins, and the PIC18F4620 is its 40-pin equivalent. They are illustrated in Figure A.3 and Figure A.4 respectively (in Appendix A). They both have the same processor architecture, with the only notable difference being that the PIC18F4620 has more ports for input or output. The microcontroller can be programmed through a 6-pin connection with the PICkit 2.0 programmer (shown in Figure A.5 in Appendix A). This connection between the PIC and the programmer is illustrated by Figure 3.1. The pins listed in the figure are present on all compatible Microchip microcontrollers (including the 2620 and 4620 PICs). Pin 1 is used to reset the PIC or to trigger its programming mode, and pins 2 and 3 can be used as a voltage source to power it. The actual programming is done through pins 4 and 5. Pin 6 does not seem to serve any purpose and can be disconnected without any consequences.

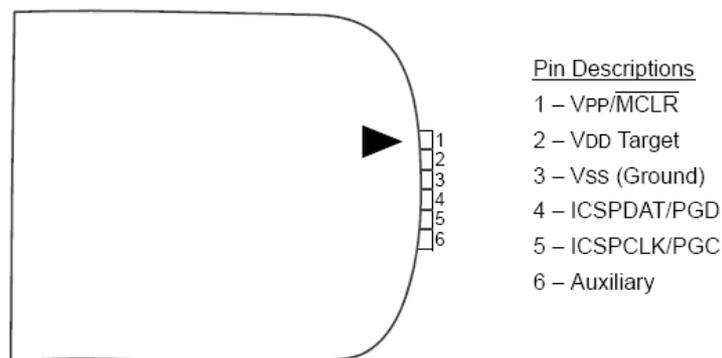


Figure 3.1: PICkit 2.0 programmer pin connections⁶.

In order to program the microcontroller, the program is first written in the MPLAB development environment. The user has the choice of either programming in Assembly or C. Assembly is more difficult and less intuitive to program in when

⁶ ww1.microchip.com/downloads/en/DeviceDoc/51553E.pdf

compared to C; however, it provides the user with complete control over the PIC. For certain cases where the application is time sensitive the programming has to be done in Assembly. For the NIHMS, however, the programs can be written in either language. Once a program is written, it is compiled by MPLAB into a *.hex file. This file is loaded into the software provided with the PICKit 2.0 programmer which transfers the hex information onto the flash memory of the microcontroller. The PIC will then begin executing the program immediately upon power up.

Chapter 4

Experimental and Design Procedures

In this chapter, a detailed account of the design procedures and experiments to test for proper functionality are given. A logical order is followed in which procedures and experiments are introduced in generally the same order in which they were performed. Components and design changes are introduced as they occur in the procedure. For a complete list of the components used for this project please refer to Table B.1 in Appendix B. There are a large number of microcontrollers available from different manufactures; however, Microchip's PIC was the processor of choice due to the large amount of literature available regarding its use.

4.1 *Microcontroller Support Circuitry*

The initial design was adopted from Dr. Reese's course website at Mississippi State University and minor modifications were made⁷. The website contains extensive information on the PIC18 architecture, along with lecture notes and laboratory procedures. The circuit diagram for the modified design is shown in Figure 4.1. Originally the PIC18F2620 was used due to its smaller size. A 9V power supply is sent through a 5V voltage regulator (7805) to provide a stable DC voltage. The 10 μ F capacitors help to eliminate any high frequency AC voltage. The output from the 7805 regulator is approximately 4.83V, not 5V. The significance of this value has to do with the A/D conversion process and will be discussed later in the report. An LED followed by a resistor is connected to the output of the voltage regulator as an on/off indicator for the

⁷ R. Reese. ECE 3724 - Microprocessors. Department of Electrical and Computer Engineering. Retrieved Nov 20, 2008, from: http://www.ece.msstate.edu/courses/ece3724/index_pic18.html.

microcontroller. The output node from the regulator is also shared by the Vdd pin on the microcontroller which is connected to the Vss pin (ground) through a 0.1 μ F decoupling capacitor. The same node is also connected to the Vpp pin through a 10k Ω resistor in series with a LED (a simple diode is sufficient) along with a switch that connects to ground to form the reset circuit. Through this the microcontroller can be reset by bringing Vpp down to ground. A crystal, along with two 15pF capacitors leading to ground, is connected to the Osc1 and Osc2 pins of the microcontroller to create a 40 MHz oscillator.

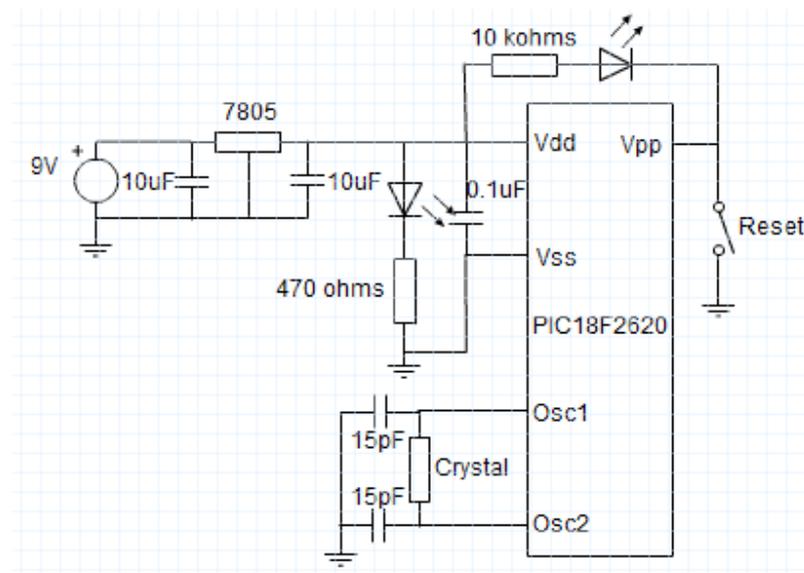


Figure 4.1: Support circuitry for the PIC.

Several tests were performed on the microcontroller to ensure proper functionality by connecting LEDs to the output ports to indicate the result of different programs. Initially the microcontroller did not execute even the simplest of programs. After research and debugging it was discovered that the MPLAB IDE used to write and compile the programs was not assigning the configuration bits to the microcontroller properly. These bits are used to customize the function of the PIC as desired by the user. They control aspects like clock speed, on/off states of certain timers and the device's operation mode, to name a few. The configuration bits were set manually, as part of the code, to resolve this issue. The PIC now executed programs without any problem. Simple programs for

blinking LEDs at different ports were written to test the digital input and output functionality of the ports.

4.2 *Analog-to-Digital Converter*

Extensive time was spent on the A/D operation of the microcontroller since that is the first step in processing the analog signals from the transducers. The microcontroller's internal voltages, $V_{dd} - V_{ss}$ ($\sim 4.83V - 0V$), were used as reference. 10 LEDs were attached to as many output ports in order to display the 10-bit binary number produced by the ADC. The program written to test the ADC first initialized its characteristics (choosing the desired input port, conversion time, result type, etc). Then, in an endless loop, the program gave the command to begin the conversion process and waited for the result to become available. The result was split into two since the converter produces a 10-bit number and the PIC's ports only have 8-bits available. One port displayed the two most significant bits (MSB) and another port displayed the rest of the bits through the attached LEDs. As long as the PIC was powered, this process repeated indefinitely. A variable voltage source was attached to one of the ADC channels and the voltage was slowly increased starting from zero. Initially the increasing voltage produced an increasing binary number, which was the desired response.

When the supplied voltage to the ADC was taken beyond 3.72V, a critical problem was encountered. All the LEDs turned and remained on for any voltage beyond that point, indicating that the ADC was saturated. Originally the hardware design used a 6V voltage source to power the circuit. It was assumed that 5V would be supplied to V_{dd} through the voltage regulator. However, when the A/D converter saturated at 3.72V instead of 5V (at 3.72V all 10 LEDs were on), it was discovered that the regulator was actually outputting 3.72V. This is a problem because the transducers will be supplying a signal up to 4.5V. The voltage source was increased to 9V in order to supply 4.83V to V_{dd} . Now the LEDs saturated at 4.83V, which is acceptable since the actual signal will not exceed 4.5V.

4.3 Thermometer

After the first transducer (thermistor Wheatstone Bridge to measure temperature) was completed, it was interfaced with the microcontroller to perform a more complete test of the hardware and ADC software design. The analog signal was received at port A and temperature was displayed on a 3 digit 7-segment LED connected to ports B and C (please refer to Figure A.3 for the location of these ports). An equation, determined by the transducer's circuitry, related the measured voltage to the corresponding temperature. Through this the temperature was determined from the result of the ADC. Port B was used to display the individual number, and port C, attached to three 2N3906 transistors, was used as the digit selects. Both ports continuously refreshed the data for all digits. Figure A.6 in Appendix A illustrates a very similar circuit except for a different microcontroller, and a keypad is used as the digit select instead of a port. In reality only one digit is displayed at a time; however, since this process repeats at a very fast rate, all 3 digits appear to the user simultaneously. Table 4.1 shows the results of the temperature transducer in comparison with a digital thermometer. The obtained values were very close to measured ones. The difference was mainly due to the fact that the digital thermometer recorded a value and then displayed that same value until reset. However, our thermometer circuit continuously updated the temperature value, which made it susceptible to artifacts. Overall the thermometer circuit was a success. The PIC firmware used along with the circuit can be found in Appendix C.

Table 4.1: Temperatures measured from different areas on the skin

Temperature from Digital Thermometer (°C)	Corresponding Temperature from Wheatstone Bridge Circuit (°C)
32.0	32.0
33.1	33.3
32.2	32.1
32.8	33.1
32.6	32.7

4.4 *LCD Module*

Since the eb301 Bluetooth module acquired for the NIHMS is operated through ASCII commands, an LCD module was also acquired so that these commands and their response from the Bluetooth module could be displayed. In this regard, it proved to be a valuable debugging tool. The LCD was attached to ports B and C, similar to the LED display. However, unlike the LED, the LCD only required certain commands to display the required characters (represented by binary values provided in the datasheet). This module was removed from the final design of the project in order to create a more power-efficient device.

The setup for the LCD module was straight-forward and outlined in detail in the provided datasheet. Before the module could be used, it had to be configured. This is done by sending the module certain 8-bit values at specific time intervals. These values govern the characteristics of the display such as how many lines will be used, the type of cursor (if present), blinking or not, etc. The only challenge faced in integrating the LCD module was determining how to acquire exact timing delays, since the module is somewhat time sensitive (commands can only be sent at certain time intervals). Pre-existing delay algorithms were implemented without success. These algorithms incorporated a large number of header files, which for some reason could not be found by the compiler, even after their existence in the required folder was confirmed. In order to get around this issue it was decided that empty for-loops with a specific number of repetitions would be used to create the needed delays. Using an oscilloscope, the execution times of various for-loops was determined. An LED would turn on at the beginning of the for-loop and turn off when the loop finished. The duration of the time the LED was on for (determined using an oscilloscope), was the execution time of the for-loop. The results from this experiment can be seen in Figure 4.2. X represents the literal value in the following for-loop delay: `for (I = 0; I <= X; I++)`; For this graph the following equation was determined, which proved very useful when ever a certain delay was required (not just for the LCD module):

$$\text{Delay in microseconds} = 2 * X + 4 \quad (1)$$

or

$$X = (\text{Delay in microseconds} - 4) / 2 \quad (2)$$

The only limitation of this method is that only even numbered delays can be produced as evident by equation 1. The actual data obtained from this experiment can be found in Table B.2 of appendix B.

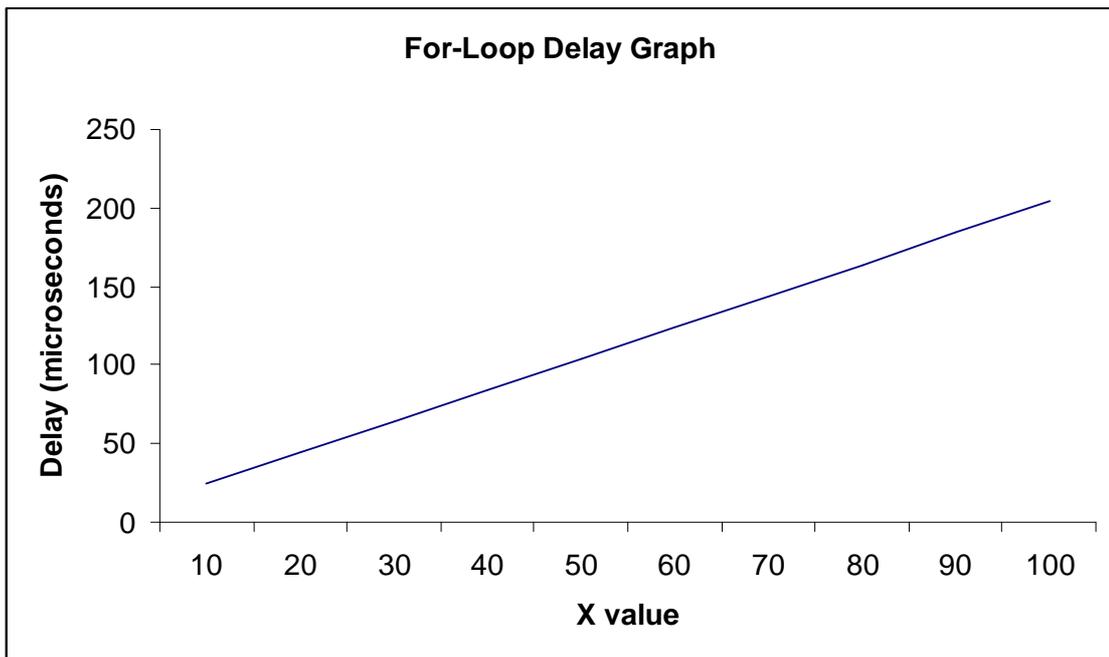


Figure 4.2: Delay produced by varying X values in a for-loop.

4.5 *USART Setup and Bluetooth Connection*

From this point onwards the PIC was changed to its 40-pin equivalent (PIC18F4602) due to a shortage of pins on the PIC18F2620. The second major challenge, after the ADC, was establishing a Bluetooth link with a computer. However, before the link could be established the USART port of the PIC had to be configured. This port was responsible for the asynchronous serial transmission between the PIC and the Bluetooth module (shown in Fig A.7 of Appendix A). Through the USART, ASCII commands can be sent to eb301 module which enables it to perform a number of actions. This includes scanning the area for other Bluetooth devices, connecting to a specific device,

disconnecting, etc. The physical connection between the USART and the eb301 is illustrated by Figure 4.3. Data is transmitted between the two components through two wires. One wire goes from the transmitting end of the PIC to the receiving end of the eb301 and vice versa for the other wire.

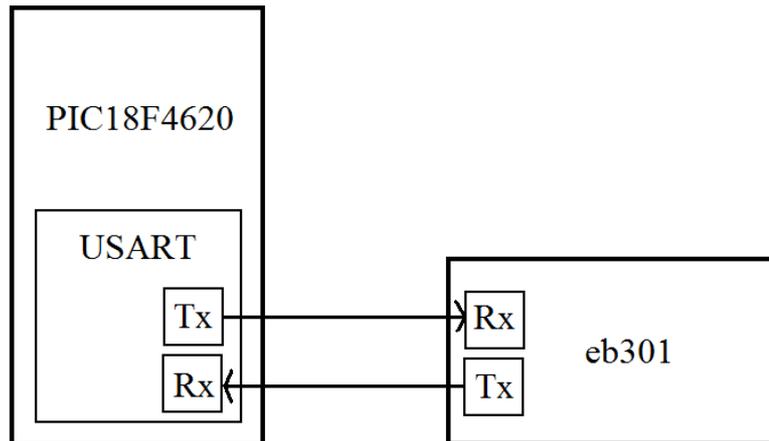


Figure 4.3: Serial connection between the PIC and the Bluetooth module.

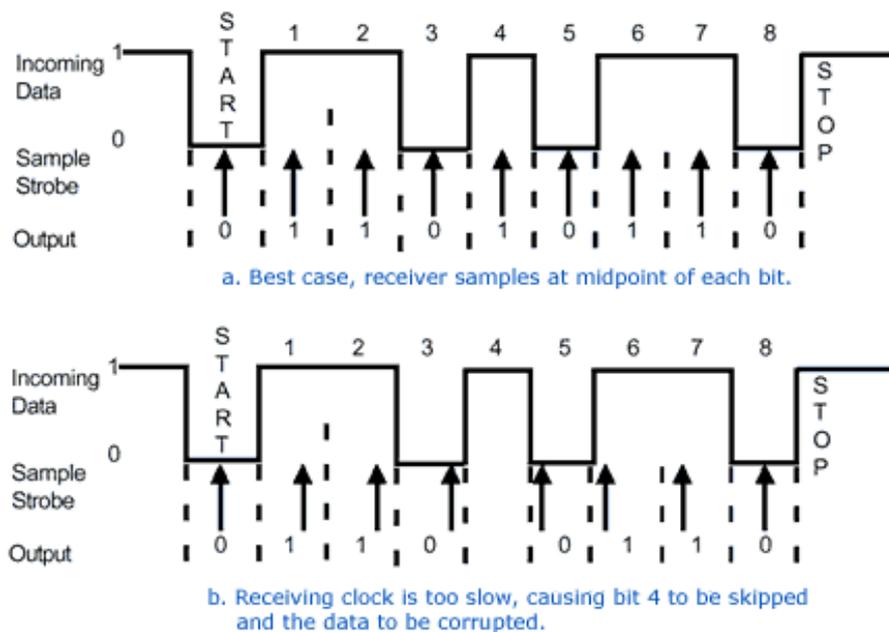


Figure 4.4: Ideal (a) and corrupted (b) data during a serial transmission⁸.

⁸ <http://www.quatech.com/support/figures/asyncl.gif>

Figure 4.4 illustrates the basics of asynchronous serial transmission. A single byte is transmitted bit and bit from the transmitting end. The process is initiated and terminated by a start and stop bit respectively. Upon receiving the start bit, the receiver samples the data at specific intervals governed by its baud rate. Baud rate is a measure of how often the serial channel changes states in one second, and is set by the user in the initial configuration of the device. In order to ensure a proper serial transmission (top diagram of Figure 4.4), both the transmitter and receiver must have the same baud rate associated with them. If the baud rates differ, then the received data is corrupted (bottom diagram of Figure 4.4). In reality baud rates always differ slightly from each other, because of which a certain percent of the data is normally corrupted.

Originally a lengthy program was written that configured and controlled every aspect of the USART setup and execution process as outlined by the PIC's datasheet. The USART was then connected with the Bluetooth module as shown in Figure 4.3. When the command to connect to a Bluetooth device was sent (by supplying the eb301 with the specific Bluetooth address of the computer) nothing happened. The computer could see the module when a scan for Bluetooth devices was performed on it. A connection could be established by initiating the command from the computer itself. However it did not resolve the issue because no data could be sent from the PIC itself. It was determined that the module was working properly and the error was present in the created program. After further investigation of the problem on online forums it was discovered that Microchip had provided pre-made libraries for controlling the USART module. Through the use of these libraries the USART port could be easily configured and ASCII commands could be sent simple by calling the `putsUSART(string)` function. Once the `usart.h` library was implement the same connect command was sent to the Bluetooth module. This time a successful connection was established with the following characteristics: 8-bit, 9600 baud rate and no parity.

For the final design of the firmware, the baud rate was increased to 57600 to increases the sample rate of the device. Furthermore instead of the NIHMS connecting to a Bluetooth device on power it, it was programmed to actively wait for another device to

connect to it. The advantage of this is that now there is no need to determine the Bluetooth address of the host device. It is also more convenient to perform a scan for Bluetooth devices on the host computer, which has a much better display and user interface. This way any device can easily connect to the NIHMS and start receiving data instantly.

4.6 Transducer Integration

Before the Bluetooth link was established, the PPG transducer developed by Yousuf Jawahar was integrated with the microcontroller to ensure that it did not change the signal acquired from the PPG circuit. At this point the analog signal could be converted into binary but could not be transmitted over Bluetooth. Instead the signal was displayed on the LCD as a continuous numerical value, and the analog signal from the transducer was displayed on an oscilloscope. The signal had negative components so an external voltage reference of $-4.5V - 4.5V$ was used instead of internal references (which could not go below zero). When the transducer was connected with the PIC it was discovered that any negative voltage, though within the reference voltage, was attenuated from the signal to 0V. Furthermore, in some cases the negative voltage was also causing the PIC to be reset, a very undesirable result. After discussing the problem with our faculty advisor, Prof T. Doyle, we were advised to shift all transducer signals to an entirely positive range. It is believed that since the microcontroller itself does not operate on a negative voltage range, it is incapable of receiving negative signals. Once the transducer signal was shifted to an entirely positive range (through the use of summing amplifiers), the PIC was able to convert the entire signal without any problems.

Once the Bluetooth link was established, a more complete integration of the transducers was performed. The two transducers for measuring PPG and two channels of EOG (vertical and horizontal) were connected to the PIC. Each channel was sampled one after another. Once all four channels were sampled, the data was composed into one string with the following form: 'B[data 1];[data 2];[data 3];[data 4]'. The data was successfully received and decoded by the host computer application and stored into four

separate files. The results of this transmission are discussed in further detail in the following chapter. The final PIC firmware used for this transmission can be found in Appendix C. A picture of the final design prototype can be found in Appendix A (Figure A.8).

4.7 Host Computer Application

Once the Bluetooth link between the NIHMS and the host computer was established, a C# program was written to receive the transmitted data. Once connected, the eb301 Bluetooth module appeared as a virtual serial port on the host computer. The C# program was successfully able to receive the data from the remote device and save it to a file, as well as displaying it on the screen. Once a successful transmission was made Yousuf Jawahar developed a more complete program in Ruby. The program was able to graph the received data, as well as saving it to a file. Furthermore Yousuf's program opened a TCP/IP channel and transmitted the data over the internet to any client connected to the server (original program on the host computer). Details of this program are available in his project report.

Chapter 5

Results and Discussion

Upon establishing a successful Bluetooth connection between the NIHMS and the eb301 module, a number of tests were conducted to measure the accuracy of the transmitted signal. The purpose of this chapter is to present and discuss these results in detail.

5.1 Transmission At 5.92 kbps

In order to determine the transmission rate, sampling frequency and signal noise characteristics of the NIHMS a 1 Hz sine wave was used as the input signal. Figure 5.1 shows a part of that acquired signal. By averaging the number of samples between each peak of the sine wave (every second since it is oscillating at 1 Hz), it was determined that approximately 37 samples were present for one period of the wave. This translates directly into a sampling frequency of 37 Hz. For each data point in Figure 5.1, a total of 20 bytes are transferred from the NIHMS to the host computer in one packet of data. This is because all four channels of the remote device are actively transmitting their data, and the sine wave is inputted to just one of the channels. Each channel sends a 4 byte data point to the host computer for a total of 16 bytes from all four channels. The data points in each packet are separated by three semicolons, and every packet starts with the character `B` to indicate its beginning. This adds 4 more bytes to the packet for a total of 20 bytes. Therefore the transmission rate can be calculated by the following equation:

$$\begin{aligned}\text{Transmission rate} &= \text{sampling frequency} * \text{number of bytes} * 8 \text{ bits per byte} \\ &= 37 \text{ Hz} * 20 \text{ bytes} * 8 \text{ bits/byte} \\ &= 5920 \text{ bits/s}\end{aligned}$$

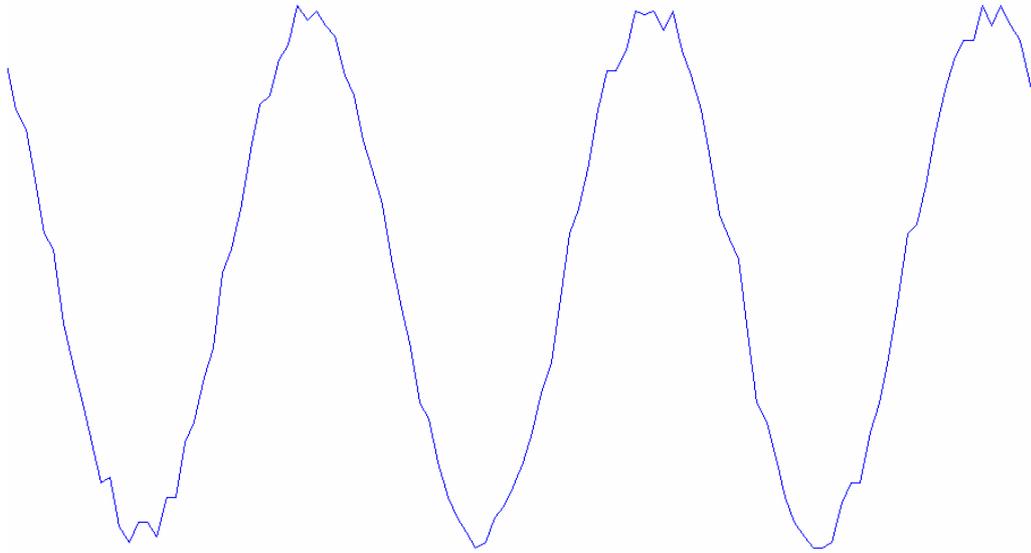


Figure 5.1: 1 Hz sine wave sampled at 37 Hz.

5.2 *Transmission at 20.8 kbps*

Although a transmission rate of 5.92 kbps provides sufficient resolution for determining the desired characteristics of the subject, a higher sampling rate would provide more accurate results. As the sampling rate is increased, the resolution of the acquired also signals increases. As discussed in chapter 2, the most significant determining factor in the sampling rate is the bit rate of the serial transmission [1]. Though the bit rate and baud rate are not identical characteristics of the serial transmission, they are closely related. Generally an increase in baud rate results in a higher bit rate and consequently a higher sampling frequency. In order to increase the acquired signal resolution the baud rate was increased to the maximum possible amount of 57600. Even though both the PIC and the Bluetooth module are capable of operating on higher baud rates, the serial transmission failed at higher values.

Similar to the previous section, a 1 Hz sine wave was transmitted through one of the four active channels. Figure 5.2 is an illustration of a small part of that signal. By averaging the number of samples between each period of the acquired signal the

sampling frequency was determined to be 130 Hz. This is a significant improvement over the previous sampling frequency of 37 Hz. The transmission rate for a baud rate of 57600 can be calculated using the same equation as in the previous section.

$$\begin{aligned}\text{Transmission rate} &= \text{sampling frequency} * \text{number of bytes} * 8 \text{ bits per byte} \\ &= 130 \text{ Hz} * 20 \text{ bytes} * 8 \text{ bits/byte} \\ &= 20.8 \text{ kbps}\end{aligned}$$

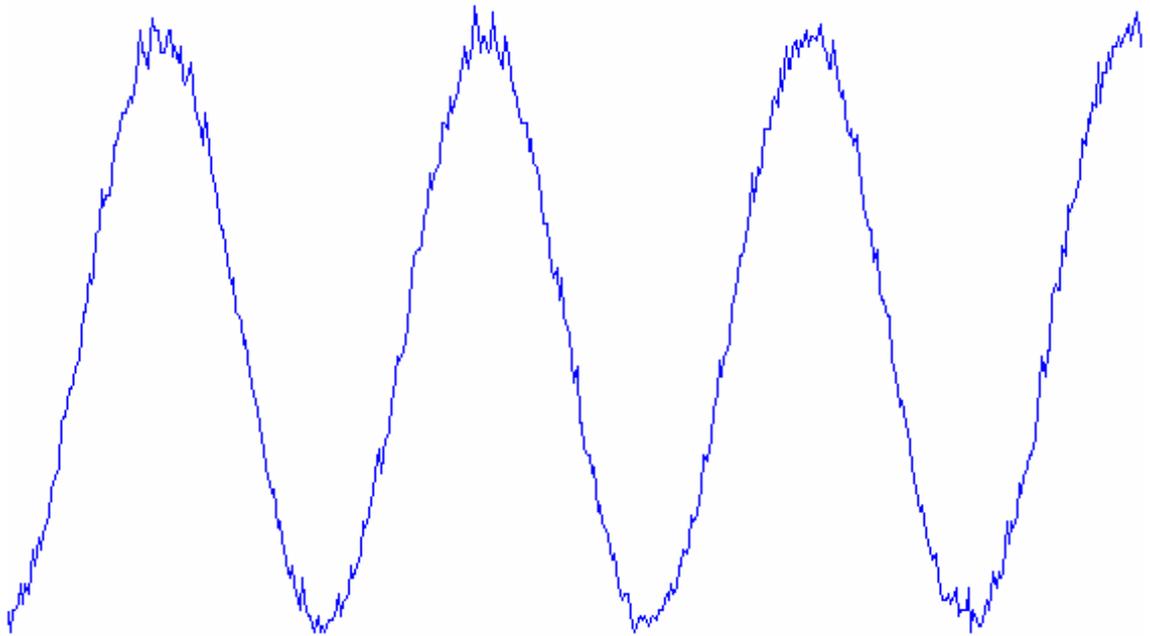


Figure 5.2: 1 Hz sine wave sampled at 130 Hz.

5.3 *Transmitted Signal Characteristics*

When the two figures of the acquire sine waves are compared, the changes are very significant and notable. Because of the higher sampling frequency (over 3.5 times greater than that of Figure 5.1) Figure 5.2 appears much more like a sine wave. Both graphs; however, contain a significant amount of error, which is most notable near the signal peaks. The error is introduced into the signal at the microcontroller, most likely during the analog-digital conversion. The increase in baud rate and sampling frequency seems to have little effect on this error. However at higher the sampling frequency it is

somewhat masked due to the resolution. The Fourier Transform of the signal in Figure 5.2 is shown in Figure 5.3 and it illustrates the significant amount of error present in the acquired signal. The peak at 1 Hz is the sine wave itself; whereas, the rest of the signal is essentially noise. This noise seems to be mostly DC noise. Though it seems to overwhelm the signal, the fact that it is mostly DC has reduced effects on the final signal. Reasons for this noise are still unknown but it is suspected that further calibration of the ADC configuration can eliminate it. If that fails, then the same signal can be acquired at multiple ADC channels and added together. If the noise is Gaussian, then with enough additions of the same signal it will cancel itself out to a large degree.

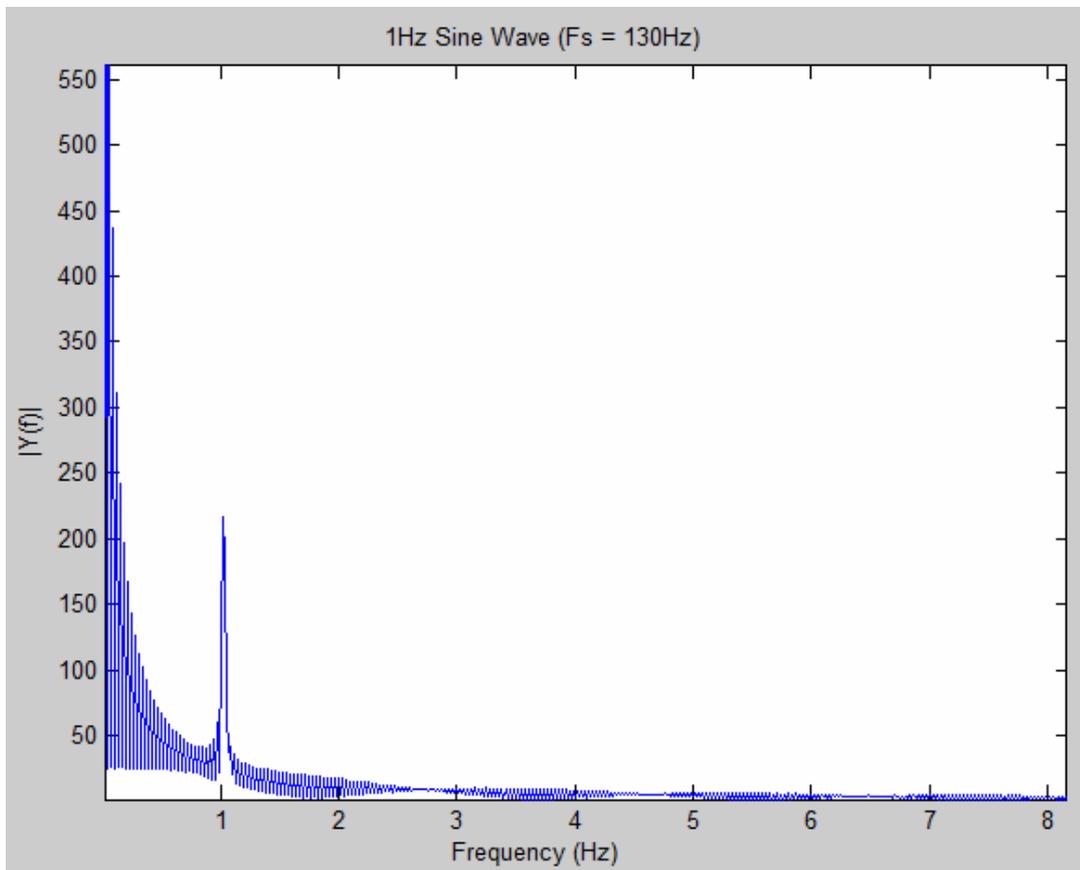


Figure 5.3: Fourier Transform of a 1 Hz sine wave sampled at 130 Hz.

Even with the noise the device is able to produce useful data. Figure 5.4 is an actual PPG measurement sampled at the lowest frequency of 37 Hz, and it represents the

worst possible signal provided by the device. Even at this configuration the heart rate can be easily determined.

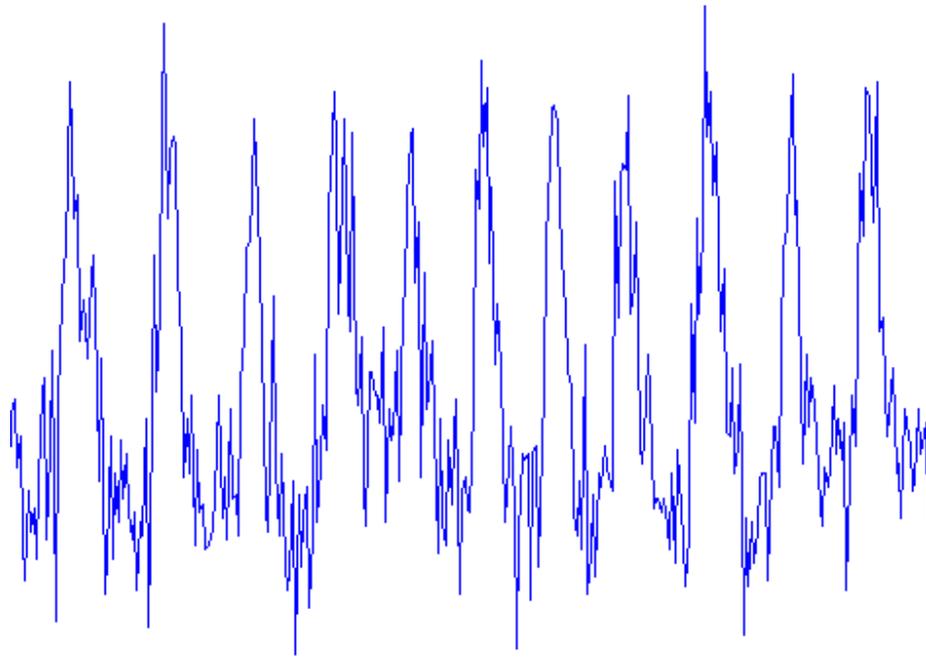


Figure 5.4: PPG waveform sampled at 37 Hz.

Chapter 6

Conclusion and Recommendations

Overall the final design of the device functioned exactly as expected with a reasonable amount of accuracy. Due to time limitations further improvements on the design were not possible. However, the initial goal was to ensure that the device functions and there is at least some data to process no matter how noisy. The acquired signals were processed successfully by other members of the group, for their respective transducers. The acquired values were compared to the ones measured from commercial devices. Though the values were not always accurate, the changes and trends of the signal were preserved even with the given noise. However, when measuring EOG signals through the NIHMS the noise made it difficult to process the signal. This was mainly due to the fact that the EOG signal already possessed a high degree of noise from motion artifact. Because of that reason a Human Computer Interface (HCI) using EOG could not be developed; however, it still remains as a strong area of interest.

The most attractive part of the NIHMS is its Bluetooth capabilities. For reasons discussed in chapter 2, the device gains a great deal of compatibility with virtually any Bluetooth enabled device. Any device that is within range of the NIHMS can receive data from it by simply connecting to it (given that an application is available to receive that data). In case that data needs to be accessed by multiple users, the developed server application can connect to the device through Bluetooth and then transmit the received signal over TCP/IP to a large number of clients. With this functionality the NIHMS fulfills the criteria for a developing wireless technology in the field of medicine, as outlined in chapter 2.

There are a number of recommendations for how to make this device better suited for the medical field. As the NIHMS stands now, it is capable of performing most of the tasks that were proposed when it was being postulated (measuring heart rate, breathing rate, blood pressure, etc.). However, the values obtained by our device can not be used for clinical applications due to the amount of error present in its recording. The following recommendations can help lessen the gap between the developed device and those used commercially for clinical practice. First and foremost, the level of noise in the acquired signal needs to be reduced. This noise originates from both the transducers and the microcontroller. It is recommended that better quality sensors be used to eliminate instrumentation noise. At the microcontroller, further calibration of ADC might help reduce the noise it adds. Digital filters, such as FIR filters, can also be implemented in the microcontroller or the host computer to further reduce the overall noise of the system.

Currently the NIHMS has a minimum range of 9 meters over which it can maintain a stable wireless connection. In some cases this range might not be enough. The choice of the eb301 Bluetooth adapter enables the device to increase its operating range by using the external antenna equivalent module. No software changes are needed, a simple switch of the modules will provide the user with a much larger range. Similarly if a more compact size for the device is desired, it is recommended that the PIC18F2620 microcontroller be used instead of the PIC18F4620. However, unlike the Bluetooth module, this switch will require minor firmware updates. Lastly, the device currently only transmits signals to the host computer, and the firmware does not support transfer of information back to the microcontroller. In the future it is recommended that bilateral communication be implemented to allow the host computer control over the microcontroller. This will eliminate the need for the microcontroller to be re-programmed if any changes are implemented, and add a multitude of new features to the device.

Appendix A: Supplementary Figures

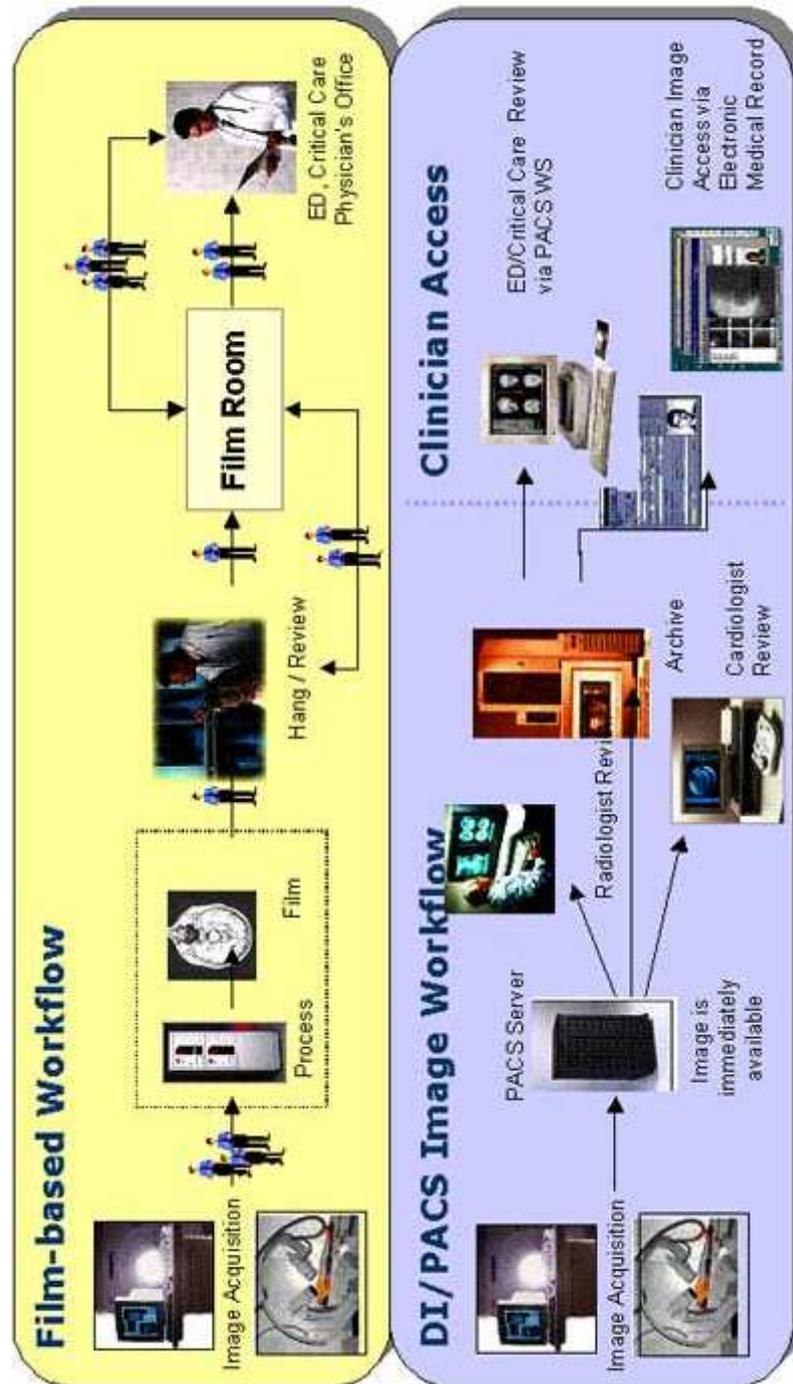


Figure A.1: PACS compared to conventional method of processing and store images⁹.

⁹ http://www.baptist-health.com/_images/photos/bapt_photo_pacs_1.jpg

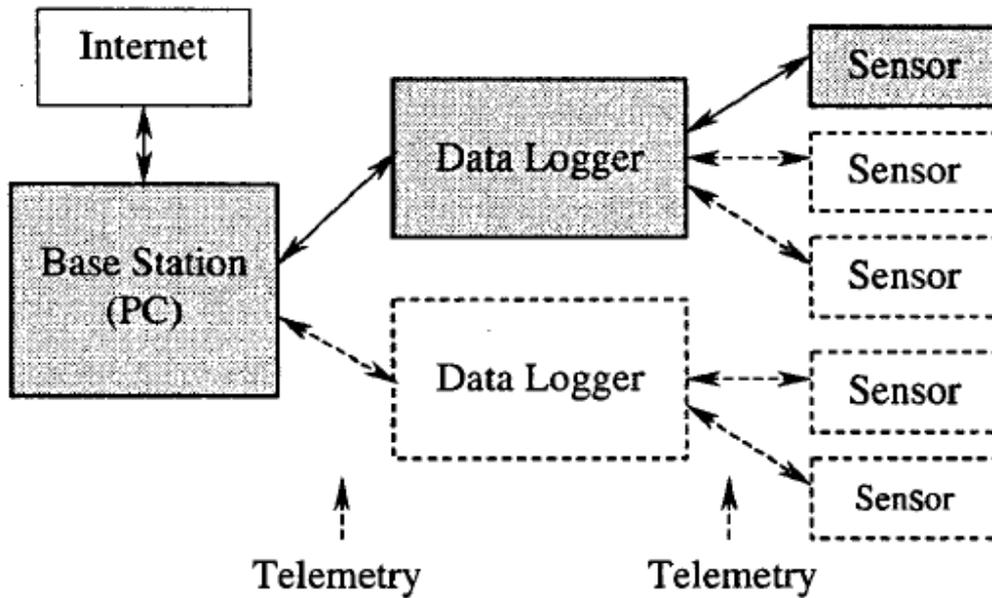


Figure A.2: Block diagram of the Edward Barnes G. et al. monitoring system [3].

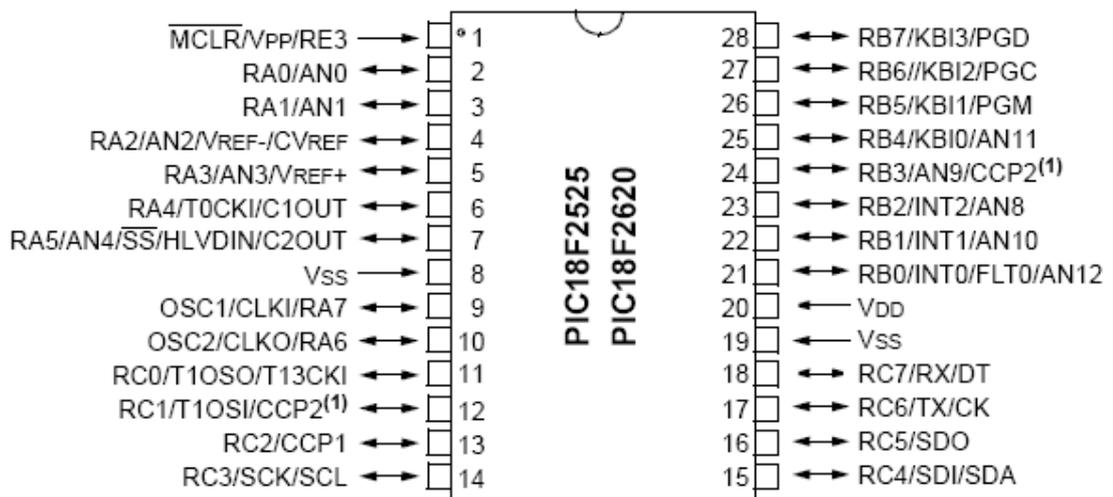


Figure A.3: Pin structure of the PIC18F2620 microcontroller¹⁰.

¹⁰ ww1.microchip.com/downloads/en/DeviceDoc/39626b.pdf

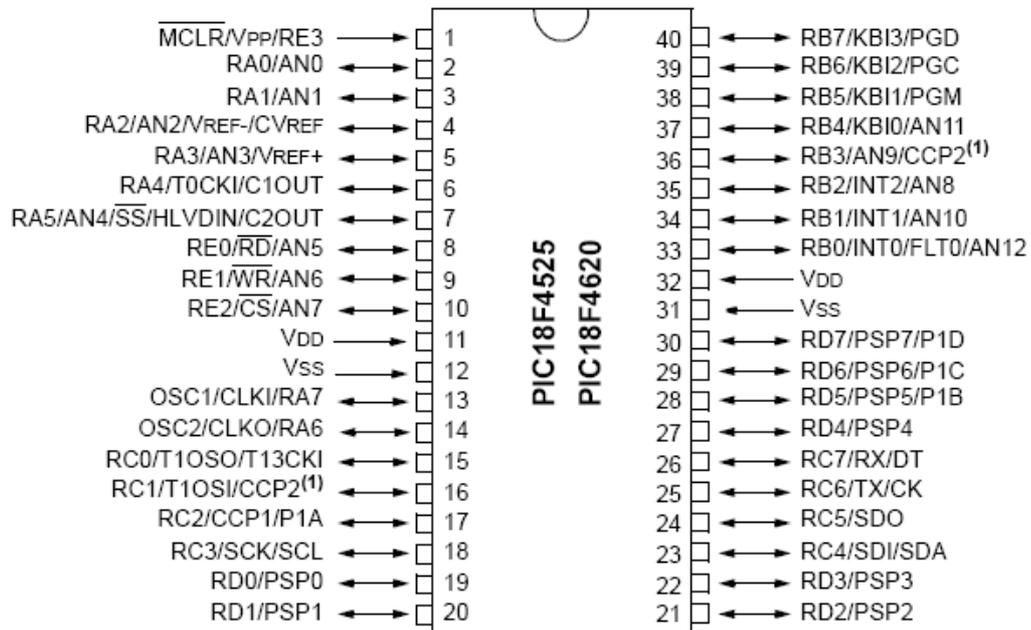


Figure A.4: Pin structure of the PIC18F4620 microcontroller².



Legend:

- | | | |
|-----------------|-------------------------|---------------------------|
| 1 – Status LEDs | 3 – Lanyard Connection | 5 – Pin 1 Marker |
| 2 – Push Button | 4 – USB Port Connection | 6 – Programming Connector |

Figure A.5: PICKIT 2.0 microcontroller programmer¹¹.

¹¹ ww1.microchip.com/downloads/en/DeviceDoc/51553E.pdf

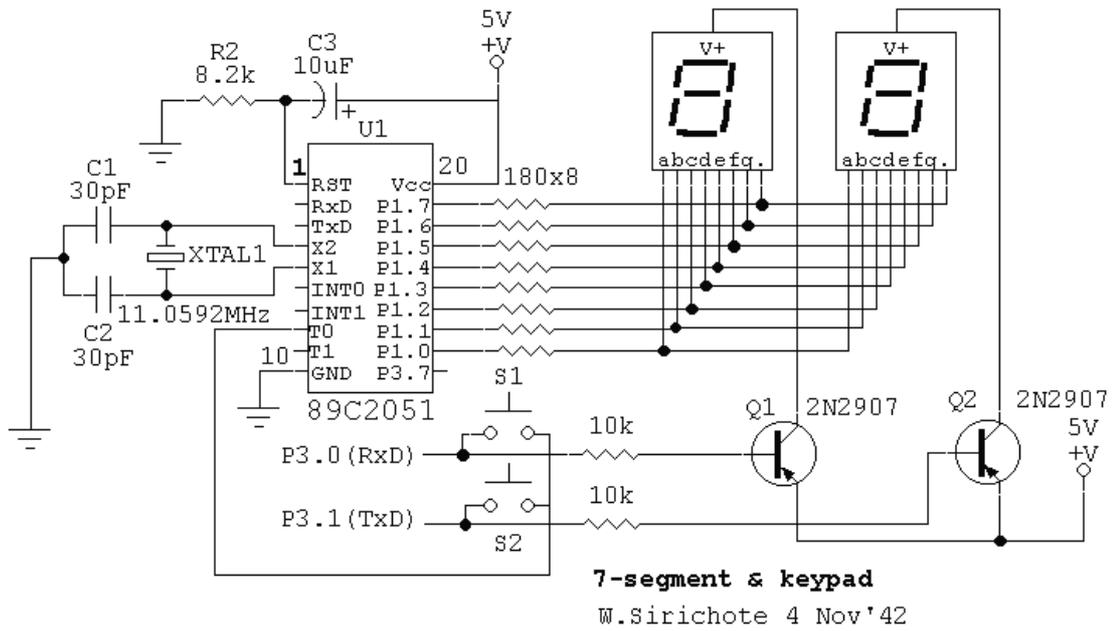


Figure A.6: 7-Segment LED display setup¹².



Figure A.7: eb301 Bluetooth module used for the NIHMS¹³.

¹² http://www.hobbyprojects.com/quick_circuits_reference/microcontroller_circuits/images/7-seg1.gif

¹³ http://www.a7eng.com/images/eb301_bluetooth_serial_adapter_woba.jpg

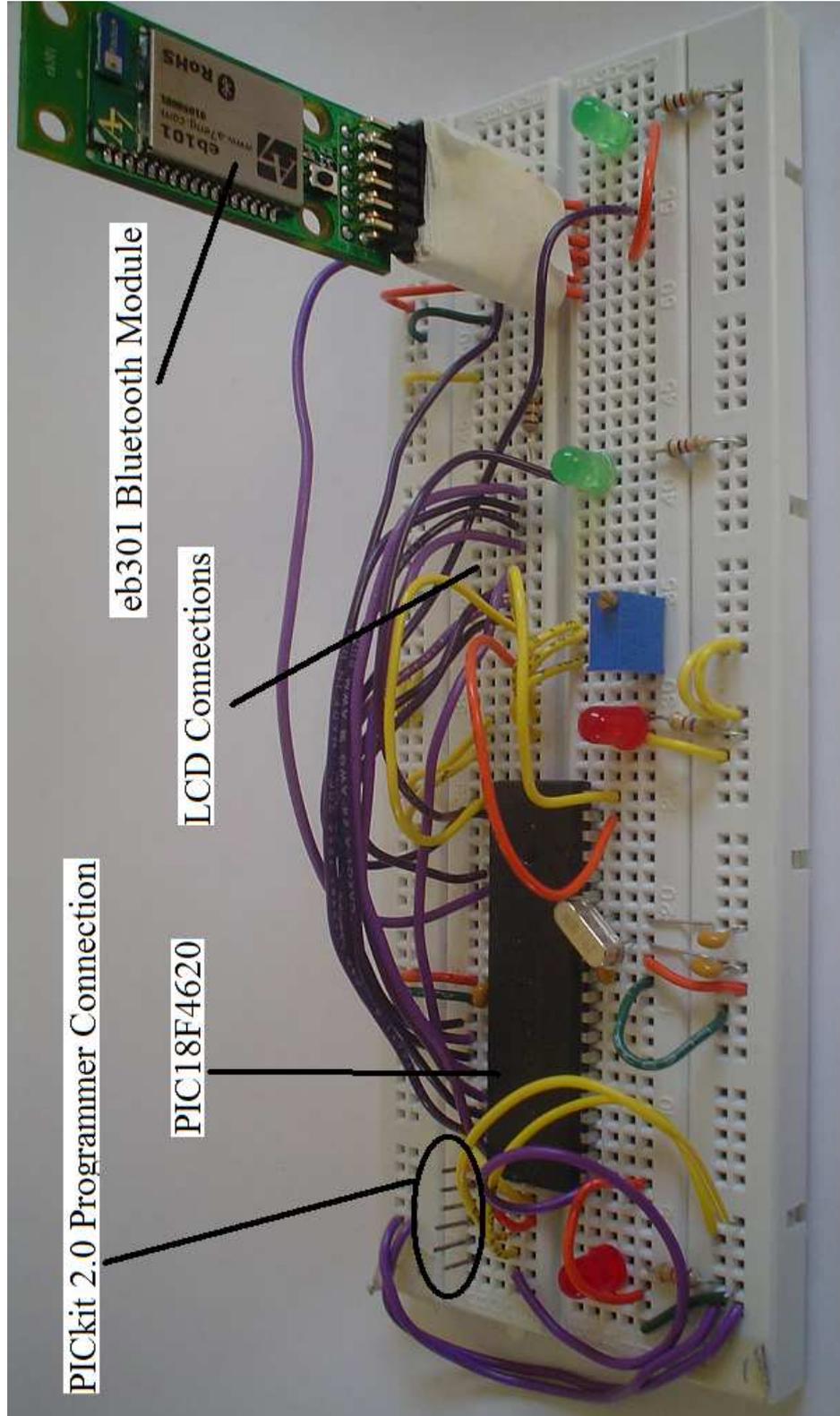


Figure A.8: Final Design Prototype.

Appendix B: Supplementary Tables

Table B.1: Complete list of components used for the project.

Item	Quantity
PIC18F2620 microcontroller	1
PIC18F4620 microcontroller	1
PICkit 2.0 programmer and software	1
LITE-ON's 7-segment 4 digit LED display (LTC-4627JR)	1
Newhaven Display's 12x2 LCD module (NHD-0212WH-AYGH-JT#)	1
A7 Engineering's eb301 Bluetooth Module	1
10 μ F capacitor	2
0.1 μ F capacitor	2
15 pF capacitor	2
470 Ω resistor*	≥ 4
10 k Ω resistor	1
7805 voltage regulator	1
Crystal oscillator	1
LEDs*	≥ 4
2N3906 Transistors	3
Protoboard	1

* - These components were used together and their amount varied in the design process.

Table B.2: Data from the for-loop delay experiment.

X	Delay (μs)	X	Delay (μs)
1	6	40	84
2	8	50	104
3	10	60	124
4	12	70	144
5	14	80	164
10	24	90	184
20	44	100	204
30	64		

Appendix C: PIC Programs

C.1: PIC Firmware for the Thermometer

```
#pragma config OSC = HSPLL, FCMEN=OFF, PWRT = OFF, BOREN = OFF, WDT = OFF, WDTPS =
128, MCLRE=ON, LPT1OSC = OFF, PBADEN=OFF, CCP2MX=PORTC, STVREN=ON, LVP = OFF,
XINST=OFF, DEBUG = OFF
#include <p18f4620.h>
```

```
void main (void) {
    char c[10];
    int i, j, t, x;
    double vin;
    TRISC = 0b00000000;
    TRISB = 0b00000000;
    j = 0xff;
    c[0] = 0b00000011;
    c[1] = 0b10011111;
    c[2] = 0b00100101;
    c[3] = 0b00001101;
    c[4] = 0b10011001;
    c[5] = 0b01001001;
    c[6] = 0b01000001;
    c[7] = 0b00011111;
    c[8] = 0b00000001;
    c[9] = 0b00001001;

    while (1) {
        ADCON1 = 0b00001110;
        ADCON0 = 0b00000000;
        ADCON2 = 0b10100001;
        ADCON0 = 0b00000001;

        for (i = 0; i <= 0xff; i++);

        ADCON0 = 0b00000011;

        while (ADCON0 == 0b00000011);

        vin = 4.764*(((double)ADRESH*2*2*2*2*2*2*2 +
(double)ADRESL)/0b11111111);
        t = (int)((-11.538*vin + 50.731) * 10); // temperature equation

        for (x = 0; x <= 0xFF; x++){
            PORTB = 0b11111111; // first digit
            PORTC = 0b00000110;
            PORTB = c[(t/100)];
            for (i = 0; i <= j; i++){ }

            PORTB = 0b11111111; // second digit
            PORTC = 0b00000101;
            PORTB = c[((t/10)%10)-0b00000001];
            for (i = 0; i <= j; i++){ }
        }
    }
}
```

```

        PORTB = 0b11111111; // third digit
        PORTC = 0b00000011;
        PORTB = c[t%10];
        for (i = 0; i <= j; i++){
            }
        }
    }
}

```

C.2: Final PIC Firmware

```

#include <p18f4620.h>
#include <stdlib.h>
#include <stdio.h>
#include <usart.h>
#define bits_on(var,mask) var |= mask
#define bits_off(var,mask) var &= ~0 ^ mask
#pragma config OSC = HSPLL, FCMEN=OFF, PWRT = OFF, BOREN = OFF, WDT = OFF, WDTPS =
128, MCLRE=ON, LPT1OSC = OFF, PBADEN=OFF, CCP2MX=PORTC, STVREN=ON, LVP = OFF,
XINST=OFF, DEBUG = OFF
char text8[30];
int data[4];
char text9[] = "set baud 57600<\r>";

void main (void){
    int i, j;
    float vin;
    TRISD = 0b0111;
    PORTD = 0b0000;
    // Delay for eb101
    for (i = 0; i < 600; i++){
        for (j = 0; j < 600; j++){
            }

        OpenUSART(    USART_TX_INT_OFF &
                    USART_RX_INT_OFF &
                    USART_ADDEN_OFF &
                    USART_BRGH_LOW &
                    USART_CONT_RX &
                    USART_EIGHT_BIT &
                    USART_ASYNC_MODE, 63);

        putsUSART(text9); // baud rate changed to 57600
        OpenUSART(    USART_TX_INT_OFF &
                    USART_RX_INT_OFF &
                    USART_ADDEN_OFF &
                    USART_BRGH_HIGH &
                    USART_CONT_RX &
                    USART_EIGHT_BIT &
                    USART_ASYNC_MODE, 42);

        bits_on(PORTD, 0b1000); // indicator bit that show active search for Bluetooth connection
        while ((PORTD & 0b0100) == 0b0100); // wait for Bluetooth connection to be established
        bits_off(PORTD, 0b1000);
    }
}

```

```

// ADC initialization
ADCON1 = 0b00001011;
ADCON0 = 0b00000000;
ADCON2 = 0b10001110;
ADCON0 = 0b00000001;

while (1){
    // ADC channel 0
    for (i = 0; i < 1; i++);
    ADCON0 = 0b00000011;
    for (i = 0; i < 1; i++);
    while (ADCON0 == 0b00000011);
    for (i = 0; i < 1; i++);
    vin = 4764*(((float)ADRESH*2*2*2*2*2*2*2 + ADRESL)/0b1111111111);
    data[0] = (int)vin;

    // ADC channel 1
    for (i = 0; i < 1; i++);
    ADCON0 = 0b00000111;
    for (i = 0; i < 1; i++);
    while (ADCON0 == 0b00000111);
    for (i = 0; i < 1; i++);
    vin = 4764*(((float)ADRESH*2*2*2*2*2*2*2 + ADRESL)/0b1111111111);
    data[1] = (int)vin;

    // ADC channel 2
    for (i = 0; i < 1; i++);
    ADCON0 = 0b00001011;
    for (i = 0; i < 1; i++);
    while (ADCON0 == 0b00001011);
    for (i = 0; i < 1; i++);
    vin = 4764*(((float)ADRESH*2*2*2*2*2*2*2 + ADRESL)/0b1111111111);
    data[2] = (int)vin;

    // AD channel 3
    for (i = 0; i < 1; i++);
    ADCON0 = 0b00001111;
    for (i = 0; i < 1; i++);
    while (ADCON0 == 0b00001111);
    for (i = 0; i < 1; i++);
    vin = 4764*(((float)ADRESH*2*2*2*2*2*2*2 + ADRESL)/0b1111111111);
    data[3] = (int)vin;

    // data from all 4 channels transmitted as a single string
    sprintf(text8,"B%04d;%04d;%04d;%04d;",data[0],data[1],data[2],data[3]);
    putsUSART(text8);
}
}

```

References

- [1] Cosmanescu A, Miller B, Magno T, Ahmed A, Kremenec I. *Design and implementation of a wireless (Bluetooth [registered trademark]) four channel bio-instrumentation amplifier and digital data acquisition device with user-selectable gain, frequency, and driven reference*, EMBS Annual International Conference, IEEE Sep. 3, 2006.
- [2] McDermott-Wells P. *What is Bluetooth?*, Potentials, IEEE 2005; 23(5): 33-35.
- [3] Edward BG, Warren S. *A wearable, Bluetooth-Enabled System for Home Health Care*, Proceedings of the Second Joint EMBS/BMES Conference, Compendex Oct. 23-26, 2002.
- [4] Barschdorff D. *Information Processing of Biomedical Sensor Signatures for Ensuring the Quality of Life*, XVIII Imeko World Congress, Sep. 2006.

Vitae

Name: Omer Waseem

Place of Birth: Lahore, Pakistan

Year of Birth: 1986

Secondary Education: Erindale Secondary School
2000-2004

Post-Secondary Education: McMaster University
2004-2009

Work Experience: 1D04 Teaching Assistant
2006-2009
LEAP Biomedical Counsellor
2006-2009