

An Authentic Ecg Simulator

2006

Paul Michalek
University of Central Florida

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

 Part of the [Computer Engineering Commons](#)

STARS Citation

Michalek, Paul, "An Authentic Ecg Simulator" (2006). *Electronic Theses and Dissertations*. 966.
<https://stars.library.ucf.edu/etd/966>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.

AN AUTHENTIC ECG SIMULATOR

by

PAUL J. MICHALEK

Bachelor of Science in Computer Engineering
University of Central Florida, 2006

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Computer Engineering
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2006

ABSTRACT

An ECG (electrocardiogram) simulator is an electronic tool that plays an essential role in the testing, design, and development of ECG monitors and other ECG equipment. Principally an ECG simulator provides ECG monitors with an electrical signal that emulates the human heart's electrical signal so that the monitor can be tested for reliability and important diagnostic capabilities. However, the current portable commercially available ECG simulators are lacking in their ability to fully test ECG monitors. Specifically, the portable simulators presently on the market do not produce authentic ECG signals but rather they endeavor to create the ECG signals mathematically. They even attempt to mathematically create arrhythmias (irregular heartbeats of which there are many different types). Arrhythmia detection is an important capability for any modern ECG monitor because arrhythmias are often the critical link to the diagnosis of heart conditions or cardiovascular disease.

The focus of this thesis is the design and implementation of a portable ECG simulator. The important innovation of this prototype simulator is that it will not create its ECG signals mathematically, but rather it will store ECG data files on a memory module and use this data to produce an authentic ECG signal. The data files will consist of different types of ECG signals including different types of arrhythmias. The data files are obtained via the internet and require formatting and storing onto a memory chip. These files are then processed by a digital to analog converter and output on a four lead network to produce an authentic ECG signal. The system is built around the ultra-low power Texas Instruments MSP430 microcontroller.

TABLE OF CONTENTS

| | |
|--|----|
| LIST OF FIGURES | v |
| LIST OF TABLES | vi |
| CHAPTER ONE: INTRODUCTION..... | 1 |
| ECG Monitors Save Lives | 1 |
| ECG Simulators – Need and Usage..... | 3 |
| Safety Requirements – Solution ECG Simulators | 3 |
| Class I devices..... | 4 |
| Class II devices | 4 |
| Class III devices..... | 5 |
| ECG Monitor Safety Issues | 5 |
| ECG Monitor Performance Testing..... | 6 |
| ECG Monitor Algorithm Development | 8 |
| Thesis Objective..... | 9 |
| Commercially Available Simulators..... | 10 |
| Ferronato - FC3D | 10 |
| Dale EHS10 | 10 |
| Dale13 ECG Simulator | 11 |
| NETECH MiniSim 1000..... | 11 |
| Bio-Tek Lionheart 3..... | 12 |
| MULTI-PRO 2000..... | 12 |
| DELTA 3000 | 13 |
| M311 ECG Simulator | 13 |
| Summary of Commercially Available Simulators..... | 14 |
| The Problem with Current ECG Simulators | 14 |
| Solving the Problem – Prototype Overview | 17 |
| CHAPTER TWO: ECG MONITORS FUNCTIONALITY | 20 |
| The Heart | 20 |
| Heart – Basics | 20 |
| The Electrochemical Perspective of the Heart..... | 21 |
| The Heartbeat Perspective | 25 |
| The Electrocardiogram..... | 25 |
| ECG Waveform Segments..... | 25 |
| What ECG Monitors Do | 28 |
| ECG Stress Test..... | 29 |
| Cardiac Arrhythmias..... | 30 |
| Arrhythmia Detection | 36 |
| Implications for ECG Simulators..... | 38 |
| CHAPTER THREE: PROVIDING AUTHENTIC HEART SIGNALS | 39 |
| The Need for Authentic Signals..... | 39 |
| Physiologic Databases – MIT-BIH..... | 39 |
| MIT-BIH Arrhythmia Databases | 40 |

| | |
|---|-----|
| Creighton University Ventricular Tachyarrhythmia Database | 41 |
| MIT-BIH Noise Stress Test Database..... | 41 |
| MIT-BIH ST Change Database | 42 |
| MIT-BIH Malignant Ventricular Arrhythmia Database..... | 42 |
| MIT-BIH Atrial Fibrillation/Flutter Database | 42 |
| MIT-BIH ECG Compression Test Database | 42 |
| MIT-BIH Supraventricular Arrhythmia Database..... | 43 |
| MIT-BIH Long-Term Database..... | 43 |
| MIT-BIH Normal Sinus Rhythm Database | 43 |
| Using the MIT-BIH Data Records..... | 43 |
| Obtaining the MIT-BIH Data..... | 43 |
| Formatting the MIT-BIH Data Files | 47 |
| Summary of the Use of MIT-BIH Database Files | 50 |
| CHAPTER FOUR: SYSTEM HARDWARE COMPONENTS..... | 51 |
| Meeting the Hardware Requirements | 51 |
| The Functionality of the Prototype | 54 |
| Description of Hardware Components..... | 55 |
| Texas Instruments MSP430F169 Microcontroller..... | 56 |
| Digital to Analog Converter..... | 59 |
| The JTAG InterfaceHardware..... | 60 |
| Samsung NAND Flash Memory..... | 61 |
| The LCD Display..... | 62 |
| CHAPTER FIVE: SYSTEM SOFTWARE..... | 65 |
| The Software Development Tools | 65 |
| System Software Overview..... | 67 |
| Programming the Input Buttons and LCD Display..... | 69 |
| Programming the NAND Flash Memory Module | 72 |
| Programming the DAC12 | 78 |
| The DAC12 Control Register (DAC12_xCTL)..... | 78 |
| The DAC12 Data Register (DAC12_xDAT)..... | 85 |
| Programming the Options for the ECG Signal | 86 |
| CHAPTER SIX: RESULTS AND CONCLUSIONS..... | 89 |
| The Authentic ECG Simulator’s Waveform..... | 89 |
| Strengths of the Authentic ECG Simulator..... | 94 |
| Areas That Require Further Work | 95 |
| APPENDIX: PERMISSION TO USE IMAGES..... | 98 |
| REFERENCES | 106 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1 Comparison of Human ECG and Simulated ECG Signals..... | 16 |
| Figure 2 High-level System Block Diagram..... | 19 |
| Figure 3 Cardiac Action Potential..... | 24 |
| Figure 4 ECG Wave Segment Breakdown | 26 |
| Figure 5 Actual Electrocardiogram output | 29 |
| Figure 6 Specialized Conductive Tissues of Atria and Ventricles | 33 |
| Figure 7 ECG waves of First, Second, and Third degree heart blocks..... | 34 |
| Figure 8 Regular vs. Irregular Heart Rhythms..... | 36 |
| Figure 9 Example of Chart-O-Matic ECG signal | 44 |
| Figure 10 Example of ECG Data from PhysioNet’s Chart-O-Matic..... | 45 |
| Figure 11 Example of a Formatted MIT-BIH Data File | 48 |
| Figure 12 Plot of Formatted vs. Unformatted Data | 49 |
| Figure 13 Olimex Development Board..... | 54 |
| Figure 14 High-level System Design Block Diagram | 56 |
| Figure 15 TI MSP430-F169 High Level Block Diagram | 58 |
| Figure 16 JTAG Hardware..... | 60 |
| Figure 19 LCD Wiring Schematic | 63 |
| Figure 20 Sparks Fun Electronics Basic 16 X 2 Character LCD..... | 64 |
| Figure 21 System Functional Block Diagram..... | 68 |
| Figure 22 LCD Display Text | 71 |
| Figure 23 DAC12 Output (binary mode)..... | 86 |
| Figure 24 Physionet file apnea_ecg_a15 waveform segment..... | 91 |
| Figure 25 File apnea_ecg_a15 snapshot take on oscilloscope..... | 91 |
| Figure 26 Physionet file cudb_cu01 waveform segment..... | 92 |
| Figure 27 File cudb_cu01 snapshot taken on oscilloscope..... | 92 |
| Figure 28 Physionet file stdb_313 waveform segment..... | 93 |
| Figure 29 File stdb_313 snapshot taken on oscilloscope..... | 93 |

LIST OF TABLES

| | | |
|---------|---|----|
| Table 1 | Command Set for NAND Memory..... | 73 |
| Table 2 | Pin Descriptions for NAND Flash Memory | 74 |
| Table 3 | Mode Selection Configurations for NAND Memory | 74 |
| Table 4 | DAC12 Control Register | 79 |
| Table 5 | DAC12LSELx Configuration..... | 81 |
| Table 6 | DAC12 Output Range..... | 82 |
| Table 7 | Settling Time vs. Power Consumption Options..... | 83 |

CHAPTER ONE:

INTRODUCTION

ECG Monitors Save Lives

Although seemingly healthy and normal by all appearances, there are people in our wonderful world who have a potentially deadly cardiac condition. Most of these individuals are unaware that they have a potentially lethal disease. This is especially true of children or adolescents with certain heart conditions. Many of these individuals die spontaneously from a deadly arrhythmia or fatal myocardial infarction. Their loved ones and friends are shocked, disbelieving, and grief-stricken. Yet, this tragedy continuously occurs in the United States¹ and throughout the world on a daily basis [1]. The reason is due to the fact that the symptoms of many heart diseases or defects can be masked or almost nonexistent [2]. Yet, there are signs that are conspicuously evident on an ECG (electrocardiography) or cardiac monitor that reveal these conditions, long before significant symptoms appear [3]. How an ECG monitor works and what it provides will be explained in chapter two but for now it is enough to know that lives can be saved and are being saved everyday by the use of cardiac monitors. When doctors use electrocardiography they can identify dangerous heart conditions, such as heartbeats that are not rhythmic, and treat the conditions appropriately [4]. However, if these conditions are not diagnosed early many of these individuals will mature into adulthood only to suffer sudden cardiac death as their first visible symptom [5]. Cardiac monitors provide a continuous display of the heart's electrical

¹ Estimates for the year 2003 are that 71,300,000 Americans have one or more forms of cardiovascular disease. Cardiovascular diseases is the number one killer in the United States. It claimed 910,614 lives in 2003 (37.3 percent of all deaths or 1 of every 2.7 deaths) [American Heart Association].

activity. Monitors can record up to twelve lead outputs providing great diagnostic capabilities [6]. Whenever possible, the patient's prior ECG data is used for comparison; monitors routinely place baseline ECG reference strips in the chart and record time and date information. Also, medical personnel need not be experts in electrocardiography in order to save lives. Today's electrocardiography equipment is so sophisticated that if professionals who come in contact with cardiac monitors and ECGs simply routinely look for certain obvious ECG signs, many people can be saved from an untimely death every year [5].

During the last century, the medical profession has come to recognize the tremendous value of the standard twelve lead ECG. Few other tests have provided the range of diagnostic utility and accuracy in determining the patient's medical health as the ECG. Today, every physician is expected to possess a reasonable level of expertise and skill in electrocardiography. National and state medical board exams test for competence in electrocardiography, as do most specialty boards; this is essential medical knowledge for all physicians. In emergencies, the ECG is the mainstay for evaluating minute-to-minute changes in vital cardiac function during life-and-death crises. Cardiac monitors displaying the heart's electrical messages are standard equipment in critical care and emergency areas. Doctors and nurses are expected to understand and interpret this information at any instant [7]. ECG monitors are essential diagnostic tools that are used in hospitals, clinics, emergency rooms, and even on most ambulances. However, electrocardiography equipment does not exist in a vacuum. These highly complex systems must be designed, built, and tested; and tremendous amounts of software must be developed to transform the primitive electrical signals of the human heart into diagnostic tools that saves lives. How does it all happen?

ECG Simulators – Need and Usage

An ECG Simulator is a human heart in a black box. That is, an ECG Simulator produces the electrical waves that the human heart would produce – without the human heart. ECG Simulators are an absolutely essential tool for the design, development and testing of any type of cardiac equipment. There are several reasons for this. The first has to do with the safety of human beings. People can not be used to provide the necessary input signals to ECG equipment during any phase of the design, or testing of the equipment (even though people would provide the best possible inputs). In fact, strict government regulations exist to insure that all medical devices that come in contact with humans have been thoroughly tested and have proven to be safe and accurate before they are ever used in a medical setting or are connected to a human being [8]. The second reason that simulator are necessary also relates to safety but has to do with the need to test all of the performance parameters of cardiac monitors and other ECG equipment before they are approved for medical use. The last reason that simulators are necessary is to allow for a way to test the ECG medical diagnostic algorithms (digital signal processing software) to insure that it is indeed, consistently, yielding the correct medical diagnosis of the heart's condition. Because these three issues are the foundation for the need and usage of ECG Simulators they will be explained in detail.

Safety Requirements – Solution ECG Simulators

In 1976 the United States Congress passed what are known as the Medical Device Amendments to the Federal Food, Drug, and Cosmetics Act [8]. The primary purpose was to ensure the safety and efficacy of new medical devices prior to marketing of the device. A “medical device” is “any item promoted for a medical purpose that does not rely on chemical action to achieve its

intended effect” [8]. Medical devices are classified in two ways. First, the division of devices into Classes—Class I, Class II, and Class III. Second, seven categories were established: preamendment, postamendment, substantially equivalent, implant, custom, investigational, and transitional. A description of the seven categories is beyond the scope of this paper (for more information on the categories see [8]), however some understanding of the medical device classes will help one understand how seriously the government takes safety issues involving any type of medical equipment. Most ECG equipment falls into the Class II category, but Class II devices must also meet Class I requirements. Some more advanced ECG equipment can also fall into Class III category [8].

Class I devices

Class I devices are subject to a comprehensive set of regulatory authorities applicable to all classes of devices, e.g., pre-market notification, registration and listing, prohibitions against adulteration and misbranding, and rules for good manufacturing practices. Class I devices account for about 40% of all medical devices; they may not be life supporting or play substantial role in preventing impairment of health. Class I devices cannot present unreasonable risk of illness or injury. Examples of class I devices include hospital beds, bandages, crutches, nasogastric tubes, and knee braces [9].

Class II devices

Class II devices include about 50% of available medical devices; these devices require special controls beyond general controls to provide reasonable assurance of safety and effectiveness.

Infusion pumps, magnetic resonance imaging (MRI), and heart monitors (ECG machines) are all class II devices [9].

Class III devices

Class III devices include about 10% of the devices available, and are the most regulated class. A class III device is defined as a medical device that supports or sustains human life or is of substantial importance in preventing impairment of human health or presents a potential, unreasonable risk of illness or injury. Insufficient information exists on a Class III device so that performance standards (Class II) or general controls (Class I) cannot provide reasonable assurance that the device is safe and effective for its intended use. For this reasons Class III devices need pre-market approval. Class III devices include pacemakers, implants of any kind, defibrillators, ventilators and vascular stents and some multipurpose monitors that include heart monitors [9], [10].

ECG Monitor Safety Issues

ECG monitors and equipment must be shown to be electrically safe. Any device that has the potential to do harm to a person is evaluated according to one or more of the following standards:

- “International Electrotechnical Commission (IEC) 60601-1 Medical Electrical Equipment- Part 1: General Requirements for Safety,
- Advancement of Medical Instrumentation (AAMI) EC 11, Diagnostic Electrocardiographic Devices,
- Underwriters Laboratory (UL) 2601-1 Amendment 1 Medical Electrical Equipment,
- American National Standards Institute (ANSI)/AAMI ES-1 Safe current limits for electromedical apparatus,
- IEC 60529 Degrees of protection provided by enclosures (IP Code) Consolidated Edition,

- IEC 60068-2 Environmental Testing - Part 2: Test methods [11].”

The features and design of the device will determine which of the above standards are used and whether other standards are appropriate in addition to or in place of these. If the device is intended for use as an ambulatory monitoring system, it also would be tested according to AAMI EC 38-1 998 Ambulatory Electrocardiographs [11]. Thus it should be clear that Cardiac monitors and ECG equipment must meet stringent pre-market testing before they can be released for sale to the public, and so an ECG Simulator (or Patient Simulator) must be used to provide a safe way of testing ECG equipment.

ECG Monitor Performance Testing

In addition to providing a safe way of testing ECG monitors, simulators are needed to test the accuracy and performance of cardiac monitors.² If a monitor is supposed to measure heart rates from 15 to 300 beats per minute, there must be some way of validating the monitor's performance in terms of this particular metric. Validation is a much abused word these days. The idea is simple enough: test the monitor against all of its requirements. However, there are many critical requirements and variables involved when dealing with human life that compound the difficulties of validating the results. And, as already discussed human beings, though they would provide the most realistic inputs, can not be used because of safety issues. To appreciate the validation process, consider how two organizations explain it. The FDA defines validation, with respect to a device, as "...establishing and documenting evidence that the device is fit for its

² The FDA requires that if a device has only been tested for its accuracy in detecting specific arrhythmias or ST level changes, it can be labeled *only* for that purpose. It is recommend that ECG and/or Cardiac Monitor be evaluated based on the ability of the device to perform basic cardiac monitoring and ECG functions according to the following standards: 1) MM1 EC 13-1993, Cardiac Monitors, Heart Rate Meters and Alarm. 2) AAMI EC 11 - 1991, Diagnostic Electrocardiographic Devices.

intended use" [12]. However, ISO 9001 says "...validation shall be performed to ensure that the product conforms to defined user needs and/or requirements," and is normally performed under operating conditions on the final product [12]. Tests also must be devised to show predictable operation outside these limits. When each of the product requirements is proven in this manner, the quality of the product is enhanced and the evidence of compliance to specifications satisfies international regulatory agencies. Modern patient monitors are multiprocessor designs with specialized instrumentation. They have custom display systems, accommodate the need for real-time data acquisition, implement digital signal processing and display waveforms instantaneously while calculating data. Most monitors measure respiration, pulse, invasive and noninvasive blood pressures, temperature and cardiac output (ECG). To test such a device in the manner that it will be used, without ever using a human being, presents some challenges.

The testing of an ECG/EKG patient monitor requires a simulator that can present the all of possible electrical/physiological waveforms that a human could present and also simulates the fault conditions which must be detected by the monitor. The input parameters for monitor testing can be acquired in two primary ways. They can be created synthetically by using formulas and calculations or they can be acquired from human beings indirectly. Patient simulators are currently available for most of the parameters that patient monitors measure (the available simulator will be examined later in this chapter). However, most of these simulators are geared for the largest possible market. For their size and cost, these simulators can provide an amazing number of features. But there are shortcomings as well. To perform the heart-rate test previously mentioned, a simulator with all possible heart rates would be ideal. Commercial simulators typically produce several fixed heart waves and heart rates, allowing for a rough,

qualitative assessment of the monitor, but not testing outside a certain range. What is needed is a way to provide testing of ECG monitors using real human data input, without, of course, the real human. How can this be done? By tapping into data that has been collect from past ECG patients and using it through a specially designed ECG simulator.

ECG Monitor Algorithm Development

An ECG Simulator is not only necessary for validating ECG Monitor performance requirements and for safety reasons, but it is also an indispensable tool for providing ECG signals for the development of diagnostic algorithms. In order to provide valuable diagnostics a cardiac monitor must do more than just provide a picture of the heart's waveforms. It must help the medical staff analyze the heart's waveforms in terms of the health or condition of the heart. This is done though the design and development of software based algorithms that scrutinize the on going signals generated by the heart. What is commonalty referred to as a QRS detector is one example of such an algorithm. A QRS detector automatically detects when the heart is beating. These beat detection algorithms are essential for many types of biomedical signal analysis and patient monitoring. In the case of the electrocardiogram, QRS detection is used in almost all automated ECG analysis algorithms and applications [13]. These include cardiac diagnosis, heart rate estimation, and arrhythmia detection. Equally important to the design of QRS algorithms is their validation. The usage and impact of the QRS algorithms in commercial medical devices requires high standards of performance assessment. For this reason, guidelines have been introduced to specify standard measures of performance such as the sensitivity and the positive predictivity. Specifically, the Association for the Advancement of Medical Instrumentation (AAMI) created a series of standards to address the problems of algorithm

benchmarking. In the assessment of these algorithms synthetic (artificially created) waveforms can not be used [14]. In order to achieve dependable results, assessment needs to be carried out on human ECG signals. Currently, several standard ECG databases are available for the evaluation of ECG detection algorithms. These include the MIT-BIH Database and the European ST-T Database [15], [16]. These databases have proven to be a valuable resource for algorithm development and enable developers to report comparable and reproducible results. The use of the database ECG signals will be further examined in Chapter Three.

Thesis Objective

It has been explained that from a safety standpoint human beings can not be used in the development phase or the testing phase of ECG equipment. Yet in order to test the performance of an ECG monitor's hardware and to develop diagnostic algorithms (software) the best test input is actual human ECG waveforms. In fact the testing of diagnostic algorithms must be performed on actual human ECG data, as required by the AAMI (Association for the Advancement of Medical Instrumentation)[14]. However, it appears (after much research) that currently there are no portable ECG simulators on the market that use actual human ECG waveforms. Thus the objective of this thesis is to show that the design of a portable ECG simulator that uses actual human ECG waveform data is completely feasible. Thus, a prototype such a tool is presented here. The high-level block diagram of the prototype design will be presented below but first it would be beneficial to examine what exactly the currently available ECG simulators (portable) can do and also to notice what they do not do.

Commercially Available Simulators

Ferronato - FC3D

The Ferronato - FC3D is designed for testing ECG monitors, Holter recorders, electrocardiographs, QRS synchronized or triggered devices, etc. It provides a digitally generated waveform, with high amplitude and rate stability. The manufacture claims the following features:

- “Realistic signals (Leads)
- Microcontroller based
- Realistic output impedances
- Long battery life (>300 h)
- Easy connection: Banana and Snap-On
- "Low Battery" indicator
- Membrane push-buttons
- Small, sturdy and lightweight [17]”

The waveform the unit simulates is a normal ECG wave (synthetic). The signal could be a Lead I, II or III, configuration according to terminals connection. The company also manufactures a 12 lead simulator with much the same features [17]. What the unit does not provide is an ECG waveform that was generated by a human being.

Dale EHS10

The Dale EHS10 is a compact, battery-operated ECG simulator that is designed to test the operating performance of ECG equipment, including electrocardiographs, patient monitors, and telemetry systems. The EHS10 performs 5-lead simulation. The unit can produce 10 ECG performance testing waveforms. The waveforms the unit can produce are types of sine, square,

and triangle waveforms. The unit has several frequency settings. The unit does not provide an ECG waveform that was generated by a human being [18].

Dale13 ECG Simulator

The Dale13 is a small battery operated ECG simulator that is installed with waveforms designed to verify the accuracy of ECG machines, monitors and telemetry units. The unit contains 12 arrhythmias signals along with the normal sinus rhythm wave, square wave and triangle wave. Several different frequency settings are available and two amplitude settings [19]. The arrhythmias statically installed on the unit are:

- “Atrial Fibrillation
- Second-Degree heart block type I
- Right-Bundle branch block
- Premature Atrial Contraction
- Premature Ventricular Contraction
- Run of 5 PVC's
- Bigeminy
- PVC R-on-T wave
- Ventricular Tachycardia
- Ventricular Fibrillation [19]”

The Dale13 does not provide ECG waveforms that were generated by a human being, even the arrhythmias are statically installed – they don't vary or change as human ECGs do.

NETECH MiniSim 1000

The MiniSim 1000, arrhythmia simulator is a comprehensive patient simulator (\$1000.00). It is designed for testing the performance of patient monitoring instrumentation quickly and easily. The MiniSim 1000 provides full 12 lead ECG simulation with 14 user selectable rates from 30 to 350 BPM and 14 user selectable amplitudes from 0.15 to 5 mV. sine, square, and triangle

performance waveforms are simulated with 14 user selectable rates from 0.1 to 100 Hz. The MiniSim also simulates 45 different arrhythmias including atrial, ventricular, tall T wave, ST elevation, ST depression, myocardial infraction, blocks, pulse, and pacer waveforms [20]. The unit says nothing about providing ECG waveforms that were generated by human beings. It appears that the waveforms are mathematically created based on combining sine, square and triangle waves.

Bio-Tek Lionheart 3

Bio-Tek's Lionheart 3 is a multipurpose simulator-tester-interpretor. As far as ECG simulation the unit provides:

- “a 12 lead configuration with independent outputs referenced to Right Leg (RL)
- ECG Waveform Selections: Adult or Pediatric
- Amplitudes: 0.5,1.0,1.5, and 2.0 mV +/-2%
- Square Wave: 2.0, 0.125 Hz
- Pulse: 30, 60 BPM, with 60 mS pulse width
- Sine Waves: 0.5, 5,10, 40, 50, 60, 100 Hz
- Triangle Wave: 2.0, 2.5 Hz
- ST Segments; Elevated or Depressed: 1mV - 0.8mV to +0.8mV in 1mV steps
- Arrhythmia Selections [21]”

The unit features say nothing about ECG waveforms being generated by a human being. Again, it appears that the waveforms are mathematically created based on combining sine, square and triangle waves.

MULTI-PRO 2000

The MULTI-PRO 2000, Electrical Safety Tester is a micro-processor based, full function electrical safety analyzer with a built-in 12 lead ECG Simulator. The ECG Simulator displays a normal sinus rhythm waveform with 13 amplitude and 14 rate selections. Square, Sine, and

Triangle waveforms for performance testing are available with 14 frequency and 13 amplitude selections [22]. The unit does not provide ECG waveforms that were generated by a human being.

DELTA 3000

The DELTA 3000, Defibrillator Analyzer combines the testing functions of a Defibrillator Tester, a Pacemaker Analyzer, and a 12 Lead ECG / Arrhythmia simulator into one instrument. The ECG simulator provides ECG: Normal Sinus Rhythm at several rates, Sine, Square, Triangle and Pulse at several rates and Arrhythmias (VFBC, VFBF, VTAC, AFIB, PVC, PVC1, PVC2, BGY, RBB, AFIB, ATRFT, NSR) [23]. The unit features say nothing about ECG waveforms being generated by a human being. Again, the waveforms are simulated sine, square and triangle waves.

M311 ECG Simulator

The M311 ECG Simulator features 10-lead output and 3 widths. It will trigger any ECG shape up to a rate of 1500 BPM. High level output. The unit's features and some specifications are:

- Waveforms - Four shapes stored in each plug-in ECG Memory Module.
- Durations - Three for each shape in a 4 to 1 ratio.
- Rate - 40 to 300 BPM in 10 steps.
- Amplitude - 0.05 to 5.0 mV in 11 steps.
- Polarity - Normal or Inverted.
- Trigger switch produces arrest, a single beat or multiple beats at any rate.
- 10 lead output - RL, RA, LA, LL & V1-V6.
- Input / Output Features: High level ECG Output. Timing pulse generated at start of ECG. Trigger ECG rate and/or shape with pace- maker or function generator, up to 1500 BPM. Mix ac or dc signals with the ECG.
- Battery power or external power source.
- Certificate of Conformance traceable to NIST is available.

Input features include the capability to trigger any of the four ECG wave shapes at any desired rate, within the range of zero to 1500 BPM, using an external pulse generator [24]. This unit has many features but says nothing about ECG waveforms being generated by a human being. Again, the waveforms are simulated.

Summary of Commercially Available Simulators

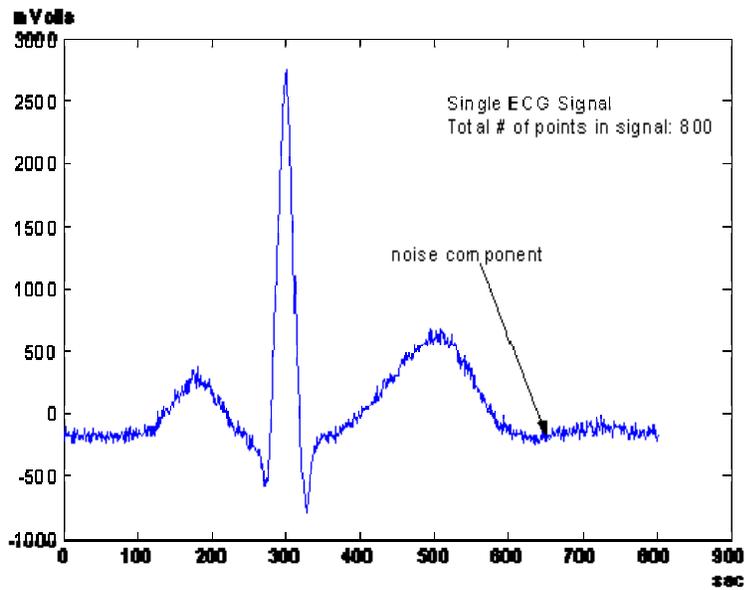
In addition to the above sited ECG Simulators several others were examined. But all of the portable commercially available Simulators³ found had one thing in common: they create their own waveforms. Typically sine, triangle, and square waves (or combinations there of) are created and used to simulate the human hearts wave form. These are also fixed or static signals that are created once and then repeatedly run to provide the appearance of an ECG waveform (perfectly identical waveforms are produces). However, the human ECG is not a fixed waveform in the shape of a sine, triangle or square wave.

The Problem with Current ECG Simulators

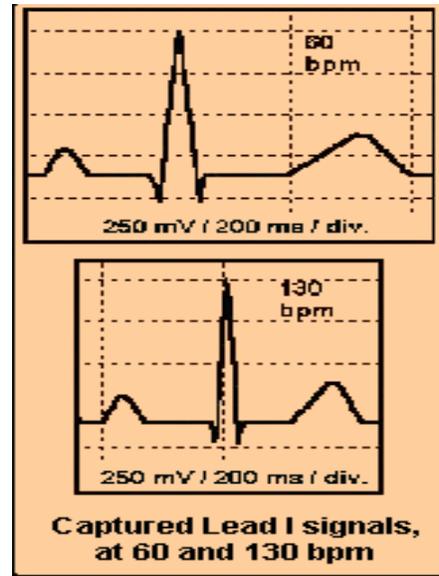
The use of static signals for the testing and development ECG monitors might not appear to be a significant weakness; in fact static signals are used for many tests. However, static signals do not present the machine under test with real-life scenarios and this can be a significant weakness. Consider the waveforms shown in Figure 1, the first waveform, Plot (a) illustrates a normal,

³ All of the simulators reviewed here are manufactured to be sold on the open market. Certainly other simulators may exist beyond these commercially available units and it could be possible that a unit uses human ECG waveforms, however, none could be found by commercially available manufactures. Non-portable simulators that use human ECG data are know to be used in the development of arrhythmia detection.

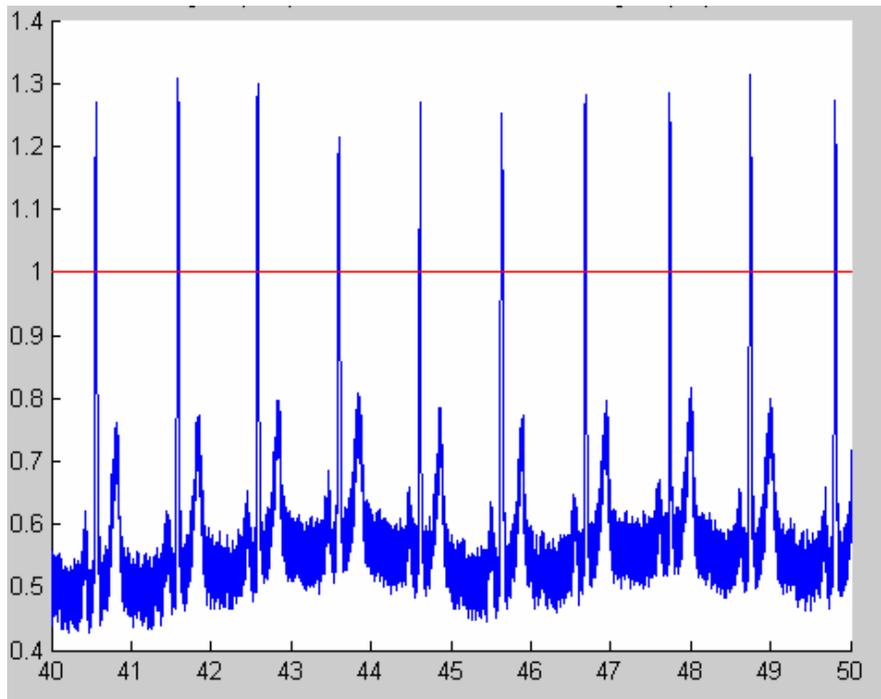
virtually noise free, human ECG waveform for one heart cycle; Plot (b) shows a fixed wave created by a typical commercially available ECG Simulator for one heart cycle (Waveform produced by the Ferronato - FC3D; see Appendix for permission to use images) [17]; Plot (c) again shows a real human ECG this time with some of the normal noise still in the signal and continuing for 10 heartbeat cycles (produced in Matlab using an ECG data file) [15]. In a real world medical setting what the cardiac monitor will get as input, and what it must perform its diagnostic analysis on, will look like Plot (c). Take note first of Plot (a) it is virtually noise free, but it is not totally noise free, which means along with the major peaks and valleys there is variation or many minor peaks and valleys – this is the reality of the human heart’s signal. Even after noise filtering is done on the ECG signal it is not a perfectly smooth wave. Now notice the ECG Simulator wave, Plot (b), it is totally smooth and totally predictable, it will repeat the cycle shown such that the next wave will be precisely identical to the one shown. This will go on continuously in order to simulate the heart – each peak relating to the heartbeat of the heart. The problem is that the simulated signal is too predictable, too smooth, there is not even the slightest amount of variation, and there can’t be, because the wave is synthetically created with mathematical formulas that never change.



Plot (a) Human ECG signal
(one heart-cycle, very little noise)



Plot (b) Commercial Simulator
ECG signal



Plot (c) A Human ECG Wave (10 heart cycles with noise)

Figure 1 Comparison of Human ECG and Simulated ECG Signals

The wave can only statically repeat itself. As already pointed out the ECG signal in Plot (c) is an actual human ECG waveform for 10 heartbeats. Notice that as time goes on this wave repeats itself as well – in an “almost identical” fashion. But an almost identical fashion is not the same thing as an “identical” fashion; if the wave is examined closely one observes that the wave does have variation. Also notice the low frequency baseline shift; this is due to the patient breathing. The point is that the ECG wave, even from a perfectly normal health heart, will produce some variation. Why is this important? The ECG diagnostic algorithms that detect heart abnormalities must work consistently and reliably on real human heart ECG waves even when those heart waves themselves are not totally without noise and variation over time.

More on ECG waveform diagrams will be discussed in Chapter 2, but the diagrams here illustrate the primary weakness with the current ECG Simulators – the signals are not authentic to the actual human heart. The commercial available portable ECG Simulators are sufficient to test ECG monitors for the most basic hardware related performance metrics. However, they are not sufficient to test whether or not the monitor’s diagnostic algorithms (software) are working correctly and accurately detecting heart abnormalities. And yet, this is what ECG monitors must unfailingly do if they are to save lives.

Solving the Problem – Prototype Overview

The goal of this thesis project is to demonstrate that a small portable low-power authentic ECG simulator can be designed, and a prototype built, that produces actual human ECG wave data and not synthetic waveforms. The prototype will not have all of the features of a fully developed ECG simulator might have; however, it will do what the commercial available portable ECG

simulators don't do. One could easily copy what the current simulators already do but that would not provide an innovation that produced a better simulator. Thus, the focus of this prototype simulator will be on doing what the commercial available portable ECG simulators do not currently do – provide an authentic human ECG wave for the monitor under test to analyze.

The Authentic ECG Simulator, as the prototype will be called, will be a portable tool that can be used to test ECG Monitors with a particular emphasis on testing the monitor's diagnostic algorithms. The unit will take as input files from the MIT/BIH Data Base. The files from the MIT/BIH Data Base will be actual ECG waveform data sets that were taken from patients in the past and reformatted for the simulator. The MIT-BIH data base files will be discussed in detail in Chapter Three. The user will be able to download any particular ECG data files that are of interest. Dozens of MIT-BIH data files will be held on a flash memory module. The files will be changeable so as to provide any type of ECG signal necessary to check an algorithm's ability to detect a particular heart defect or symptom. The user will also have the ability to manipulate the amplitude and the frequency of the data as he or she deems necessary for the equipment test. This will be done using a simple set of push buttons and a small LCD display. The data will then be processed by the microcontroller and then converted to an analog signals and output via a four lead network. These four leads will be the input for the ECG monitor under test. The high level system block diagram of the Authentic ECG Simulator is shown in Figure 2.

Chapter Two will discuss: first, how the human heart produces electrical signals, second, how those signals are then observed and recorded by ECG monitors, third, how ECG monitors can aid

in the identification of heart irregularities and thus save lives, and fourth, based on ECG monitors are required to do, the implications for ECG simulators.

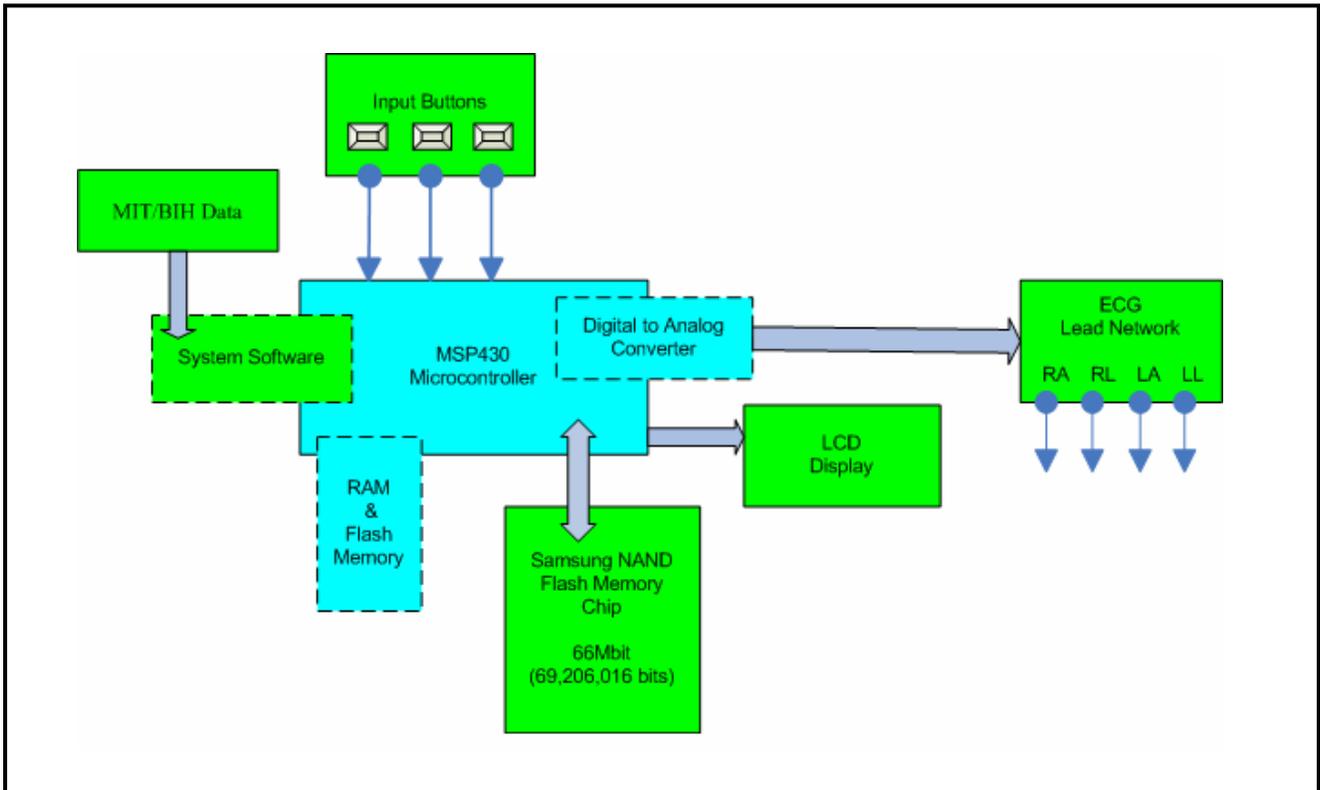


Figure 2 High-level System Block Diagram

Chapter Three will discuss how authentic ECG signals can be obtained from data bases and how this data can be formatted for use by the Authentic ECG Simulator. Chapter Four will discuss all of the major hardware components required by the Authentic ECG Simulator. Chapter Five will discuss the development of the software that was necessary for the Authentic ECG Simulator. Finally, Chapter Six will discuss the important results and conclusions that were obtained during the development of the Authentic ECG Simulator prototype.

CHAPTER TWO: ECG MONITORS FUNCTIONALITY

The Heart

Heart – Basics

Before one can appreciate the details of an electrocardiogram produced by an ECG monitor a basic understanding of the heart and the electrical signals generated by the heart is necessary. The cardiovascular system is the key element in transporting and exchanging nutrients, wastes, oxygen, and carbon dioxide between the environment and the cells that function in tissues. It is the heart that provides the force to move blood through the circulatory system so that these metabolic fuels and waste products can be transported to and from the body tissues [25]. In this biological pump, the muscles of the chamber walls of the heart provide the driving force, the valves direct the flow, and the great vessels convey the blood to and from body tissues. The chambers of the heart alternately contract and relax in a rhythmic cycle. During the period of contraction (systole), the heart pumps blood out through the arteries; during the period of relaxation (diastole), the heart fills with blood. One complete sequence of filling and pumping blood is called a cardiac cycle, or heartbeat. The heart's rhythm of contraction is controlled by the sinoatrial node (SA node), often called the pacemaker. This node is part of the heart's intrinsic conduction system, which is made up of specialized myocardial cells called nodal cells. The heart will beat without input from the nervous system and will continue to beat, even outside the body, as long as its cells are alive. This automatic nature of the heartbeat is due to the

spontaneous electrical activity of the SA node. Electrical impulses generated from the SA node spread through the heart via a nodal tissue pathway that coordinates the events of the cardiac cycle [25].

Like the spark-plug of an automobile the SA node generates a number of “sparks” per minute. Each “spark” travels across a specialized electrical pathway and stimulates the muscle walls of the four chambers of the heart to contract (and thus empty) in a certain sequence or pattern. The upper chambers or atria are stimulated first. This is followed by a slight delay to allow the two atria to empty. Finally, the two ventricles are electrically stimulated. In an automobile, the number of sparks per minute generated by a spark plug is increased when you press the accelerator. This increases the RPMs of the engine. In case of the heart, adrenaline and the bodies need for oxygen act as an accelerator and cause the SA node to increase the number of sparks per minute, which in turn increases the heart rate. The release of adrenaline is controlled by the nervous system. The heart normally beats at around 72 times per minute and the SA node speeds up during exertion, emotional stress, or whenever our body needs an extra boost of blood supply. In contrast, it slows down during rest or under the influence of certain medications [26].

The Electrochemical Perspective of the Heart

From an electrical perspective the human heart produces bioelectric potentials (electrical signals) as a result of the chemical activity of certain cells known as excitable cells. These excitable cells exhibit a resting potential and when stimulated by the SA node an action potential. When in the resting state the electrical potential is in the range of -50 to -100 millivolts, relative to the external state (outside the cell). The cell membrane is very thin (7-15nm) lipoprotein complex

that is essentially impermeable to intracellular protein. The membrane in the resting state is only slightly permeable to Na^+ but rather freely permeable to K^+ and Cl^- [27]. The permeability of the resting membrane to potassium ions is approximately 50 to 100 times larger than its permeability to sodium ion. This concentration imbalance creates a diffusion gradient that is directed outward across the membrane. The movement of the K^+ along this diffusion gradient makes the interior of the cell more negative relative to the outside of the cell (positive charge is removed from the interior). The membrane thus may be viewed electrically as a leaky capacitor. Maintaining steady-state ionic imbalance between the internal and external media of the cell requires the continual active transport of ions against their normal electrochemical gradients. The active transport mechanism is located within the membrane and is referred to as the *sodium-potassium pump*. It actively transports Na^+ out of the cell and K^+ into the cell in the ratio 3 Na^+ to 2 K^+ . The associated pump current is a net outward current that increases the negativity of the intracellular potential. Thus the factors influencing the flow of ions across the membrane are (1) diffusion gradients, (2) the inwardly directed electric field, (3) membrane structure, and (4) active transport of ions against an established electrochemical gradient. Thus the distribution of ions across the cell membrane and the structure of the membrane account for the resting potential [27].

Another property of an excitable cell is the ability to conduct an action potential when adequately stimulated. The stimulus brings about a depolarization in the membrane that is sufficient to exceed the threshold potential of the membrane and thereby elicit an all-or-none action potential that travels in an unattenuated fashion at a constant conduction velocity along the membrane of the excitable cell. Because of the steady resting potential (-50 to -100 mV) the cell

membrane is said to be *polarized* (action potential). A lessening of the magnitude of this polarization is called *depolarization*; an increase in its magnitude is referred to as *hyperpolarization*. Because of the all-or-none property of the action potential the membrane potential goes through a very characteristic cycle – from the resting level for a certain fixed duration to the action potential for a fixed duration. For a nerve fiber, the change in voltage is about 120 mV. and the duration is approximately 1 ms [27]. The action potential (active state) is the result of the voltage and the time dependent nature of the membrane permeabilities to Na^+ and K^+ . As the cell membrane is depolarized the membrane permeability to Na^+ is significantly increased. As a result, Na^+ rushes into the internal medium of the cell bring about further depolarization, which in turn brings about a further increase in the conductance of the membrane to Na^+ . If the membrane threshold is exceeded, this process is self-regenerative and leads to a *runaway depolarization*. Under these conditions the membrane potential approaches a value of about +60mV. The membrane potential never quite reaches this level, however, because of two factors: first, Na^+ conduction is not only voltage-dependent but also time-dependent, and it is relatively short-lived compared with the action potential, and second, there is a delayed increase in K^+ conduction that acts as a hyperpolarizing influence, tending to return the membrane to the resting level [27]. Consider the graphical representation of the action potential shown in Figure 3 (see Appendix for permission to use images). The action potential has 5 phases (numbered 0-4) [28]. These phases directly correspond to segments of the ECG wave observed on an ECG monitor. The ECG wave segments (Q, R, S, P, T) will be explained in detail in the section titled “ECG Waveform Segments” below. Starting at the far left of Figure 3, phase 4 is the resting membrane potential, and describes the membrane potential when the cell is not being stimulated (-50 to -100 mV).

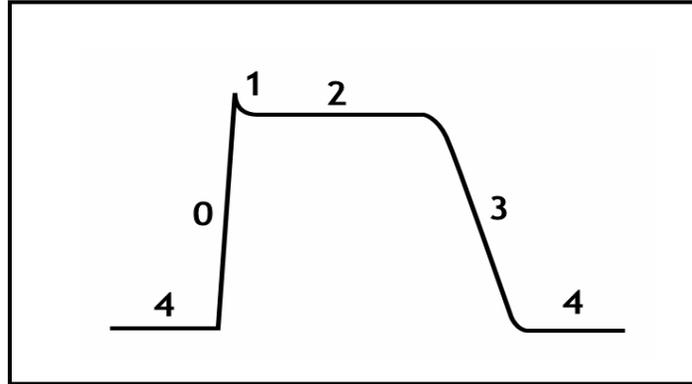


Figure 3 Cardiac Action Potential

The next phase, phase 0, is the rapid depolarization phase. The slope of phase 0 is determined by the maximum rate of depolarization of the cell. Phase 1 of the action potential is due to closure of the Na^+ channels. The transient net outward current is due to the movement of K^+ and Cl^- ions. Together, 0 and 1 correspond to the R and S waves of the ECG. The voltage at point 1 in the figure approaches a value of approximately +60mV [27], [28]. Phase 2 of the action potential corresponds to the S-T segment of the ECG. This “plateau” phase of the cardiac action potential is sustained by a balance between inward movement of Ca^{2+} through L-type calcium channels and outward movement of K^+ through potassium channels. During phase 3 of the action potential, the K^+ channel is still open, allowing more K^+ to leave the cell. This net loss of positive charge causes the cell to repolarize. The k^+ channels close when the membrane potential is restored to about -40 to -45 mV. Phase 3 of the action potential corresponds to the T wave on the ECG [27], [28].

The Heartbeat Perspective

As the SA node fires, each electrical impulse travels through the right and left atrium. This electrical activity causes the two upper chambers of the heart to contract (the P wave on the patient's ECG). The electrical impulse then moves to an area known as the AV (atrio-ventricular) node. This node sits just above the ventricles. Here, the electrical impulse is held up for a brief period. This delay allows the right and left atrium to continue emptying blood into the two ventricles (called the “PR interval” of the ECG). The AV node thus acts as a “relay station” delaying stimulation of the ventricles long enough to allow the two atria to finish emptying. Following the delay, the electrical impulse travels through both ventricles (via special electrical pathways known as the right and left bundle branches). The electrically stimulated ventricles contract and blood is pumped into the pulmonary artery and aorta (called the “QRS complex” of an ECG). The ventricles then recover from this electrical stimulation and generate what is called the “S-T segment” and the T wave on the ECG [26], [28].

The Electrocardiogram

ECG Waveform Segments

With every heartbeat the heart generates the sequence of electrical activity just explained. As the heart undergoes depolarization and repolarization, the electrical currents that are generated spread not only within the heart, but also throughout the body. This electrical activity generated by the heart can be measured by an array of electrodes placed on the body surface. The tracing of this signal is called an electrocardiogram (ECG, or EKG) and can be displayed on paper or on

a display monitor. The different wave segments that comprise the ECG represent the sequence of depolarization and repolarization of the atria and ventricles [29].

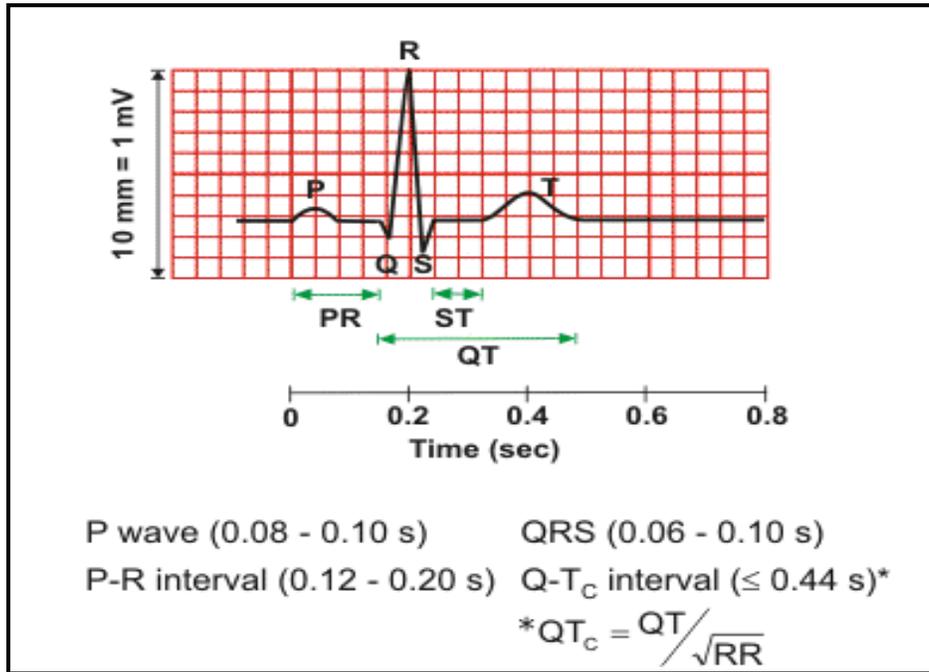


Figure 4 ECG Wave Segment Breakdown

The ECG is typically recorded at a speed of approximately 25 mm/sec, and the voltages are calibrated so that 1 mV = 10 mm in the vertical direction [29]. Figure 4 illustrates a normal ECG wave for one heartbeat cycle and identifies the wave segments (see Appendix for permission to use images). Starting from the left in Figure 4, the P wave represents the part of the heart's cycle when depolarization spreads from the SA node throughout the atria, and is usually 0.08 to 0.1 seconds in duration. The brief isoelectric (zero voltage) period after the P wave represents the time in which the impulse is traveling within the AV node (where the conduction velocity is

greatly retarded) and the bundle of His⁴. The period of time from the onset of the P wave to the beginning of the QRS complex is termed the P-R interval, which normally ranges from 0.12 to 0.20 seconds in duration. This interval represents the time between the onset of atrial depolarization and the onset of ventricular depolarization [29].

The next segment of the ECG wave is called the QRS complex (see Figure 4). The QRS complex represents ventricular depolarization. The duration of the QRS complex is normally 0.06 to 0.1 seconds. This relatively short duration indicates that ventricular depolarization normally occurs very rapidly. If the QRS complex is prolonged, conduction is impaired within the ventricles. This can occur with bundle branch blocks or whenever a ventricular foci (abnormal pacemaker site) becomes the pacemaker driving the ventricle. Such an ectopic foci nearly always results in impulses being conducted over slower pathways within the heart, thereby increasing the time for depolarization and the duration of the QRS complex [30]. The shape of the QRS complex in Figure 4 is the ideal. The shape will change when there is abnormal conduction of electrical impulses within the ventricles.

The next segment of the ECG wave is the S-T segment. This isoelectric period following the QRS is the time at which the entire ventricle is depolarized and roughly corresponds to the plateau phase of the ventricular action potential. The S-T segment is important in the diagnosis of ventricular ischemia or hypoxia because under those conditions, the S-T segment can become either depressed or elevated [29]. These conditions will be covered later under arrhythmias

⁴ The “bundle of His” is a specialized muscle – fibers in the wall between the ventricles that carry the electric impulse to the ventricles.

(irregular heartbeats). The T wave represents ventricular repolarization and is longer in duration than depolarization because the conduction of the repolarization wave is slower than the conduction of the wave of depolarization [28].

The Q-T interval represents the time for both ventricular depolarization and repolarization to occur and therefore roughly estimates the duration of an average ventricular action potential. This interval can range from 0.2 to 0.4 seconds depending upon heart rate. At high heart rates, ventricular action potentials shorten in duration, which decreases the Q-T interval. Because prolonged Q-T intervals can be diagnostic for susceptibility to certain types of tachyarrhythmias (abnormally fast heartbeat), it is important to determine if a given Q-T interval is excessively long. In practice, the Q-T interval is expressed as a corrected Q-T by taking the Q-T interval and dividing it by the square root of the R-R interval (interval between ventricular depolarizations). This allows an assessment of the Q-T interval that is independent of heart rate. Normal corrected Q-T (correct) intervals are less than 0.44 seconds. There is no distinctly visible wave representing atrial repolarization in the ECG because it occurs during ventricular depolarization. Because the wave of atrial repolarization is relatively small in amplitude (i.e., has low voltage), it is masked by the much larger ventricular-generated QRS complex [29], [31].

What ECG Monitors Do

The ECG monitor is a real-time system that continuously produces the heart's waveform and continuously analyzes the on-going pattern of the signal. Just one heart cycle is not enough to shed light on the condition of the heart; the electrocardiogram is a series of heartbeat signals that are analyzed as a whole. Figure 5 illustrates an actual electrocardiogram output (see Appendix

for permission to use images) [28]. The electrocardiogram is an important part of the initial evaluation of a patient who is suspected to have any heart related problem. The ECG is an extremely safe non-invasive test and there is no risk involved [32].

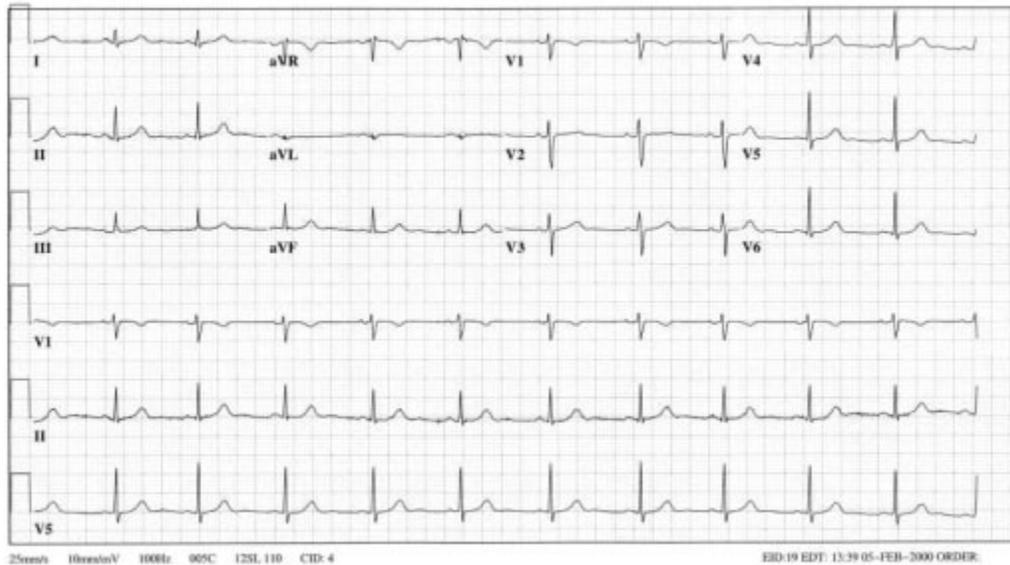


Figure 5 Actual Electrocardiogram output

The ECG can provide important information about the patient's heart rhythm, a previous heart attack, increased thickness of heart muscle, signs of decreased oxygen delivery to the heart, and problems with conduction of the electrical current from one portion of the heart to another. For example, the ECG tracing may demonstrate an acute or ongoing heart attack involving the bottom or inferior portion of the heart [31].

ECG Stress Test

Patients with coronary artery blockages may have minimal symptoms and an unremarkable or unchanged ECG while they are at rest. Also, patients with no heart disease may have symptoms and a suspicious ECG. A cardiac stress test can expose hidden heart disease or to help rule it

out. The heart may be stressed by having a patient exercise on a treadmill or a stationary bicycle. If the patient is unable to exercise due to physical limitations the physician may choose a chemical form of test. Stress testing, particularly those employing exercise, can help reveal the following:

1. "The physical tolerance and conditioning of the patient.
2. Shortness of breath, chest discomfort, dizziness and unexpected weakness may suggest underlying heart disease.
3. The blood pressure is recorded at intervals during the stress test. High blood pressure during exercise may provide an early clue to heart related problems.
4. A drop in blood pressure during exercise may also indicate heart disease.
5. Exercise may provoke arrhythmias which may not be seen at rest and may or may not point to heart disease.
6. The ECG is constantly monitored during the exercise and recorded on paper at intervals and compared to the ECG obtained at rest. Changes in the S-T segment and T-waves may indicate heart disease [33]."

During stress testing, the heart rhythm is constantly displayed on a heart monitor. This allows the physician to keep an eye on the patient's heart rate, and to watch for telltale ECG changes and irregular heart rhythm. There are several other variants of the stress test as well, all of them involving the use of an ECG monitor along with other patient monitoring devices.

Cardiac Arrhythmias

The rhythm of the heart is normally generated and regulated by the sinoatrial (SA) node, located within the wall of the right atrium. SA nodal pacemaker activity normally governs the rhythm of the atria and ventricles. Normal rhythm is very regular, with minimal cyclical fluctuation.

Furthermore, atrial contraction is always followed by ventricular contraction in the normal heart.

When this rhythm becomes irregular, too fast (tachycardia) or too slow (bradycardia), or the frequency of the atrial and ventricular beats are different, this is called an arrhythmia.

Arrhythmia is a generalized term used to denote any disturbances in the heart's rhythm. Normal

sinus rhythm is characterized by a regular rhythm and P-R interval duration in the range of 0.12 seconds to 0.20 seconds [34].

About 14 million people in the United States have arrhythmias (5% of the population). The most common disorders are atrial fibrillation and flutter. The incidence is highly related to age and the presence of underlying heart disease; the incidence approaches 30% following open heart surgery. Proper diagnosis of arrhythmias requires an electrocardiogram. Arrhythmias can be either benign or more serious in nature depending on the hemodynamic (circulation of the blood) consequence of the arrhythmia. Occasional premature ventricular complexes (PVCs), while annoying to a patient, are generally considered benign because they have little hemodynamic effect. Consequently, PVCs if not too frequent, are generally not treated. In contrast, ventricular tachycardia is a serious condition that can lead to heart failure, or worse, to ventricular fibrillation and death [29], [35].

Arrhythmia is often caused by an interruption in the conduction path of the stimulus from the SA node. Recall, from the SA node electrical impulses quickly spread through the right and left atria via muscle cells in the myocardium. The next stop for the electrical impulses is the atrioventricular (AV) node, low in the right atrium. Here, impulses are delayed for about a tenth of a second. The impulses then pass into the ventricles, where they meet the bundle of His. The bundle of His is like a telephone cable and transmits impulses from the AV node to the rest of the ventricles. It divides into two thinner “cables,” known as the left and right bundle branches. The bundle branches divide into still thinner cables called Purkinje fibers, which transmit

impulses to all parts of the ventricles. After leaving the Purkinje fibers, impulses travel from muscle cell to muscle cell to all cells of the ventricular muscle layer [36].

When the conduction path from the bundle of His is interrupted, to only a slight degree the resulting arrhythmia is referred to a first-degree heart block or first-degree AV nodal block. Figure 6 illustrates the positions of the AV node and the Bundle of His relative to the ventricular and atrial muscles. The conduction velocities of the action potential, during the signal transmission, are shown on Figure 6 (see Appendix for permission to use images). In the case of first-degree block the atrial impulses reach the ventricles, but the P-R interval is abnormally prolonged because of an increase in transmission time through the region. The conduction velocity is slowed so that the P-R interval is increased to greater than 0.2 seconds [27], [35]. In second-degree AV nodal block the conduction velocity is slowed to the point where some impulses from the atria cannot pass through the AV node—not all atrial impulses are conducted to the ventricles. There may be, for example, one ventricular beat every second or third atrial beat. This can result in p-waves that are not followed by QRS complexes. For example, one or two p-waves may occur before one is followed by a QRS. When the QRS follows the p-wave, the P-R interval is increased. In this type of block, the ventricular rhythm will be less than the sinus rhythm [27], [35].

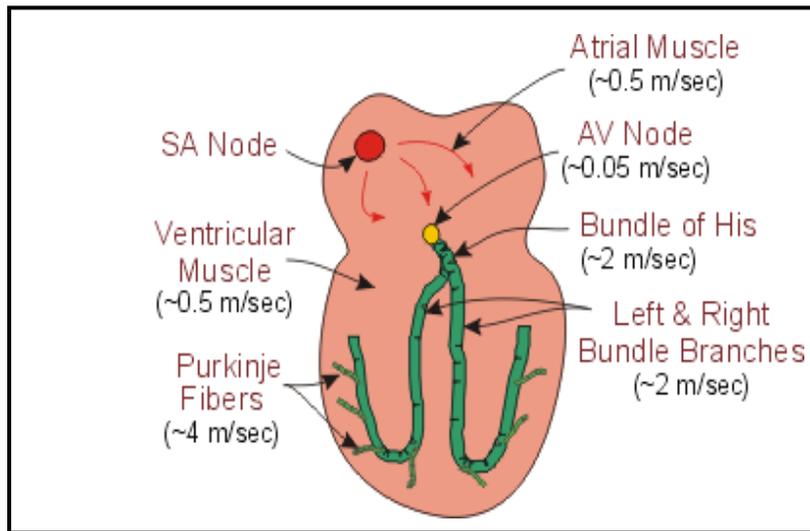


Figure 6 Specialized Conductive Tissues of Atria and Ventricles

In third-degree AV nodal block conduction through the AV node is completely blocked so that no impulses are able to be transmitted from the atria to the ventricles. QRS complexes will still occur, but they will originate from within the AV node, the Bundle of His, or other ventricular regions. Therefore, QRS complexes will not be preceded by p-waves. Furthermore, there will be complete asynchrony between the p-wave and QRS complexes. The atrial rhythm may be completely normal, but the ventricular rhythm will be greatly reduced depending upon the location of the site generating the ventricular impulse. Ventricular rates typically range from 30 to 40 beats per minute [27], [37]. Figure 7 illustrates the ECG wave forms of normal and first, second, and third degree abnormal cardiac rhythms (see Appendix for permission to use images).

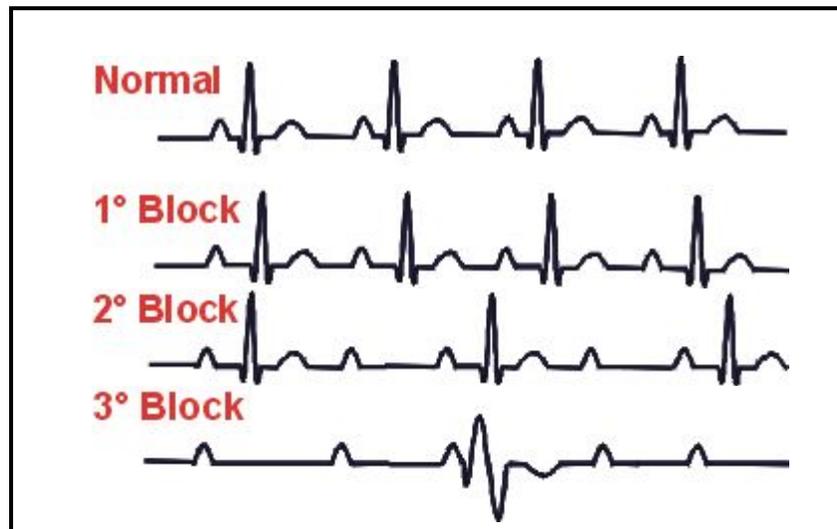


Figure 7 ECG waves of First, Second, and Third degree heart blocks

In another form of incomplete heart block involving the AV node, the P-R interval progressively lengthens until the atrial impulse fails to conduct to the ventricle. The first conducted beat after the pause has a shorter P-R interval than any subsequent P-R interval. Then the process of the lengthening of the P-R interval begins anew, progressing over several cardiac cycles until another beat is dropped. The electrocardiographic sequence starting with the ventricular pause and ending with the next blocked atrial beat constitutes a *Wenckebach* period. The ratio of the number of P-waves to QRS complexes determines the block (for example 6:5 or 5:4 *Wenckebach* periods) [27]. Rhythmic disturbances can arise from sources other than bundle of His interruptions or competing pacemakers, and thus there are many other types of arrhythmias. A few specific arrhythmias are:

- “Sinus bradycardia - low sinus rate: less than 60 beats/min,
- Sinus tachycardia - high sinus rate of 100-180 beats/min (typically occurs during exercise, excitement or anxiety),

- Sick sinus syndrome - a disturbance of SA nodal function that results in a markedly variable rhythm,
- Atrial tachycardia - a series of 3 or more consecutive atrial premature beats occurring at a frequency greater than 100/min,
- Atrial flutter - sinus rate of 250-350 beats/min.,
- Atrial fibrillation - uncoordinated atrial depolarizations,
- Junctional escape rhythm - SA node suppression can result in AV node-generated rhythm of 40-60 beats/min (not preceded by p-wave),
- Supraventricular tachycardia (SVT) - usually caused by reentry currents within the atria or between ventricles and atria producing high heart rates of 140-250,
- Ventricular premature beats (VPBs) - caused by ectopic ventricular foci; characterized by widened QRS,
- Ventricular tachycardia (VT) - high ventricular rate caused by aberrant ventricular automaticity or by intraventricular reentry; characterized by widened QRS; rates of 100 to 200 beats/min,
- Ventricular flutter - ventricular depolarizations greater than 200/min,
- Ventricular fibrillation - uncoordinated ventricular depolarizations [37].”

These arrhythmias have been listed (there are quite a few more) to show how many types of cardiac arrhythmias can exist and to note that many of the above types of arrhythmias can be recognized by critically evaluating the electrocardiogram. The point is that a first-rate ECG monitor can be a vital tool in the diagnosis of all kinds of arrhythmias and the related heart diseases. However, none these arrhythmias jump off the ECG monitor saying, “Here I am” or “Warning! You have an arrhythmia.” In fact, the ECG monitor must be programmed with highly sophisticated digital signal processing algorithms in order to help medical personnel detect these different types of arrhythmias. This requires that these arrhythmias be inputted to the ECG equipment during its development and testing. That requires an ECG Simulator that can produce all different kinds of arrhythmias. The process of detecting an arrhythmia on an electrocardiogram not always a simple one, which is why doctors and nurses need all the help they can get from the ECG equipment. If algorithm can be developed as software to aid in the arrhythmia diagnosis they will be a tremendous asset in the process. The next section outlines some of the steps in arrhythmia detection and diagnosis.

Arrhythmia Detection

There are several steps involved in identifying a arrhythmias using ECG monitor equipment. First, it is important to find out the patient's history and use this information in conjunction with the ECG. After finding out the patient's medical history, one begins by identifying and noting the exact placement of the P wave, P-R interval, QRS complex, Q-T interval and T wave on the electrocardiogram. Next, the atrial and ventricular heart rates must be calculated. A rate of less than 60 beats per minute is slow (bradycardia), 60-100 beats per minute is normal, and greater than 100 beats per minute is fast (tachycardia). The next step is to determine if the rhythm is regular or irregular. This is done by accessing whether the R-R intervals and P-P intervals are regularly spaced. The electrocardiograms in Figure 8 illustrate both regular and irregular heart rhythms (see Appendix for permission to use images). However, much of the time the irregular heart rhythms will not be this easy to identify. If the rhythm is irregular, the next step is to determine if the rhythm is occasionally irregular or regularly irregular (there is a pattern to the irregularity) or irregularly irregular (there is no pattern to the irregularity) [34].

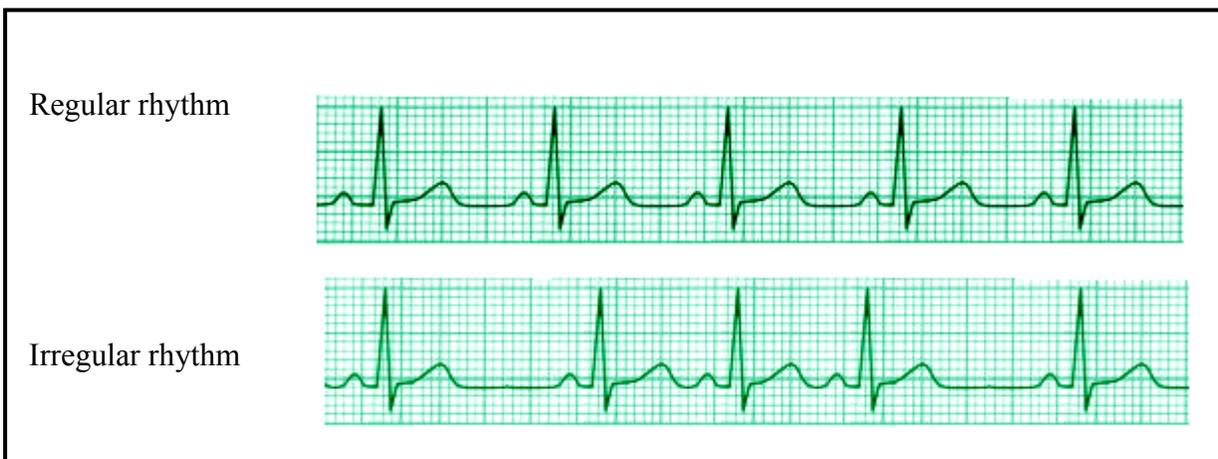


Figure 8 Regular vs. Irregular Heart Rhythms

At this point the waveform of the ECG must be evaluated in greater detail – searching for additional clues that might point to the type of arrhythmia such as:

1. “The shape of the P-wave must be scrutinized. Normally, P-waves are upright and each P wave is related closely to a QRS complex. If the P-wave is inverted, the impulse is spreading from the ventricles to the atria in a retrograde manner.
2. Determine if the P-R interval is of normal length (0.12-0.2 seconds).
3. The QRS complexes must be examined to determine if the complex is wide or narrow. Narrow QRS complexes (less than 0.12 sec) indicate that the rhythm is supraventricular. Wide QRS complexes (longer than 0.12 sec), indicate that the rhythm is originating in the ventricles or that there is an intraventricular block.
4. Determine if the S-T segment is displaced from the mid-line [34].”

Following the electrocardiogram analysis the attending physician must return to the patient's medical history and evaluate the rhythm in light of the ECG history of the patient. The following ECG clues can be used to recognize cardiac arrhythmias in non-sinus rhythm electrocardiograms: If the rhythm is regular but too fast or slow, it could be an indication of either:

1. “Sinus bradycardia: The rhythm is regular and looks normal but is slower than 60 beats per minute. The RR interval is longer, often more than one second. P-waves are present and regular and each P-wave is followed by a QRS complex in a ratio of 1:1 [34].”
2. “Sinus tachycardia: here rhythm is regular and looks normal but at a rate greater than 100 beats per minute. The R-R interval is shorter (usually less than 0.6 seconds). P-waves are present and regular and each P-wave is followed by a QRS complex in a ratio of 1:1 [34].”

If the rhythm is irregular, the following are possibilities:

1. “Atrial flutter: Atrial flutter waves (F-waves) with a characteristic saw-tooth form will be observed at a rate of 200-350 BPM.
2. Atrial fibrillation: No P-waves will be observable. Rather, a wavy base-line is observed [34].”

If there are no P-waves, it could be an indication of either:

1. “Atrial fibrillation: No P-waves observable; instead, a wavy base-line is observed.
2. Sinus arrest with junctional or ventricular escape [34].”

If P-waves are not associated with QRS complexes, it could indicate either:

1. “Ventricular tachycardia.
2. Third degree AV block [34].”

Implications for ECG Simulators

It is not the purpose this work to examine all of the possible arrhythmias that exist or even all of the subtleties of how ECG monitors work. The details provided here are only discussed to further an appreciation of what goes into cardiac ECG analysis. What must be understood is that the diagnosis of arrhythmias involves the use of complex digital signal processing algorithms that must be verified using authentic human arrhythmias. Digital signal processing algorithms for diagnostic purposes are written into the software of the ECG monitors’ system. Doctors and nurses still make the final diagnosis but they are aided by a tool that can detect subtle variations in electrocardiograms that medical personnel might miss. In fact, the task of developing reliable diagnostic algorithms for electrocardiogram equipment is an on going pursuit of both the medical profession and the manufactures of the ECG equipment. While in this pursuit the other critical component that is necessary is something to provide many types of authentic human arrhythmias to the system so it can be tested and perfected in the detection of arrhythmias. The Authentic ECG Simulator that is being proposed here will be able to provide authentic human heart arrhythmias of any type that currently exists in a vast array of ECG data bases. These ECG data bases that contain authentic ECG monitor recorded data will be discussed in the next chapter, along with the process of obtaining and using this data in the Authentic ECG Simulator.

CHAPTER THREE: PROVIDING AUTHENTIC HEART SIGNALS

The Need for Authentic Signals

If an ECG Simulator is to produce authentic human heart signals that can be used to improve the testing of ECG equipment and develop diagnostic algorithms it will be necessary to obtain these authentic heart signals from human beings. However, as already discussed, for safety reasons human beings can not be used directly to provide these heart signals. Thus the human heart signals will need to be obtained indirectly. The focus of this chapter will be on discussing how authentic human heart signals can be obtained from database files and how these files were formatted for use by the Authentic ECG Simulator of this project.

Physiologic Databases – MIT-BIH

The Laboratory for Computational Physiology (LCP) at the Harvard-MIT Division of Health Sciences and Technology is a center for research into subjects such as cardiac arrhythmia detection and heart rate variability as well as the compression, transmission, storage, and retrieval of physiologic signals. What is now the LCP was originally a research group established in the mid-1970s within the MIT Biomedical Engineering Center for Clinical Instrumentation [15]. Their research during the past 25 years has required numerous collections of physiologic data, which they have made available to other researchers (on digital and analog tapes between 1980 and 1990, on CD-ROMs beginning in 1989, via FTP between 1993 and 1999, and on the Web since 1996). Making use of these physiologic databases requires

specialized software that they have also made available at [15]. The software reads and writes digitized signals and annotations using a highly portable, flexible, and efficient set of database interface functions (the WFDB library). The software and the library are freely available, by downloading them from [15]. However, for smaller data files the software can be run directly on the MIT-BIH database sister website physionet.org simply by following the instructions and links provided at [38]. For this project only small files are necessary. Thus all of the files for this project were obtained directly from the [38] website. An explanation of the details of obtaining files and formatting the files is discussed below, but first some of the significant ECG databases available from MIT-BIH are listed and briefly explained.

MIT-BIH Arrhythmia Databases

Since 1975, the laboratories at Boston's Beth Israel Hospital (BIH) and at MIT have supported research into cardiac arrhythmia analysis and related subjects. One of the first byproducts of that effort was the MIT-BIH Arrhythmia Database, which was completed in 1980. The database was the first generally available set of standard test material for evaluation of arrhythmia detectors, and has been used for that purpose as well as for basic research into cardiac dynamics.

Originally, the database was made available on digital tape and on FM analog tape. In August, 1989, the data was made available on a CD-ROM but today the database is available via the PhysioNet website [39].

The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Twenty-three recordings were chosen at random from a set of 4000 24-hour

ambulatory ECG recordings collected from a mixed population of inpatients and outpatients at Boston's Beth Israel Hospital; the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample.

The recordings have been digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. Two or more cardiologists independently annotated each record; disagreements were resolved to obtain the computer-readable reference annotations for each beat (approximately 110,000 annotations in all) included with the database [39].

Creighton University Ventricular Tachyarrhythmia Database

This database consists of 35 records, each of which shows the onset of ventricular fibrillation (VF) [40]. These rhythms are extremely rare, and of great importance both for development and evaluation of VF detectors and for basic research on the dynamics of VF. All signals were passed through an active second-order Bessel low-pass filter with a cutoff of 70 Hz, and were digitized at 250 Hz with 12-bit resolution over a 10 V range (10 mV nominal relative to the unamplified signals) [40].

MIT-BIH Noise Stress Test Database

This database consists of 15 thirty-minute records. Three contain noise of the types typically observed in ECG recordings. They were obtained using a Holter recorder on an active subject. Electrode motion artifact noise is usually the most troublesome type of noise for arrhythmia detectors since it can closely mimic characteristics of the ECG. The remaining records reproduce MIT-BIH Arrhythmia Database records with noise added at various levels. Since the

correct beat labels are known for these records, they may be used to test the noise tolerance of an arrhythmia detector [41].

MIT-BIH ST Change Database

This database consists of 28 records. Most were obtained during exercise stress tests, these records exhibit transient ST depression in response to exercise-induced ischemia. Four records were obtained from Holter tapes, and show ST elevation. Ten records contain only one signal; the rest contain two signals [42].

MIT-BIH Malignant Ventricular Arrhythmia Database

This database consists of 22 records, obtained from Holter tapes of 16 subjects. It is annotated only with respect to rhythm changes, which include 89 episodes of ventricular tachycardia, 60 episodes of ventricular flutter, and 42 episodes of ventricular fibrillation [43].

MIT-BIH Atrial Fibrillation/Flutter Database

This database may be useful for development and evaluation of atrial fibrillation/flutter detectors that rely on timing information only. It consists of 25 records (obtained from Holter tapes of 25 subjects) containing about 300 episodes of atrial fibrillation and 40 episodes of atrial flutter [44].

MIT-BIH ECG Compression Test Database

This database consists of 168 un-annotated records, obtained from Holter tapes from 38 subjects. The records exhibit a wide variety of arrhythmias, conduction disturbances, and noise. Many were selected because the characteristics of the signal or noise may be expected to pose a problem for an ECG compression method that is not exactly invertible [45].

MIT-BIH Supraventricular Arrhythmia Database

This database consists of 78 records, obtained from Holter tapes, to supplement the examples of supraventricular arrhythmias in the MIT-BIH Arrhythmia Database [46].

MIT-BIH Long-Term Database

This database contains seven annotated long-term records ranging in length from 14 to 24 hours. These records are complete Holter tapes from seven subjects [47].

MIT-BIH Normal Sinus Rhythm Database

This database contains 18 records, each between 20 and 24 hours in length, from subjects without diagnosed cardiac abnormalities. The header files for these records include the age and gender of each subject, and the starting time of each record [48].

Using the MIT-BIH Data Records

Obtaining the MIT-BIH Data

As mentioned earlier, using the MIT-BIH database files for viewing, analyzing, and creating recordings of physiologic signals requires specialized software. Fortunately, MIT-BIH has developed a large collection of such software over the past twenty years, and much of this software has been interfaced to the PhysioNet website [38] (it can also be downloaded and used independently). By navigating through several links one can go to the PhysioNet's "Chart-O-Matic" [49] and obtain one minute ECG data files. From the "Chart-O-Matic" page one begins by choosing a database from a scroll-down menu and then clicking continue. All of the above

mentioned databases are available there and quite a few more. By simply following the instructions on the webpage the software can be executed and a GUI window opens prompting the user for more input parameters. A particular record must be chosen, annotations or no annotations can be selected, a start time is input and the chart width is selected. Clicking “Show chart” then displays the electrocardiogram. Figure 9 shows what a “Chart-O-Matic” electrocardiogram looks like. In addition to the chart of the signal, one can also obtain one minute’s worth of the data from the record. This is precisely what is needed for the project at hand. The data files can be viewed by clicking “Convert signals to text.” An example of a MIT-BIH ECG signal file is shown in Figure 10; this is the data corresponding to the plot shown in Figure 9 (see Appendix for permission to use images).



Figure 9 Example of Chart-O-Matic ECG signal

| time (sec) | MLII (mV) | V5 (mV) |
|---------------|--------------|------------|
| 0.000 | -0.145 | -0.065 |
| 0.003 | -0.145 | -0.065 |
| 0.006 | -0.145 | -0.065 |
| 0.008 | -0.145 | -0.065 |
| 0.011 | -0.145 | -0.065 |
| 0.014 | -0.145 | -0.065 |
| 0.017 | -0.145 | -0.065 |
| 0.019 | -0.145 | -0.065 |
| 0.022 | -0.120 | -0.080 |
| 0.025 | -0.135 | -0.080 |
| 0.028 | -0.145 | -0.085 |
| 0.031 | -0.150 | -0.085 |
| 0.033 | -0.160 | -0.075 |
| 0.036 | -0.155 | -0.070 |
| 0.039 | -0.160 | -0.070 |
| 0.042 | -0.175 | -0.065 |
| 0.044 | -0.180 | -0.055 |
| 0.047 | -0.185 | -0.050 |
| 0.050 | -0.170 | -0.050 |
| 0.053 | -0.155 | -0.040 |
| 0.056 | -0.175 | -0.040 |
| 0.058 | -0.180 | -0.055 |
| 0.061 | -0.190 | -0.075 |
| 0.064 | -0.180 | -0.080 |
| 0.067 | -0.155 | -0.085 |
| 0.069 | -0.135 | -0.070 |
| 0.072 | -0.155 | -0.080 |
| 0.075 | -0.190 | -0.080 |
| 0.078 | -0.205 | -0.090 |

Figure 10 Example of ECG Data from PhysioNet’s Chart-O-Matic

As can be seen in the Figure 10 example there are heading at the top of the columns of data. The initial formatting of the data removes these headings from the file (the complete formatting will be discussed in the next section. These files can be cut-and-pasted into a text file or one can email files to oneself. The “Chart-O-Matic” files contain one minutes worth of data; longer files are available, however, one minute ECGs should be long enough to test an ECG monitor. The data files can also be programmed to re-start the output over when the execution comes to the end and thus the signal can run continuously. The size of the flash memory being implemented on the Authentic ECG Simulator presented in this thesis is 66M bit (69,206,016 bits) [50]. This then becomes the limiting factor for the file size or the number of files that can be placed on the simulator’s memory (the NAND Flash Memory will be discussed in detail in Chapter 4 and 5). Also from Figure 10 one can see that there are three columns of data. The first column is the

time the signal was recorded which becomes the data for the x-axis on the plot. This first column data is not needed for the Simulator because it can be recreated based on the number of data points in the file and the time duration of the signal (one minute). The second column is labeled MLII which stands for Modified Lead II. This is the top voltage signal on Figure 9, which is plotted on the y-axis of the ECG plot. In most MIT-BIH records, the upper signal is a MLII, also referred to as Right Arm – Left Leg; this configuration is obtained by placing the electrodes on the chest in specific locations [51]. The MLII lead configuration places the LL electrode at the left iliac crest and the RA electrode in the infraclavicular fossa, medial to the border of the deltoid muscle and 2 centimeters below the lower border of the clavicle [38]. The third column of data is also voltage plotted on the y-axis and represents the lower signal on the plot. This is typically either a modified lead V1 or V5 arrangement; the electrodes are also placed on the chest but in different locations. This configuration of three columns of data is routinely used by the BIH Arrhythmia Laboratory; however some data files have only the first two columns of data. Normal QRS complexes are usually prominent in the upper signal (MLII signal, second column data) so this is the data that is implemented on the ECG Simulator prototype [51]. However, the third column of data could also be used if it contained important information not present in the MLII signal. The lead axis for the lower signal may be nearly orthogonal to the mean cardiac electrical axis (i.e., normal beats are usually biphasic and may be nearly isoelectric). Thus normal beats are frequently difficult to discern in the lower signal, although ectopic beats will often be more prominent [51]. It should be noted that the diagram of the signal data in Figure 10 only shows the data for a few dozen rows, however, the actual file data has thousands of rows of data (whatever is necessary for a one minute waveform). For

example a one minute waveform with a sample rate of 360 samples per second requires 21,600 rows of data for each column.

Formatting the MIT-BIH Data Files

The data in the MIT-BIH data files must be formatted so that it can be run through the digital to analog converter (DAC) available on the MSP430 microcontroller. The MSP430's DAC is discussed in detail in Chapter 5; however for the formatting of the data it is enough to know that the DAC is designed to receive unsigned integer type data with values from zero to 4095 [52]. Thus the voltage that is given by the MIT-BIH files as a decimal numbers (type floating point) must be converted to the correctly corresponding integer value. This is done by determining a scaling factor, and then multiplying the floating point data minus the minimum by the scaling factor and then converting the result to an unsigned integer. The formula for the scaling factor is:

$$S = \frac{2^n - 1}{(\text{Max} - \text{Min})} \quad (\text{Equation 3.1})$$

Where S is the scaling factor, n is the DAC resolution in bits (12 in this case), Max is the maximum data value in the data file and Min is the minimum value found in the data file. The formula for the new unsigned integer values that will be used by the MSP430's DAC is then given by the formula:

$$\text{intValue} = (\text{unsigned int}) S \times (\text{data} - \text{Min}) \quad (\text{Equation 3.2})$$

Where S is the scaling factor (Equation 3.1), $data$ is the data value from the MIT-BIH data file, and Min is the minimum data value in the MIT-BIH data file. To accomplish the formatting of the data a program was written in 'C' to format the data in any MIT-BIH one minute data file and place the new data along with the file size into a newly created text file. This new formatted file is then the file that must be stored onto the flash memory so that it can be read by the MSP430 and used by the DAC to produce the analogue voltage signal of a human heart on the ECG Simulator. An example of what the formatted file data looks like is given in Figure 11. The first row contains the file size (21,600 data points) the rows after this contain the voltage formatted to unsigned integer values (only a few dozen rows are shown).

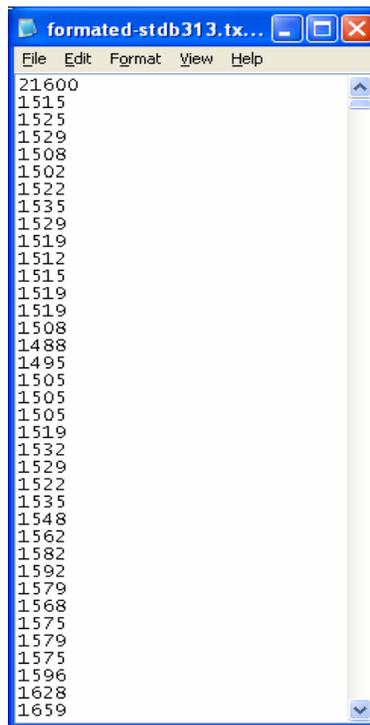


Figure 11 Example of a Formatted MIT-BIH Data File

The file size is necessary to calculate the sample rate ($\text{fileSize} / 60$) and is also necessary to calculate the number of pages that will be needed for storing the file on the NAND memory module. It is of course critical to the validity of the entire project that this formatting of the data is correct. Because of this a program was written in Matlab to plot both the original ECG data directly from the MIT-BIH data file and the integer data from the formatted data file. If the formatting has been done correctly the ECG signal waves will appear to be identical except for the amplitude. Figure 12 shows the output from the Matlab program for the MIT-BIH data file “stdb300.txt”.

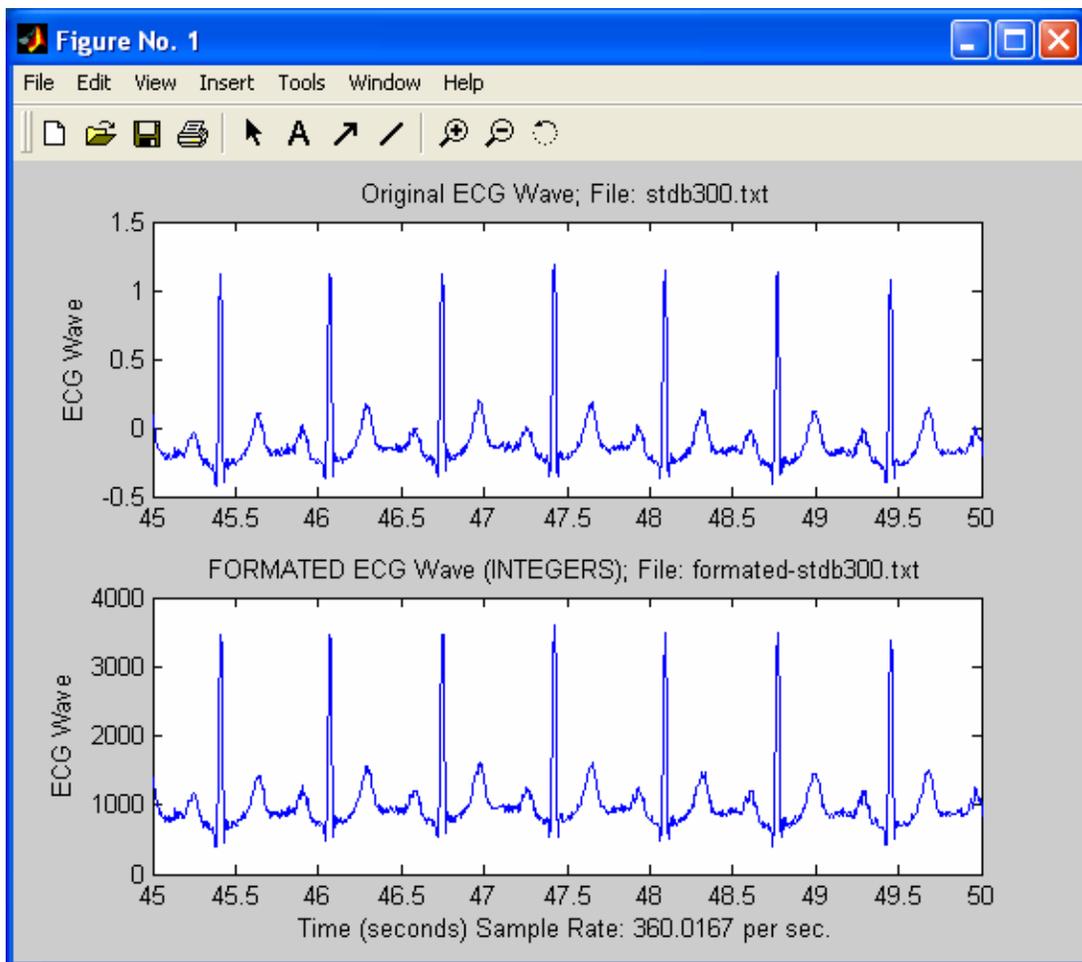


Figure 12 Plot of Formatted vs. Unformatted Data

The top ECG signal wave is produced using the original MIT-BIH data. The lower ECG signal wave is produced using the file data that has been formatted into integer values from zero to 4095. Figure 12 makes it clear that the plots are identical in form and thus the formatting of the data is correct.

Summary of the Use of MIT-BIH Database Files

Through the use of PhysioNet's "Chart-O-Matic" system a large number of ECG files are available (there are several hundred ECG files at [38]). These files can be easily and quickly obtained and formatted for use by the prototype ECG Simulator. The flash memory on the prototype ECG simulator will hold well over one hundred one minute MIT-BIH ECG data files. This means the simulator would be capable of testing for many different types of arrhythmia detection capabilities on monitors. Seven MIT-BIH ECG data files were formatted for use by the Authentic ECG Simulator prototype so they could be used to test the system. The next chapter discusses the primary hardware components that were necessary to meet the requirements for the Authentic ECG Simulator.

CHAPTER FOUR:

SYSTEM HARDWARE COMPONENTS

The purpose of this chapter is to explain why certain hardware components were chosen for the Authentic ECG Simulator, and to briefly explain the basic functionality, specifications, and implementation of the components. Before the individual components are discussed in detail an explanation of how the most important hardware requirements were met is provided along with an overview of the functionality of the Authentic ECG Simulator.

Meeting the Hardware Requirements

One of the most basic requirements for the Authentic ECG Simulator is that it be a portable device. The Authentic ECG Simulator being designed to be portable is important because this means it can be easily taken into any hospital or medical facility and used to check the ECG equipment on-sight (simulators that use MIT-BIH data that are not portable already exist). Portable in this case means that the device should be capable of being battery operated and should be easily carried by one person. The exact size is not extremely important, however, a small hand-held device is preferred. This means using a personal computer is out, and even a laptop size device would be larger than the preferred hand-held size. Based on this requirement the first hardware component that was specified for the system was a low-power microcontroller. Microcontrollers provide a very small package and some can be operated in the low-power battery operated range.

A second requirement of the system, that strongly affected the choice of the specific microcontroller, is the need to do digital to analog conversion. This is necessary in order to use the MIT-BIH data files. The Texas Instruments MSP430F169 microcontroller is ideal in terms of size and power consumption. Also the MSP430F169 has a built-in 12-bit digital to analog converter [53].⁵ Thus it was decided that the MSP430F169 microcontroller should be the heart and brains of the prototype Authentic ECG Simulator. The MSP430 microcontrollers are considered to be ultra-low power consumption devices and the model F169 can handle all of the processing requirements necessary for the ECG Simulator. The specifications of the MSP430F169 microcontroller will be covered in detail later in this chapter.

A third requirement of the Authentic ECG Simulator was that it be capable of storing a relatively large quantity of data. This is also necessary due to the use of the MIT-BIH data files. The on-chip memory of the microcontroller would not provide enough storage capacity to meet this requirement. What can provide a large data storage space is a flash memory chip. The advantages of flash memory are: first, the size is very small, second, flash memory is generally low power, and third, flash memory is non-volatile which means the data remains stored even when power is removed from the device. The flash memory chosen for the system was a NAND flash memory module [50]. The specific details of the NAND memory module will be discussed later in this chapter.

⁵ An independent D/A converter could also have been implemented. However, this would require sending the data from the NAND flash memory through the microcontroller and then to A/D converter and from there outputting the analog data. This would require more transfer time and the rates of transfer are not necessarily the same to the different components which would greatly complicate specifying the sample rate. A better solution was to find a microcontroller with a built-in A/D converter. This is why the MSP430F169 is the perfect choice for this system. The specific details of the 12 bit A/D converter will be discussed below under the section on the microcontroller.

A fourth requirement for the prototype Authentic ECG Simulator was that it provide for some type of user input interface that would enable choices and display options and status information. A large graphical user interface along with an entry keypad would be ideal for this, but due to the cost of such a unit it was decided that a small LCD and some input buttons would suffice for the prototype. When the MSP430F169 microcontroller was being pursued to meet the processing requirements a development board was found that came with a Samsung NAND flash memory module, and a small LCD along with 3 input buttons [54]. The LCD was a no-name 2 by 16 character display. Although the LCD has no part number the wiring schematic makes it clear that it is a parallel, Hitachi 44780 controller driven LCD. These are in fact the most common inexpensive LCDs on market today. Since this development board was very reasonably priced and met the most important requirement of having the MSP430F169 microcontroller and a NAND flash memory module and an LCD display it was purchased and used for the implementation of the ECG Simulator prototype. This development board is designed and manufactured by Olimex [54] and marketed through distributors Spark Fun Electronics [55]. The location of the Olimex Company is not provided on their web site and they did not respond to several emails. A picture of the development board is shown in Figure 13 (see Appendix for permission to use images). The biggest drawback of the board was the fact that almost no documentation is available for the board. A wiring schematic is the only available documentation on the Olimex web site, and unfortunately the schematic doesn't even have all of the component part numbers on it (nothing is given for the LCD). Two obvious improvements over the Olimex board that would be necessary for a full featured Authentic ECG Simulator would be a larger LCD unit and an alphanumeric keypad input. However, these are not necessary for the prototype simulator.



Figure 13 Olimex Development Board

A final requirement of the Authentic ECG Simulator is to provide the 4 lead ECG Network that is the output of the system. These 4 leads are what connect to the ECG monitor under test.

Based on lead types I or II, the 4 lead network will simply require the output from the DAC to be connected to the right arm lead, and the other three leads (left arm, right leg and left leg) to be connected to the common ground. If a lead type III network is desired this would require the output from the DAC to be connected to the left arm lead, and the other three leads (right arm, left leg, and right leg) to be connected to the common ground.

The Functionality of the Prototype

The purpose of the Authentic ECG Simulator is to provide an authentic human ECG voltage signal on the four lead output that can be used to test an ECG monitor. In order to do this first several MIT-BIH ECG data files are stored in the NAND flash memory module of the Simulator.

After the files are stored to the NAND flash the user can run the actual simulator application software. When the Simulator is started the user will first be given some options for choosing a data file and then choosing an amplitude and frequency at which to run the signal. The Simulator is then activated and the file data is applied to the digital to analog converter to create the ECG voltage signal. The digital to analog converter transforms the digital representations of the data into an analog voltage and finally that single voltage is run through the four lead network.

Description of Hardware Components

In the introduction of this thesis the system hardware was shown in Figure 2 in Chapter One and is reproduced here in Figure 14. The primary hardware components, and those that will be discussed in detail here, are the MSP430 microcontroller, the digital to analog converter on the microcontroller, the hardware necessary to program the MSP430 (not shown in the figure), the Samsung NAND flash memory chip and the input buttons and LCD display. The four lead ECG network was discussed in the “Meeting the Hardware Requirements” section and does not necessitate further explanation.

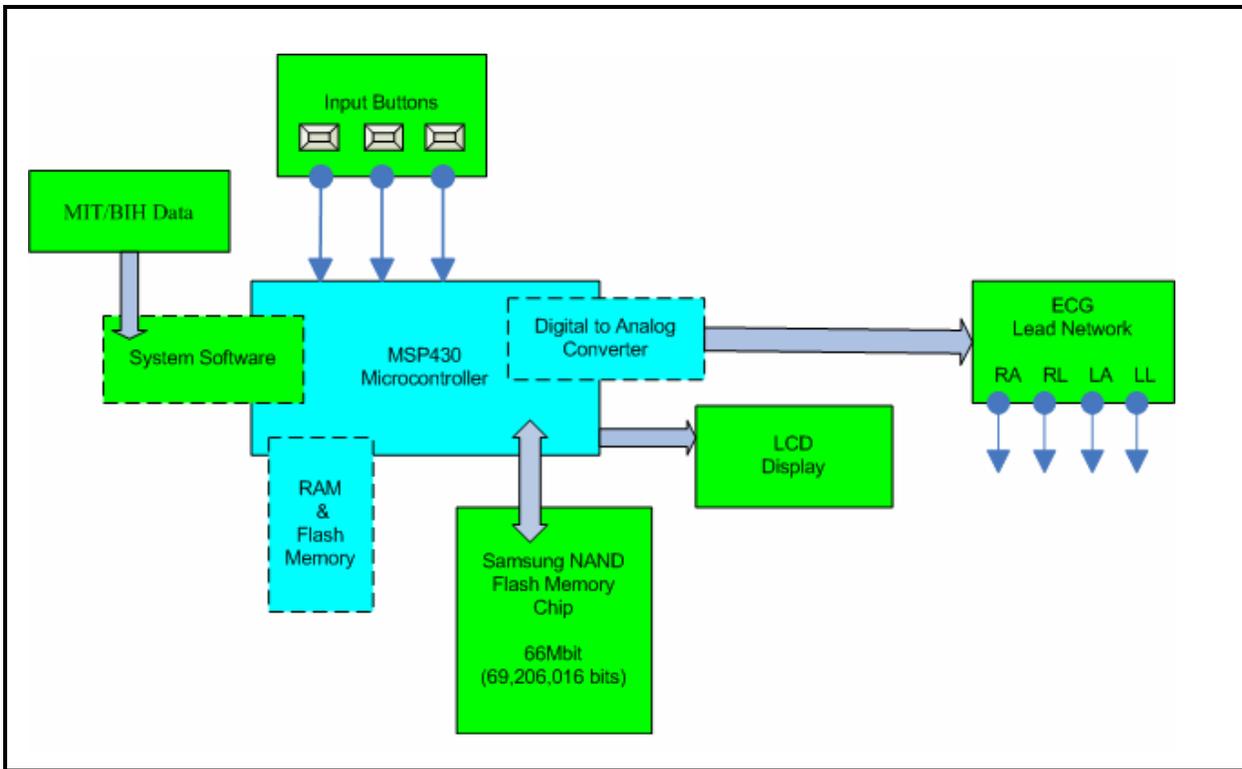


Figure 14 High-level System Design Block Diagram

Texas Instruments MSP430F169 Microcontroller

The Texas Instrument's MSP430 microcontroller family has been around for about 10 years. The primary usage was originally measurement applications which were battery powered, (intelligent sensors) with or without LCD-displays, but today the usage is becoming much broader. In the past, the development tools for the MSP430 were not very attractive; however, today the tools available are quite extensive, especially due to the introduction of JTAG-based programming and debugging [56]. All of the MSP430 devices today are in-circuit programmable via JTAG or BSL (Boot Strap Loader via RS-232 link). The MSP430 family is built around a modern 16-bit RISC CPU, 16-bit registers, and constant generators that attribute to maximum code efficiency and high execution performance [57]. The MSP430 comes in a

variety of configurations featuring all kinds of peripherals – analog to digital converters, a digital to analog converters, comparators, timers, USARTs, LCD drivers, watchdog, HW multiplier, internal oscillator, etc. One of the greatest strengths of the MSP430 line of microcontrollers lies in the fact that they are considered to be the industry’s lowest power 16-bit microcontrollers [58]. The MSP430 model implemented in this project was the F169. The architecture of the F169 supports five low power modes to achieve extended battery life in portable applications. The digitally controlled oscillator allows wake-up from low-power modes to active mode in less than 6 μ s. The current drawn when the MSP430 is in idle mode is only a few micro amps. Thus very few microcontrollers can match the MSP430 in terms of power consumption. The F169 series is configured with two built-in 16-bit timers, a fast 12-bit A/D converter, dual 12-bit D/A converters, one or two universal serial synchronous/asynchronous communication interfaces (USART), I2C, DMA, and 48 I/O pins [53]. The block diagram of the TI MSP430F169 microcontroller is shown in Figure 15 (see Appendix for permission to use images). The MSP430-F169 series has 60KB + 256B flash memory, and 2KB RAM memory [57]. This is much more memory than the MSP430 had just a couple years ago, however, it is still not large enough for the MIT-BIH data files. This is why the additional NAND flash memory was a must for this application. The size of the F169 chip is slightly smaller than the size of a dime, the cost per chip is less than ten dollars depending on the quantity ordered.

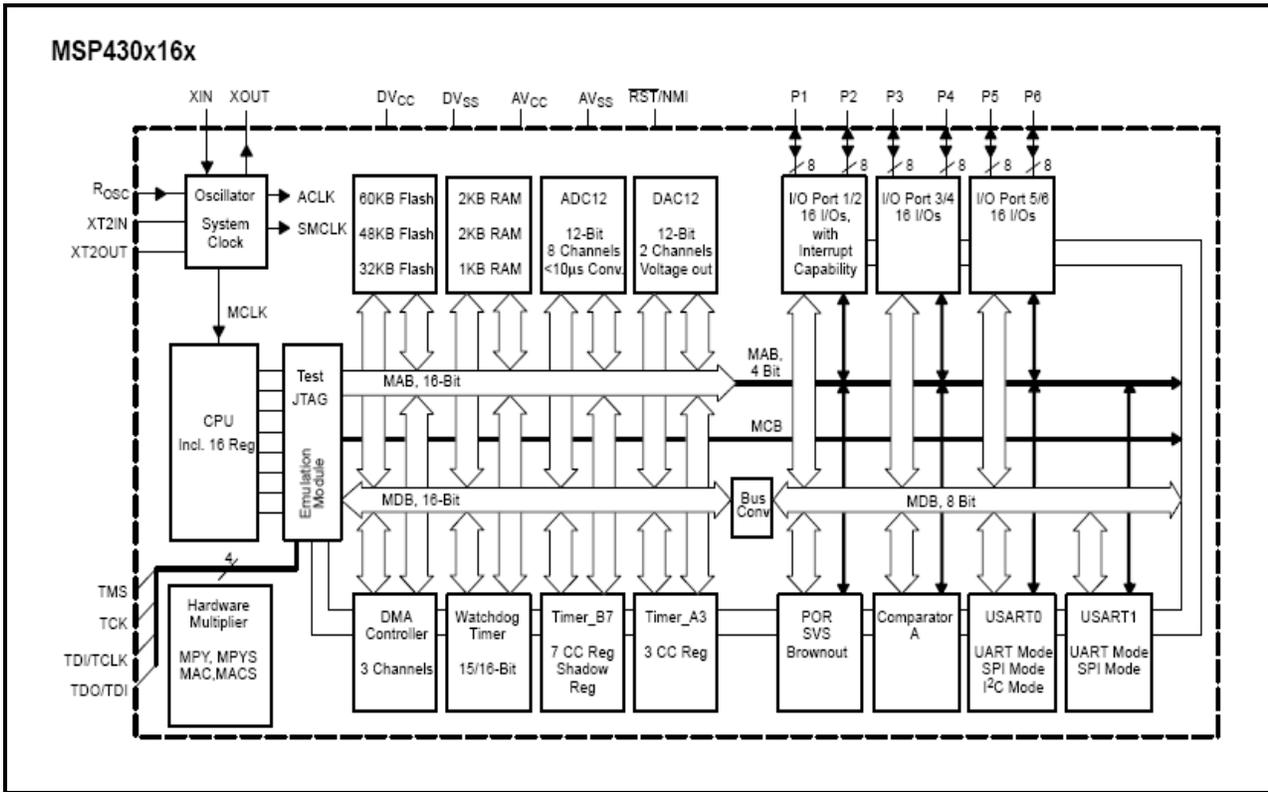


Figure 15 TI MSP430-F169 High Level Block Diagram

IAR Systems provides software development tools that can be downloaded for free from the Texas Instruments web site [53]. The IAR development tools are very popular and very well designed. However, the free version provides only a crippled C/C++ compiler that limits the number of lines that can be compiled. This means that any of the larger C libraries can not be used on this version. This was a problem for this project because it is necessary to do file input in order to access the MIT-BIH data file. The file input is not supported by the free version, however IAR Systems did provide a less restricted version of the software that was able to support input from a file [59]. The IAR Systems software for the MSP430 is further discussed in Chapter 5 – Developing the Software.

Digital to Analog Converter

A digital to analog converter is a device that converts digital code (numbers) into analog signals – voltage in this case. The primary goal of this project is to convert the digital code of the MIT-BIH data files into analog signals that replicate a human heart's ECG signal. The MSP430F169 has a digital to analog converter (DAC) built-in to it. The DAC is a 12-bit monotonic output type DAC. A monotonic DAC has an output that changes in the same direction (or remains constant) for each increase in the input code. The converse is true as well for decreasing input code [60]. The DAC12, as it is referred to by Texas Instruments, is an R-ladder type DAC. This means it is a binary weighted DAC that creates each value with a repeating structure of 2 resistor values, R and R times two. This improves DAC precision due to the ease of producing many equally matched values of resistors or current sources, but lowers conversion speed due to parasitic capacitance [61]. The DAC12 may be used in a 12-bit or an 8-bit mode. The 8 bit mode only reduces the resolution so this project chose to implement the DAC12 in the 12-bit mode [62]. A DAC's resolution refers to the number of possible output levels the DAC is designed to reproduce. This is usually stated as the number of bits it uses, which is the base two logarithm of the number of levels. For instance a 1 bit DAC is designed to reproduce 2^1 levels while an 8 bit DAC is designed for 2^8 (256) levels – the more levels the higher the resolution. Other DAC options available are: choosing settling time speed versus power consumption (faster settling time requires more power), using an internal voltage reference or an external voltage reference, using straight binary or two's complement data format, a self-calibration of offset voltage, and the ability to use two DAC12s synchronized (the MSP430F169 actually has two DAC12s) [62]. All of these options, or configurations, are chosen using the DAC12 Control Register (DAC12_xCTL). The Control Register's 16 bits are configured using the

microcontroller's software. The other register related to the DAC is the data register. The data register is also a 16-bit register; however, the DAC12 uses only bits zero through eleven [62]. The DAC12 data register is designed to receive unsigned integers from zero to 4095 (2^{12}). This is why the data in the MIT-BIH data files had to be re-formatted (see Chapter 3, Formatting the MIT-BIH Data Files). More of the details on the configuration of the control register and the use of the data register are discussed in Chapter Five – Developing the Software.

The JTAG Interface Hardware

The JTAG, as it is called by Texas Instruments, is the hardware device that connects the MSP430 to the parallel port of a PC to the microcontroller in order to download the application software and any other special data to the microcontroller. Figure 16 is a picture of the JTAG hardware device (see Appendix for permission to use images) [64]. All MSP430 flash microcontrollers can use the standard 2 x 7 pin JTAG connector.



Figure 16 JTAG Hardware

The JTAG itself requires no external power supply; the power is taken from the parallel port of the PC.

Samsung NAND Flash Memory

The Samsung NAND 8M x 8 Bit flash memory chip was already on the development board (Olimex) selected for the MSP430F169. The capacity of the flash memory is (8M + 256K) x 8bit which translates to 69,206,016 bits. The memory is organized as 16,384 rows (or pages) by 528 columns – each column being one byte. That produces 8,650,752 bytes, but only 8,388,608 bytes are deemed usable. Whole memory blocks (16 pages) can be invalid (unusable) due to the presence of one invalid bit [50]. The chip can come from the factory with some invalid blocks and more invalid blocks may develop. However, even with the losses due to invalid blocks the chip can still hold a remarkable quantity of data. The typical one minute MIT-BIH data file is 43,200 bytes (21,600 integers). This translates to 81 full pages of memory plus one partial page of memory (for one MIT-BIH data file). The NAND capacity in pages is 16K (16,384 pages) thus if there was no memory space lost due to fragmentation this would be enough for almost 200 one minute data files. The actual number of files that can be written into the NAND will be less due to fragmentation issues – the data needs to be stored contiguously so it can be accessed quickly. The bottom line is that the size of the NAND flash memory is more than sufficient for the prototype ECG Simulator. The unit runs on a voltage of anywhere from 2.7 to 3.6 volts [50]. The programming, or writing, of data to the NAND must be done on a single page basis. Writing one (528 byte) page of data to the NAND takes approximately 200 microseconds. An erase operation must be done on a block basis (16 pages) and takes approximately 2 milliseconds. Any area of the memory array to be written to must first be erased. Writing over

data without first doing an erase is prohibited. Reading the data back out of the NAND flash memory requires a 10 microsecond delay per page and then the data comes out at a rate of one byte each 50 nanoseconds [50]. This rate of transmission is sufficient for the MIT-BIH data files which have a maximum sampling rate of 360 data points per second (some are much slower). The Samsung NAND also has a power saving mode – inactive – which reduces its power consumption when it is not reading, writing, or erasing [50]. The NAND Flash Memory module (NAND) is a low-level memory device external to the MSP430 microcontroller. The fact that the device is low level provides the advantage of a faster data transfer rate between the memory chip and the microcontroller, however, the disadvantage is that the programming of the device is far more involved. The programming of the NAND is discussed in Chapter Five – Developing the Software.

The LCD Display

The LCD implemented in the project came as a part of the Olimex Development board. It has no identifying numbers or labels so its origin was originally unknown, however, based on the wiring diagram, it was determined to be a Hitachi 44780 controller driven parallel LCD. A zoomed in picture of the schematic for the development board focused on the LCD is shown in Figure 19 (see Appendix for permission to use images) [54]. The wiring of the LCD makes it clear that the unit is a parallel LCD 2X16 character display that is wired to receive its data on four input pins DB4 – DB7 (the heavy blue line connects the LCD to the MSP430 port 4). The development board that holds the LCD was purchased from Spark Fun Electronics and on their website a “Basic 16x2 Character LCD” is available that looks exactly like the LCD on the board [65]. This LCD unit, shown in Figure 20, is the same unit on the board (see Appendix for permission to use

images). The unit has a backlight that is turned on and off using pins 15 and 16. The "R/S" bit is used to select whether data or an instruction is being transferred between the microcontroller and the LCD. If the R/S bit is set to one, then the byte at the current LCD "Cursor" Position can be read or written. When the R/S bit is set to zero, either an instruction is being sent to the LCD or the execution status of the last instruction is read back (whether or not it has completed).

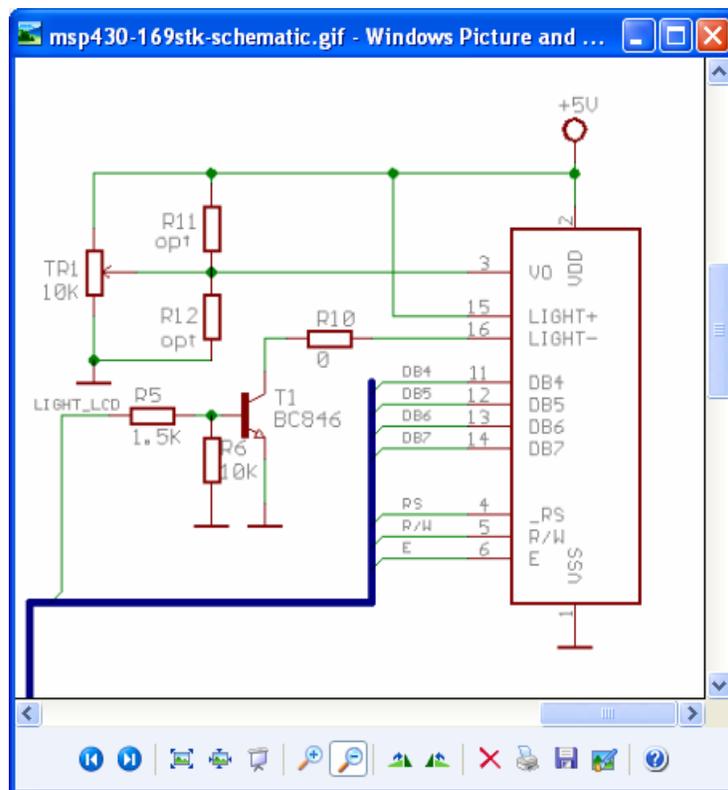


Figure 17 LCD Wiring Schematic



Figure 18 Sparks Fun Electronics Basic 16 X 2 Character LCD

The unit is wired in the 4 bit mode which means two nibbles of data (send high four bits and then low four bits with an "E" clock pulse for each nibble) are sent to make up a full 8 bit transfer.

The "E" clock is used to initiate the data transfer within the LCD. The "R/W" pin is used to poll the busy flag on a read or write to determine completion. This bit is set while the LCD is processing [66]. The programming of this LCD is discussed in chapter Five – Developing the Software.

CHAPTER FIVE:

SYSTEM SOFTWARE

One of the primary aspects of the Authentic ECG Simulator prototype is the software. This chapter discusses the development of the software based on the operations necessary for the simulator. The software development tools are briefly discussed first and then the system operation as a whole is discussed. Finally, the development of the software is covered based on the major hardware components of the system. The main components of the software are the operations of the LCD and input buttons, the operations of the Samsung NAND Flash Memory module, the operations of the digital to analog converter (DAC12), and finally the operations of the user input options for selecting a data file, the frequency and the amplitude.

The Software Development Tools

The development software tools for the programming of the MSP430 were obtained from IAR Systems on an evaluation basis. The evaluation basis means the software can be used for a period of time for free (30 days). IAR Systems provides a wide range of development tools for embedded systems: integrated development environments (IDE) with C/C++ compilers and debuggers. IAR Embedded Workbench for MSP430 is an integrated development environment for building and debugging embedded applications. The software provides an interface common to the assembler, a C/C++ compiler, a project manager, an editor, and a build and debugger tool [59].

A few of the key components of the software development tools are:

- “Optimizing compiler supporting C and C++
- Configuration files for all MSP430 devices
- Debugger support
- Run-time libraries
- Relocating MSP430 assembler
- Linker and librarian tools
- C-SPY debugger with MSP430 simulator and support for RTOS-aware debugging on hardware [59].”

The code for the ECG simulator was written in embedded C. Embedded C is the most widely used high-level programming language used in the embedded systems industry. Using the MSP430 IAR C/C++ compiler, one can build freestanding applications that follow the standard ISO 9899:1990 [67]. This standard is commonly known as ANSI C. When using the C language for programming the MSP430 the hardware model device specific file must be included (ie: `#include <msp430x16x.h>`).

There is one weakness in the IAR Systems development software which even IAR Systems tech support admitted to – the software is not designed for data file input. It will do file input, however, only at a very slow rate. Because of this storing the data files to NAND flash memory takes a long time (2 hours for 21,600 integers). This creates a weak point for the entire simulator system. Not only can it take over two hours to store a one minute data file on to the flash memory but also the software will often crash (about 50% of the time) due to over-running the stack. Both of these problems can only be over come with better development software, which might be available from a different software vendor. However, the time invested in obtaining and learning the IAR System’s software prohibited the testing of other development platforms.

System Software Overview

Since the Authentic ECG Simulator prototype requires the implementation of several different software applications in order to understand how these different programs interact a view of the simulator system from a higher level of abstraction is necessary. The functional block diagram in Figure 21 provides this view and illustrates how the different software applications interface to produce the overall system of the Authentic ECG Simulator. The process begins with obtaining an MIT-BIH ECG data file from the Physionet web site (top box in Figure 21). This was discussed in detail in Chapter Three. The MIT-BIH data obtained is then formatted (from floating point numbers into unsigned integers) by the formatting software (second box in Figure 21). The first software application that actually executes on the MSP430 is the program (data_to_NAND in Figure 21) this program stores the formatted data on to the NAND flash memory module. Each execution of the software will store one MIT-BIH data file (converted from unsigned integers into bytes) on the NAND. The user who executes this program should record in a log what MIT-BIH data file has been placed on the NAND along with the size of the file in terms of memory pages used and where on the memory the file is located (what memory page does the data begin at). Once the data is placed on the NAND memory the software to verify that the file has been correctly stored should be executed (program: verify_data). This program reads the data from the NAND, converts the data back into unsigned integers and prints them to the development software's I/O Terminal window. The user can visually compare the data read from the NAND with the original data. This data can also be run through a second program (program: Plot_ECG) that plots the integer data in Matlab along with a plot of the original data for comparison purposes (refer to Chapter Three for further details).

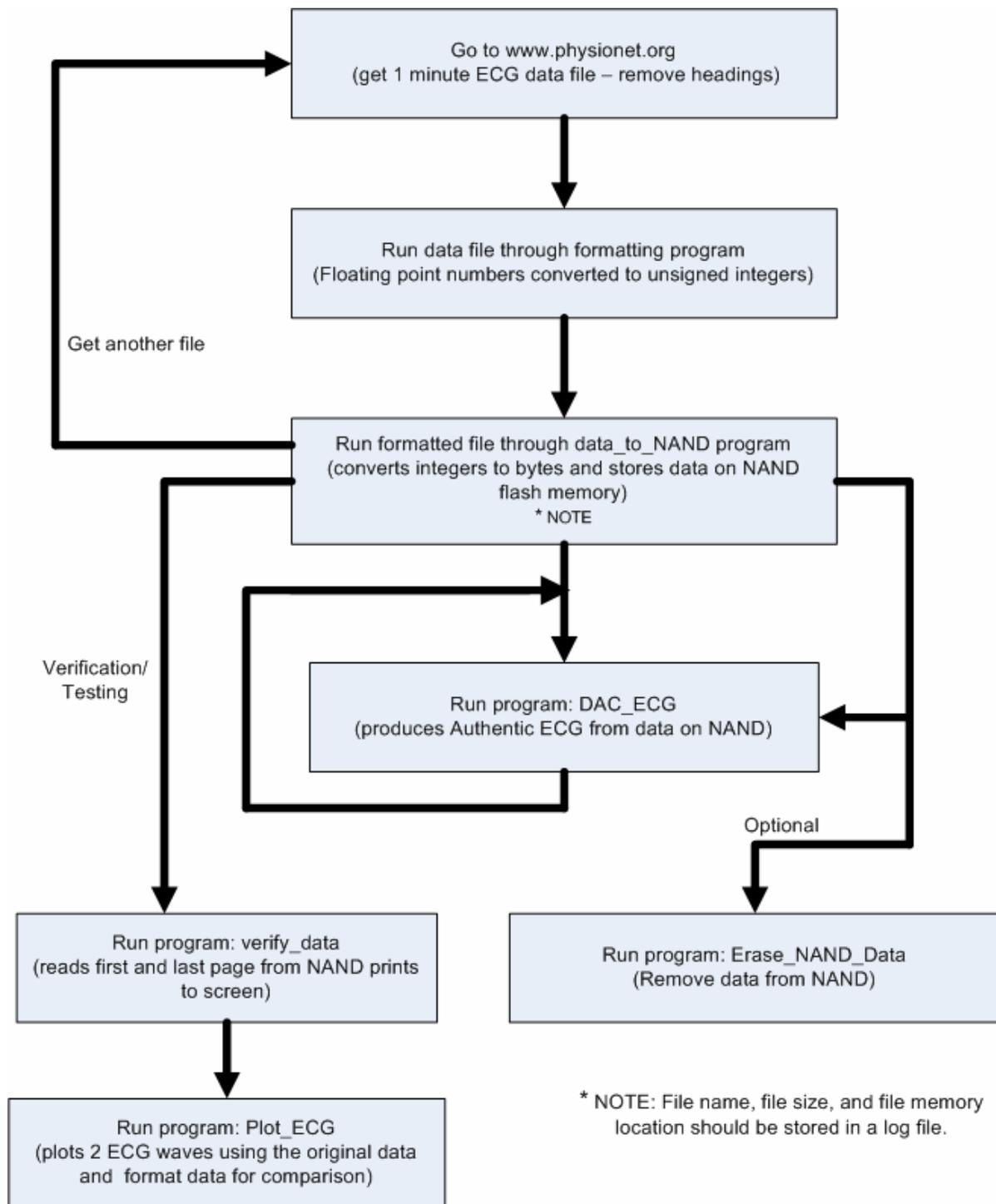


Figure 19 System Functional Block Diagram

This verifies that the data is on the NAND in the know specific location and the data is correctly formatted. Once the user is confident the data is correctly stored on the NAND he or she can start this process all over again to store another data file on the NAND. Seven MIT-BIH one minute data files were stored on the simulator prototype (many more could be added). Now having the data stored on the NAND the application software that actually runs the data through the DAC and produces the ECG signal can be executed (program: DAC_ECG). The other software application that can be executed is one to erase data from the NAND (program: Erase_NAND_Data). This would be used when the NAND was full and one or more data files needed to be removed so others could be stored on the NAND. The following sections will discuss some of the details of programming many of the aspects of the major components of the software applications just presented.

Programming the Input Buttons and LCD Display

The LCD and input buttons are both major aspects of all of the software run on the MSP430. Together they provide the user interface necessary for prompting for input choices, receiving inputs, and displaying the current status of the system. In order to make the LCD and the input buttons easier to use without constantly rethinking all that goes into each use of the LCD or the input buttons, define statements were used to specify variables that would be more self-explanatory. For example the input buttons are defined in terms of their port number and pin number and the port direction (input). In 'C' this is done with the following statements:

```
// Three push buttons on board defined as inputs on port ONE, pins 5, 6 and 7.
#define button1    BIT5 & P1IN        // button1 - Port 1.5
#define button2    BIT6 & P1IN        // button2 - Port 1.6
#define button3    BIT7 & P1IN        // button3 - Port 1.7
```

These statements now place the buttons in the default state. The default state of the buttons (not pushed) is one, when they are pushed (held down) the variables (button1, button2, button3) become zero. The moment the button is released the variable returns to a one. These variables, along with flags that retain the last status change, can now be used in conditional statements that test the status of the buttons and execute statements based on that status. Along the same lines the LCD inputs are defined using the following statements:

```
// LCD input lines (prewired):
#define E_HIGH      P4OUT |= BIT1          // Enable transition to high
#define E_LOW       P4OUT &= ~BIT1        // Enable transition to low
#define RS_HIGH     P4OUT |= BIT3         // set LCD to data mode
#define RS_LOW      P4OUT &= ~BIT3       // set LCD to command mode
#define LCD_Data    P4OUT                 // Sending data out on port 4 to LCD
#define LCD_LIGHT_ON P4OUT |= BIT0       // Turns on the LCD light
#define LCD_LIGHT_OFF P4OUT &= ~BIT0    // Turns off the LCD light
```

The LCD commands for the Hitachi 44780 controller are given in hexadecimal numbers [67].

To make the source code more readable these commands can also be defined in terms of what they will do to the LCD display. The following are the most common LCD commands and thus they are defined for the ECG Simulator prototype:

```
// Commands to control LCD Display
#define DISP_ON      0x0c                // Turn on LCD
#define DISP_OFF     0x08                // Turn off LCD
#define CLR_DISP     0x01                // Clears LCD Display
#define CUR_HOME     0x02                // Move LCD cursor to home
#define DD_RAM_ADDR  0x80                // Set-Up for new message
#define DD_RAM_ADDR2 0xc0                // Move cursor to row 2
```

In order to make the LCD easier to use and to be able to send complete messages to the LCD functions for the LCD were written.

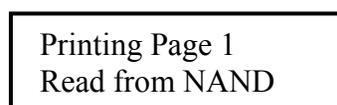
The following are the function prototypes used for the LCD display on the ECG Simulator:

```
// LCD Function Prototypes
void Delay (unsigned int a);           // Creates delay
void Delayx100us(unsigned char b);    // Creates longer delay
void longDelay(unsigned int c);       // Creates longest delay
void LCD_enable(void);                // Sends high->low to enable pin on LCD
void SEND_CHAR_LCD (unsigned char d); // Sends one character to LCD
void SEND_CMD_LCD (unsigned char e);  // Sends a command to the LCD
void InitLCD(void);                   // Initializes LCD display
void full_LCD_message(const unsigned char array[]); // send 32 chars to LCD
void oneLine_LCD_message(const unsigned char array[]); // send 16 chars to LCD
```

The first three functions merely create execution delays of varying duration. These can be used when ever a control signal must be present for a specific length of time or for the display of an LCD message that only displays for a specified duration. Since every input sent to the LCD requires the LCD enable bit to transition from high to low the function *LCD_enable* was written to provide for this. The function *SEND_CHAR_LCD* sends the data across the data lines necessary for one ASCII character. The function *SEND_CMD_LCD* sends one command to the LCD. The function *full_LCD_message* sends a two line message to the LCD which is 32 characters. A complete 32 character message is defined in the following manner:

```
const unsigned char ReadMessage1[] = " Printing Page 1 Read from NAND";
```

A message is displayed on the LCD by simply calling the *full_LCD_message* function with the message name as its argument. The statement, “*message_to_LCD(ReadMessage1);*” displays the text on the LCD display as shown in Figure 22.



```
Printing Page 1
Read from NAND
```

Figure 20 LCD Display Text

The function *oneLine_LCD_message* sends a one line message to the LCD which is 16 characters. Developing these types of functions makes the LCD display much more usable and thus the LCD can be used much more frequently. The functions also make the source code more readable and easier to document.

Programming the NAND Flash Memory Module

Programming the read, write, and erase operations of the Samsung NAND turned out to be one of the greatest challenges of the simulator prototype. The data sheet for the Samsung NAND is poorly written which makes it difficult to follow. It has many incomplete sentences and seemingly no organizational or topical structure. The information necessary to program the NAND is scattered through out the 30 page data sheet. This coupled with the fact that the NAND module is a low level device made the programming of the device quite a challenge. The NAND chip can receive only a single byte command, or a single byte address or a single byte of data at a time (every input to the device must be individually sent as an electrical signal and only one byte of data can be sent to the device at a time). All of these are received (or sent for data) via the same single port (I/O₀ – I/O₇) [51]. In order to send a command, or address, or data byte, to the NAND, first all of the control pins on the chip must be properly set. This is called setting the mode. The commands are then used to read, write (program), or erase areas of the memory. The command set for the NAND is shown in Table 1 [50]. Commands are given as hexadecimal numbers. The commands (or addresses or data) are executed on the NAND by bringing the write enable (WE) low while the chip enable (CE) is low – data is latched on the rise of the WE. Any undefined command inputs are prohibited [50]. Before anything can be sent to the NAND first

the mode must be configured; then either an address, or a command, or a byte of data is transmitted to or from the NAND chip.

Table 1 Command Set for NAND Memory

| Function | 1st Cycle | 2nd Cycle | Description |
|-----------------|-----------------------------|-----------------------------|---|
| Read 1 | 00h / 01h | - | A page of data (528 bytes) is transferred to the data registers in less than 10ms. |
| Read 2 | 50h | - | Data is transferred from the spare area (bytes 512-527) to the data register |
| Reset | FFh | - | Reset will abort a read, program or erase operation. The data will be partially read, programmed or erased. |
| Page Program | 80h | 10h | The device is programmed on a page basis (528 bytes or less at a time). |
| Block Erase | 60h | D0h | The device erases data from its storage based on a block basis (16 pages at a time). |
| Read Status | 70h | - | The Status Register may be read to find out whether the program or erase operation is completed and completed successfully. |

The entire process of configuration, then placing a command, or address, or byte of data on the I/O port, and then latching the I/O port must be done thousands of times in order to place data on the NAND or read data from the NAND. Because of this, the implementation of functions that perform these operations is absolutely necessary. The NAND control pins and the descriptions of these functions are listed in Table 2 [50]. When all of these pins are set for a particular configuration this places the NAND chip in a operational mode. The valid modes for the NAND are shown in Table 3 [50].

Table 2 Pin Descriptions for NAND Flash Memory

| Pin | Function Description |
|-------------|---|
| I/O0 - I/O7 | Data Input/Output: The I/O pins are used to input commands, addresses and data, and to output data during read operations. The I/O pins float to high-z when the chip is deselected or when the outputs are disabled. |
| CLE | Command Latch Enable: The CLE input controls the activating path for commands sent to the command register. When active high, commands are latched into the command register through the I/O ports on the rising edge of the WE signal. |
| ALE | Address Latch Enable: The ALE input controls the activating path for address to the internal address registers. Addresses are latched on the rising edge of WE with ALE high. |
| CE | Chip Enable: The CE input is the device selection control. When the device is in the Busy state, CE high is ignored, and the device does not return to standby mode in program or erase operation. |
| RE | Read Enable: The RE input is the serial data-out control, and when active drives the data onto the I/O bus. Data is valid to READ after the falling edge of RE which also increments the internal column address counter by one. |
| WE | Write Enable: The WE input controls writes to the I/O port. Commands, address and data are latched on the rising edge of the WE pulse. |
| WP | Write Protect: The WP pin provides inadvertent write/erase protection during power transitions. The internal high voltage generator is reset when the WP pin is active low. |
| R/B | Ready/Busy: The R/B output indicates the status of the device operation. When low, it indicates that a program, erase or random read operation is in process and returns to high state upon completion. It is an open drain output and does not float to high-z condition when the chip is deselected or when outputs are disabled. |

Table 3 Mode Selection Configurations for NAND Memory

| CLE | ALE | CE' | WE' | RE' | WP' | NAND Mode |
|-----|-----|-----|-----|-----|-----|-------------------------------|
| H | L | L | ↑ | H | X | Read Command Input |
| L | H | L | ↑ | H | X | Read Address Input (3 clock) |
| H | L | L | ↑ | H | H | Write Command Input |
| L | H | L | ↑ | H | H | Write Address Input (3 clock) |
| L | L | L | ↑ | H | H | Data Input (write to NAND) |
| L | L | L | H | ↓ | X | Data Output (read from NAND) |
| X | X | H | X | X | Vcc | Stand-by (Inactive) |

As with the LCD, define statements are used to make the programming of the NAND a little easier and to make the source code more readable. The define statements used for the NAND

are:

```

#define CE_low      P2OUT &= ~BIT0    // CE Set to ON (it's active low)
#define CE_high    P2OUT |= BIT0      // CE set to OFF
#define RE_low     P2OUT &= ~BIT1    // RE set to ON (it's active low)
#define RE_high    P2OUT |= BIT1      // RE set to OFF
#define WE_low     P2OUT &= ~BIT2    // WE set to ON (WE is active low)
#define WE_high    P2OUT |= BIT2      // WE set to OFF
#define ALE_high   P2OUT |= BIT3      // ALE set to ON (ALE active high)
#define ALE_low    P2OUT &= ~BIT3    // ALE set to OFF
#define CLE_high   P2OUT |= BIT4      // CLE set to ON (CLE active high)
#define CLE_low    P2OUT &= ~BIT4    // CLE set to OFF
#define R_B        P2IN & BIT7       // Port 2.7 connects to R/B on NAND

// Control Commands for NAND Memory, (see NAND datasheet)
#define READ_2     0x50                // 50h = READ_2 command
#define READ_0     0x00                // 00h = starting address of 1st half of reg.
#define READ_1     0x01                // 01h = starting address of 2nd half of reg.
#define READ_STATUS 0x70                // 70h = Read Status Reg.
#define WRITE_PAGE 0x80                // 80h = Sequential Input Command
#define WRITE_AKN  0x10                // 10h = 2nd Cycle Page Program
#define ERASE_BLOCK 0x60                // 60h = Block Erase
#define ERASE_AKN  0xD0                // D0h = 2nd Cycle Block Erase

```

The names of the functions necessary to make the read, write, and erase operations of NAND memory usable follow; each function is discussed briefly in turn below.

```

void Inactive_Flash(void);           // Pulls flash pins to inactive state
void Send_to_NAND(unsigned char a);  // send command, address or data
unsigned char Get_from_NAND(void);    // gets 1 byte from NAND
unsigned char Erase_Flash(unsigned char BLOCK_ADDL, unsigned char
BLOCK_ADDH);
unsigned char WRITE_to_NAND(unsigned char COL_ADD, unsigned char
ROW_ADDL, unsigned char ROW_ADDH, unsigned int NUMBER);
void Read_from_NAND(unsigned char COL_ADD, unsigned char ROW_ADDL,
unsigned char ROW_ADDH, unsigned int NUMBER);

```

```
// Data Conversion Prototypes
void integer_2_bytes(unsigned int cnt); // change array of ints to array of bytes
void bytes_2_integers(unsigned int size); // change array of bytes to array of ints
```

The first function listed, *Inactive_Flash*, sets the control pins for the NAND to a configuration necessary to place the NAND into the inactive mode. This is done to save power consumption on the NAND and to create a known starting point for all the NAND pins. The second function, *Send_to_NAND*, places one byte of data on the NAND's data port (I/O₀ – I/O₇). This byte of data can be either a command sent to the NAND, or an address location on the NAND, or a byte of data to be stored to the NAND. The function *Get_from_NAND* sets up the microcontroller and the NAND for one byte of data to be received from the NAND by the microcontroller. This byte of data is typically the data that has been stored on the NAND; however, the function is also used to get the “status data” from the NAND after an erase or a write. The status is either one or zero, indicating whether the operation completed error free or not. The function *Erase_Flash* is used to erase a block of data on the NAND. The *Erase_Flash* function takes two single byte arguments (*address_Low* and *address_High*) which together are used to define the address of the block to be erased. Nothing less than an entire block of data can be erased (1 block = 16 pages). This can pose a problem if a single block contains the data from two or more MIT-BIH files and one only desires to erase one of the files (this is not possible). Thus this limitation should be kept in mind when the data is stored on the NAND. The function *WRITE_to_NAND* is the function that stores data onto the NAND memory chip. *WRITE_to_NAND* takes four arguments. The first three arguments are addresses and the fourth is the quantity of data to be written to the NAND (number of bytes). Both reading from, and writing to, the NAND requires three separate

bytes of information to define the memory location. The first address is the column address within the specified page. This is a byte's location within a page (0 – 528) and is zero if one wants to start storing data at the very beginning of a page (always in this application). The next two arguments (*address_Low* and *address_High*) together provide the address of the page on the NAND chip. These addresses can be specified in decimal from zero to 255 (one byte). The function *WRITE_to_NAND* returns one byte of data back to the microcontroller upon the write completion which indicates the status of the operation. A '1' indicates the successful programming of NAND and a '0' indicates an error took place in the programming operation. Only one page of data can be written to the NAND memory at a time and then the page number must be incremented (or changed) before the next page is written. The function *Read_from_NAND* retrieves data from the NAND memory placing it into an array. The *Read_from_NAND* function takes the same four arguments as the *WRITE_to_NAND* function discussed above and they provide the same purpose. The *Read_from_NAND* function does not return anything because the data retrieved is placed into a global array. The data sheet for the NAND discusses a “sequential row read” that sounds like it should continue to read sequential pages of data from the NAND without stopping, however, the technical support person for the NAND recommended that a “page read” operation be used (only one page of data read at a time, placed in a for loop incrementing the page number) so this is what was used to read the data from the memory.

There are two other functions that are critical to the use of the NAND memory module in this application. They are named *integer_2_bytes* and *bytes_2_integers*. These functions are necessary because the application needs to store integers on the NAND memory and yet the

NAND memory can only store data in terms of bytes (one byte at a time). Thus the integers from the formatted MIT-BIH data file must be converted into bytes (one integer produces two bytes) and then stored on the NAND memory. And when retrieving the data from the NAND the opposite conversion is necessary. These functions each work on a memory page sized array of either integers or bytes converting that array of data into a new array of the contrasting data.

Programming the DAC12

As discussed under the hardware section the two digital to analog converters (DAC12s) available on the MSP430 each have two registers that facilities there implementation. The control registers called DAC12_0CTL or DAC12_1CTL and the data registers called DAC12_0DAT or DAC12_1DAT [62]. This application only requires the use of one of the DAC12s so only the zero control and data requesters will be discussed (both DAC12s function identically and have the same configuration bits). The data register takes the data from the MIT-BIH formatted data file, but first the DAC12 control register must be properly configured. The next two sections discuss the configuration of the DAC12 control register and the implementation of the DAC12 data register respectively.

The DAC12 Control Register (DAC12_xCTL)

A layout of the control register is shown in Table 4 and the purpose of the control bits will be explained in turn (see Appendix for permission to use images). The control register (DAC12_xCTL) contains 16 bits (0-15), however, only 15 require settings (bit 15 is not used). Bits 14 and 13 are referred to as the DAC12SREFx (see Table 4) and are used to select the reference voltage for the DAC12. The DAC12 can be configured to use either an external

reference voltage (V_{REF+}) or the internal 1.5-V or 2.5-V reference (V_{REF+}) from the ADC12 module with the DAC12SREF_x bits. When DAC12SREF_x = {0,1} the V_{REF+} signal is used as the reference and when DAC12SREF_x = {2,3} the external V_{REF+} signal is used as the reference [62]. To use the ADC12 internal reference, it must be enabled and configured via the applicable ADC12 control bits.

Table 4 DAC12 Control Register

| | | | | | | | |
|---|------------------------|-----------------------------------|----------|------------------------|----------|----------------|--------------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | DAC12SREF _x | | DAC12RES | DAC12LSEL _x | | DAC12 CALON | DAC12IR |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DAC12AMP _x | | | DAC12DF | DAC12IE | DAC12IFG | DAC12ENC | DAC12 GRP |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |
|  | | Modifiable only when DAC12ENC = 0 | | | | | |

Once the ADC12 reference is configured, the reference voltage appears on the V_{REF+} signal. For this application the internal reference (V_{REF+}) was used, and both the 1.5 volt internal reference and the 2.5 volt internal reference are used. The user chooses an amplitude of either low (1.5 V.) or high (2.5 V.) during the setup for the simulator. Then based on the user's preference the DAC12 applies either the 1.5 volt reference or the 2.5 volt reference. This

provides for two amplitude options. The DAC12 output signal (the ECG wave) will then be in the range of zero to 1.5 volts or zero to 2.5 volts. The DAC12 output voltage has a maximum limit equal to the voltage the microcontroller is being run on (between 1.8 and 3.6 volts). The unit used in this application runs on a voltage of 3.3 volts. This becomes the maximum voltage the DAC12 can output.

Bit 12 of the control register is called DAC12RES (see Table 4) and is used to set the desired resolution. A zero at bit 12 of the control register sets the DAC12 to implement 12-bit resolution and a one at bit 12 sets the DAC12 to implement 8-bit resolution [62]. The simulator prototype utilized 12-bits of resolution.

Bits 11 and 10 (see Table 4) of the control register are referred to as DAC12LSELx bits and are used to select the load trigger for the DAC12 latch. The DAC12ENC must be set for the DAC12 to update, except when DAC12LSELx is zero (DAC12ENC is discussed below). The configuration for the load select is shown in Table 5 below. The purpose of this option is so the DAC12 data register can be connected directly to the DAC12 core or double buffered. The trigger for updating the DAC12 voltage output is selected with the DAC12LSELx bits. When $\text{DAC12LSEL}_x = 0$ the data latch is transparent and the DAC12 data register is applied directly to the DAC12 core. The DAC12 output updates immediately when new DAC12 data is written to the DAC12 data register, regardless of the state of the DAC12ENC bit. When $\text{DAC12LSEL}_x = 1$, DAC12 data is latched and applied to the DAC12 core after new data is written to DAC12_xDAT. When $\text{DAC12LSEL}_x = 2$ or 3, data is latched on the rising edge from the Timer_A or Timer_B respectively. DAC12ENC must be set to latch the new data when

DAC12LSELx is not zero [62]. The ECG simulator prototype will latch immediately when the data is written (setting: 0 0). The simulator prototype did not implement the timer, but used a delay instead to provide for the signal frequency based on the sample rate. The delay routine turned out to be very accurate and easy to change and thus it was used to vary the ECG frequency.

Table 5 DAC12LSELx Configuration

| Bits 11-10 | Load Select Control Description |
|---------------|---|
| 0 0 | DAC12 latch loads when data is written (DAC12ENC is ignored) |
| 0 1 | DAC12 latch loads when data is written or, if grouped, when all DAC12 data registers have been written. |
| 1 0 | DAC12 latch loads at the rising edge of Timer_A.OUT1 (TA1) |
| 1 1 | DAC12 latch loads at the rising edge of Timer_B.OUT2 (TB2) |

Bit 9 of the control register (see Table 4) is referred to as DAC12CALON and is used to turn the calibration option for the DAC12. The DAC12 has the capability to calibrate the offset voltage⁶ of the output amplifier. This bit initiates the DAC12 offset calibration sequence and is automatically resets when the calibration completes. A zero (at bit 9) makes the calibration option inactivate and a one initiates calibration. The calibration should complete before using the DAC12. The DAC12AMPx bits should be configured before calibration and for best calibration results, port and CPU activity should be minimized during calibration [62]. The

⁶ Offset Voltage (Error): The output voltage that exists when the input digital code is set to give an ideal output of zero volts. All the digital codes in the transfer curve are offset by the same value. Offset error is usually expressed in LSBs.

Authentic ECG Simulator prototype did not utilize the calibration option because the off-set voltage is not a concern in this type of application.

Bit 8 of the control register (see Table 4) is referred to as DAC12IR and is used to select the output range. This bit sets the reference input and the voltage output range. The full-scale output is programmable to be 1X or 3X of the selected reference voltage via the DAC12IR bit. This allows the user to control the range of the DAC12. The formulas for the output voltages are shown in Table 6 (see Appendix for permission to use images) [62]. A zero at the DAC12IR bit adds a multiplier of three to the formula, and a one at DAC12IR does not. During the development of the simulator prototype the 3X option was tested, however, the 3X option caused the DAC12 output to exceed its maximum voltage output and thus clipped off the top peaks of the ECG wave (on some data files). Because of this the 3X option was not used but selecting the voltage reference was used instead, this allows for two different voltage amplitudes.

Table 6 DAC12 Output Range

| Resolution | DAC12RES | DAC12IR | Output Voltage Formula |
|------------|----------|---------|--|
| 12 bit | 0 | 0 | $V_{out} = V_{ref} \times 3 \times \frac{DAC12_xDAT}{4096}$ |
| 12 bit | 0 | 1 | $V_{out} = V_{ref} \times \frac{DAC12_xDAT}{4096}$ |
| 8 bit | 1 | 0 | $V_{out} = V_{ref} \times 3 \times \frac{DAC12_xDAT}{256}$ |
| 8 bit | 1 | 1 | $V_{out} = V_{ref} \times \frac{DAC12_xDAT}{256}$ |

In 8-bit mode the maximum useable value for DAC12_xDAT is 0FFh and in 12-bit mode the maximum useable value for DAC12_xDAT is 0FFFh. Values greater than these may be written to the register, but all leading bits are ignored.

Bits 7 – 5 of the control register (see Table 4) are referred to as the DAC12AMPx bits and are used to select the settling time of the DAC12 versus the current consumption for input and output amplifiers. One of eight possible selections is chosen using the DAC12AMPx bits. The settling time options are shown in Table 7 [62]. In the low/low setting, the settling time is the slowest, and the current consumption of both buffers is the lowest. The medium and high settings have faster settling times, but the current consumption increases. The Authentic ECG Simulator prototype can be set for medium or even low because the frequency of the data transfer is relatively slow.

Table 7 Settling Time vs. Power Consumption Options

| DAC12AMPx | Input Buffer | Output Buffer |
|-----------|------------------------|--------------------------|
| 000 | Off | DAC12 off, output high Z |
| 001 | Off | DAC12 off, output 0 V. |
| 010 | Low speed & current | Low speed & current |
| 011 | Low speed & current | Medium speed & current |
| 100 | Low speed & current | High speed & current |
| 101 | Medium speed & current | Medium speed & current |
| 110 | Medium speed & current | High speed & current |
| 111 | High speed & current | High speed & current |

Bit 4 of the control register is referred to as the DAC12DF bit (see Table 4) and is used to specify the data format. A zero at bit 4 sets the format to straight binary and a one at bit 4 sets the format to two’s compliment. The formulas given in Table 6 are for using straight binary. When using the two’s compliment data format, the range is shifted such that a data value of 0800h (0080h in 8-bit mode) results in a zero output voltage, 0000h is the mid-scale output voltage, and 07FFh (007Fh for 8-bit mode) is the full-scale voltage output. When using the

straight binary data format, the full-scale output value is 0FFFh in 12-bit mode (0FFh in 8-bit mode) [62]. The Authentic ECG Simulator prototype uses the straight binary data format.

Bit 3 of the control register is referred to as the DAC12IE bit (see Table 4) and is used to enable or disable interrupts to the DAC12. Bit 2 of the control register is the actual interrupt flag bit, referred to as DAC12IFG (Interrupt Flag). If the DAC12IE bit is set to zero (disabled), the DAC12IFG flag will not trigger an interrupt transfer. If both the DAC12IE and GIE bits are set to one, the DAC12IFG generates an interrupt request. The DAC12IFG flag is not reset automatically. It must be reset by software [62]. The simulator prototype does not require interrupts so the DAC12IE bit was set to zero (disabled) and the DAC12IFG bit does not matter.

Bit 1 of the control register is referred to as the DAC12ENC bit (see Table 4) and is used to enable digital to analog conversion. This bit enables the DAC12 module when DAC12LSELx is greater than zero (0 on DAC12ENC = DAC12 disabled; 1 on DAC12ENC = DAC12 enabled). When DAC12LSELx is zero, DAC12ENC is ignored. Thus, this bit is used along with the DAC12LSELx to control the load trigger for the DAC12 latch. The reader is referred to the section on the DAC12LSELx above (bits 11 and 10) for more information [62]. For the simulator prototype the timer was not used. Instead a delay was implemented that could be changed to produce different frequencies based on the sample rate. This proved to work very accurately.

Bit 0 of the control register is referred to as the DAC12GRP bit (see Table 4) and is used when several DAC12s are used together. Multiple DAC12s can be grouped together with the

DAC12GRP bit to synchronize the update of each DAC12 output. Hardware ensures that all DAC12 modules in a group update simultaneously independent of any interrupt or NMI event [62]. The simulator prototype only uses one DAC12 so this bit is set to zero (one for grouping DAC12s).

The DAC12 Data Register (DAC12_xDAT)

The data register holds the value that is applied to the formula in Table 5 (DAC12_xDAT). In this application this is the value input from the MIT-BIH formatted data file. Since the DAC12 is only a 12-bit converter only the lower 12 bits of the 16-bit register are used. In the 8-bit straight binary mode the maximum useable value for DAC12_xDAT is 0FFh (255 in decimal) and in 12-bit straight binary mode the maximum useable value for DAC12_xDAT is 0FFFh (4095 in decimal). Values placed in the register greater than that which can be held in 12 bits may be written to the data register, but the leading bits (above 12) will be ignored. The DAC12 data are right-justified so bit 11 is the most significant bit [62]. The output voltage produced by the DAC12 will follow the plot, shown in Figure 23, for a straight binary mode input. The input data follows the x-axis and the output voltage follows the y-axis.

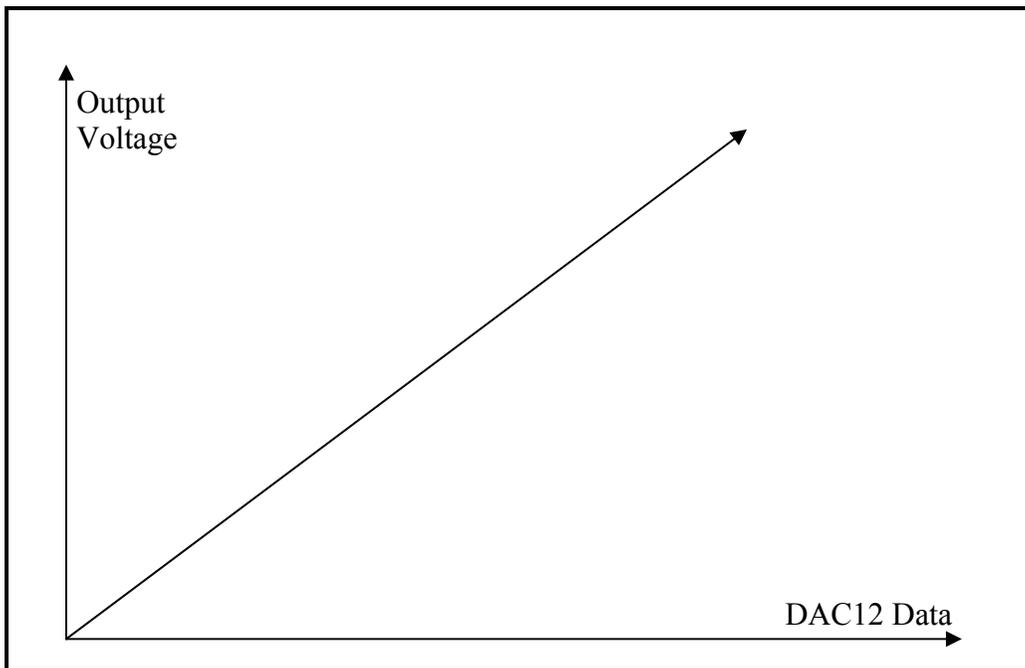


Figure 21 DAC12 Output (binary mode)

Programming the Options for the ECG Signal

The software that actually produces the ECG wave requires the user to make several selections. The user is first prompted to choose a data file to run on the simulator. In order to provide the choices a user menu is provided. After some initial instructions are displayed on the LCD the user can scroll through all the data files, displaying each one on the LCD, by pushing one of the buttons. When the user finds the file name he or she wants to execute they push a second button to choose that file and thus to input the necessary information to the program (data file location on the NAND memory and data file size). Next the user is provided a prompt which gives them the option of choosing the amplitude of the ECG signal. Two amplitude options are provided

low, which uses the 1.5 voltage reference and high which uses the 2.5 voltage reference. After the choice is made by the user the display will show the choice that was made and then display another prompt. This prompt will ask the user to choose a frequency. Two frequency options are shown – normal and high. (More frequency options could also be easily added.) After the user makes the frequency choice the display shows the choice that was made and then prompts the user to push a button to run the ECG signal. The DAC12 producing the ECG signal is designed to run continuously until the user pushes the stop button. The functions that were written to implement these options are listed below and will be briefly discussed in turn.

```
/****** Functions Related to DAC12 *****/  
unsigned char setFile(void);           // File choice input from User (to run on DAC)  
unsigned char setAmplitude(void);     // Amplitude choice input from user  
void initialize_DAC (unsigned char);  // Initialize DAC12 control register  
unsigned char setFrequency(void);     // sets the users desired frequency
```

The first function necessary to facilitate the file options is the *setFile* function. This function provides the menu of file choices (actually displaying each file name on the LCD) and when the user makes the selection from the menu the function initializes all of the associated values for the file chosen (the file size, the file start address, and the delay necessary for the sample rate for the file). The next function, *setAmplitude*, prompts the user to push a certain button in order to select an amplitude. The function displays the selection on the LCD and then sends the choice to the function *initialize_DAC* (listed next). The *initialize_DAC* function then initializes the control registers for the DAC12 and the ADC (analog to digital converter). Based on the choice made the control register is configured for either a 1.5V. amplitude or a 2.5V. amplitude. The next function, *setFrequency*, prompts the user for the frequency at which to run the ECG signal and

then applies the choice to a formula to create the proper frequency for the DAC12 – based on a delay routine.

CHAPTER SIX: RESULTS AND CONCLUSIONS

The Authentic ECG Simulator's Waveform

The most important result of this thesis is what the ECG waveform produced by the prototype simulator looks like. The goal of this thesis, as was stated in Chapter One, is that the prototype should demonstrate that a portable ECG simulator can be designed to produce an ECG waveform signal exactly like an actual human patient's ECG waveform signal. After the Authentic ECG Simulator prototype was built it was tested on a digital oscilloscope to determine what the simulator's output looked like. The digital oscilloscope allows the dynamic waveform to be captured and a snapshot taken of the waveform segment from the oscilloscope screen. Several of these snapshots were taken so they could be compared with the original waveforms obtained from MIT-BIH. The waveforms from the MIT-BIH database are known to be authentic human ECG signal. The question is: how well does the prototype's waveform match up with the actual human waveform that the data originated from. In order to make these comparisons plots were obtained from the Physionet web site [38] for all the ECG data files placed on the simulator. A critical issue in comparing the plots is making sure that the same time frame is being examined in the two waveforms. Both the Physionet plots and the oscilloscope snapshots are only able to capture a small portion of the one minute waveform. Several plots taken from Physionet and oscilloscope snapshots are now provided for comparison. The figures of the plots are placed on the same pages following this section so the reader can compare the waveforms more easily.

Figure 24 shows a zoomed in plot of the data file “apnea_ecg_a15” from the Physionet web site [38]. The “apnea_ecg_a15” file is also on the prototype simulator. This time segment was chosen because it is the only area on the one minute waveform where the ECG signal dips down substantially. The dip on the waveform provides a landmark, so to speak, so the wave can be compared to the oscilloscope output from the prototype simulator. The oscilloscope snapshot of the “apnea_ecg_a15” waveform segment produced by the simulator is shown in Figure 25. The waveform patterns are virtually identical, the difference in appearance is due to the time per division settings on the two plots being different and the fact that the oscilloscope snapshots appear to have higher resolution than the plots from the Physionet web site.

Figure 26 and Figure 27 are wave segments from the MIT-BIH data file “cudb_cu01 [38].” Figure 26 is the plot from Physionet and Figure 27 is from the simulator oscilloscope snapshot. Again, this segment was chosen because it is the only area on the one minute waveform where the ECG signal dips down substantially. The two wave segments, again, appear to be identical apart from the difference due to the time per division settings. A last comparison can be seen in Figures 28 and 29. These two plots are from the same wave segment on the data file “stdb_313.” This time the sharp downward spike in the middle of the plots provides a landmark. And again the waveform segments are identical in form (see Appendix for permission to use images).

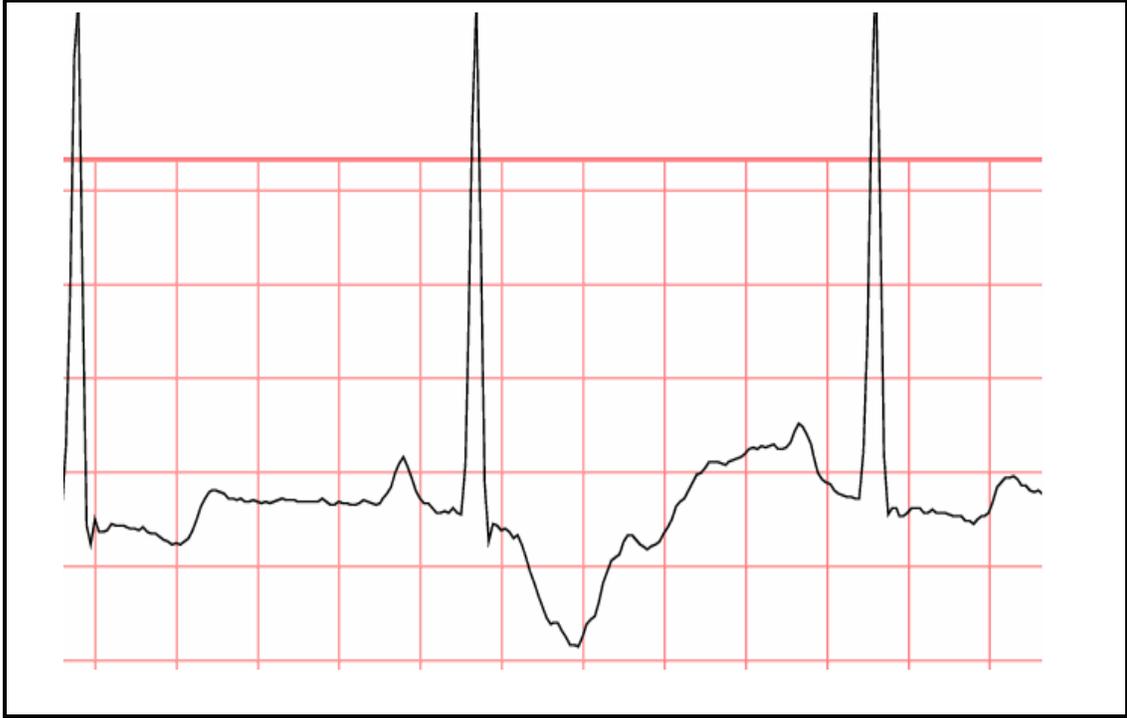


Figure 22 Physionet File apnea_ecg_a15 Waveform Segment

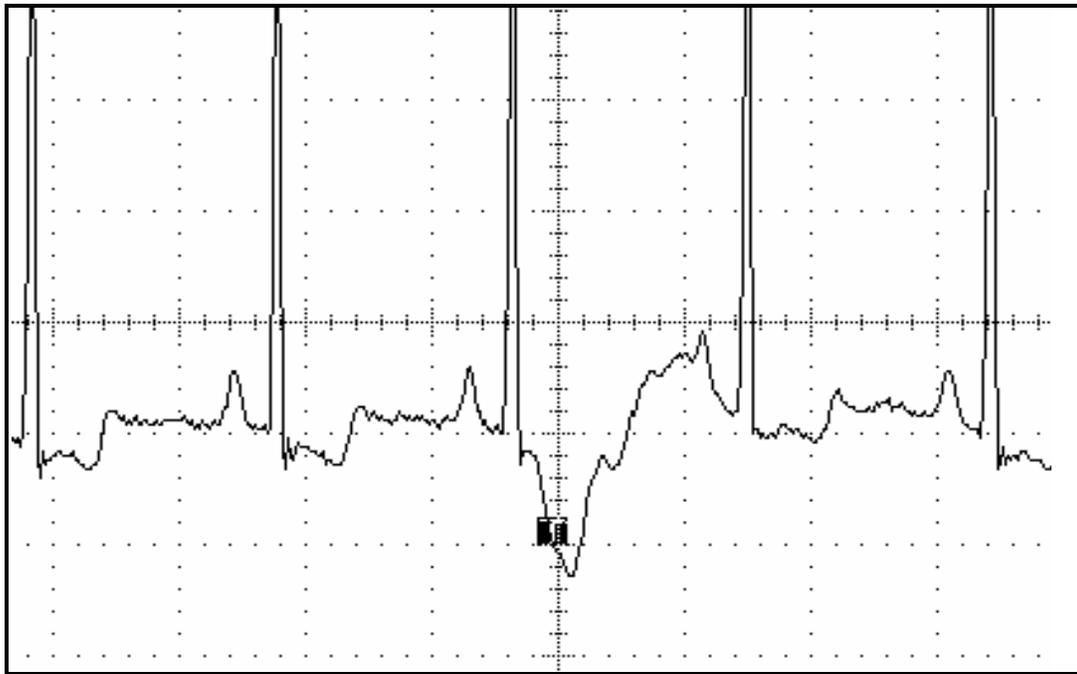


Figure 23 File apnea_ecg_a15 Snapshot take on Oscilloscope

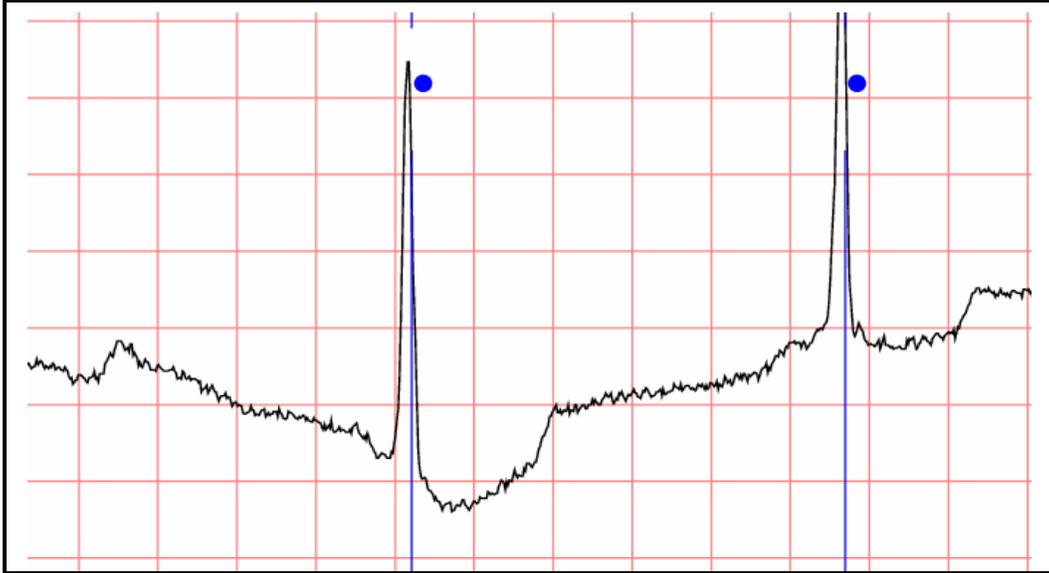


Figure 24 Physionet File cudb_cu01 Waveform Segment

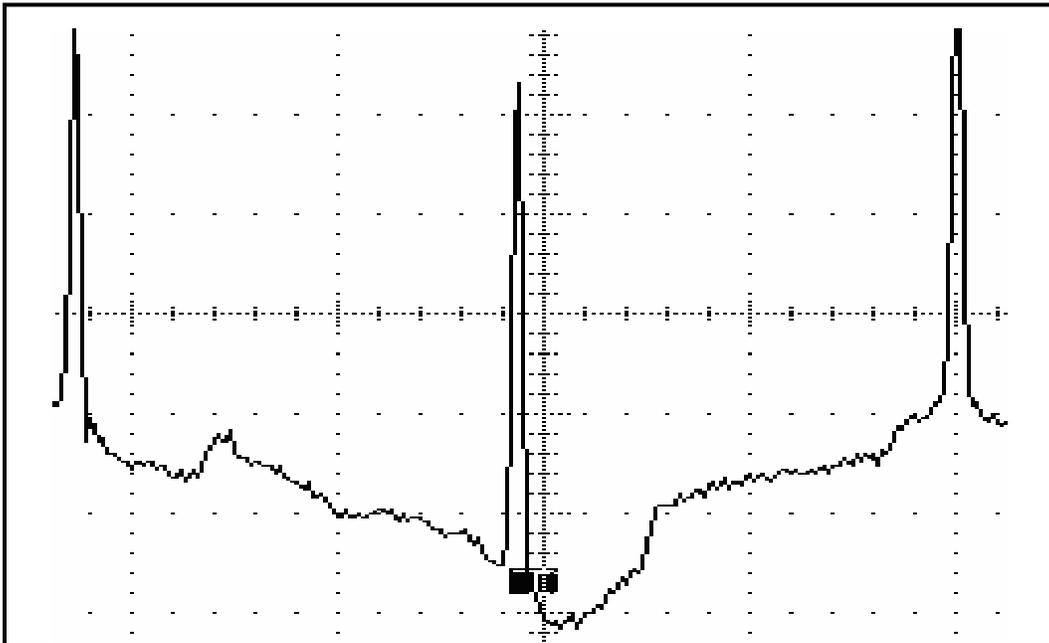


Figure 25 File cudb_cu01 Snapshot taken on Oscilloscope

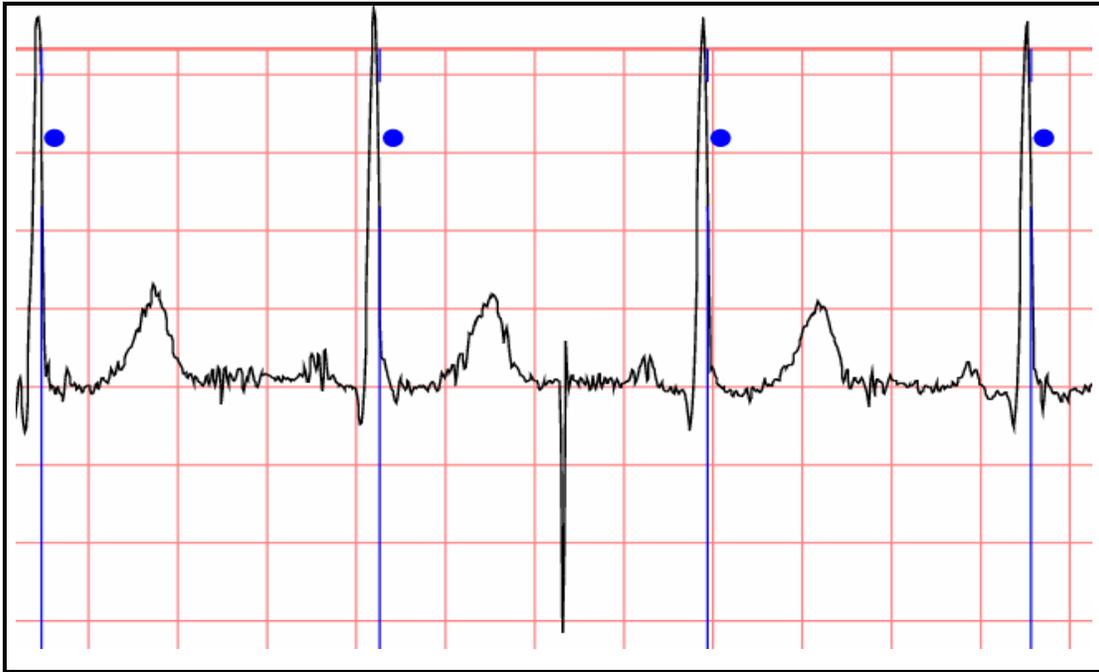


Figure 26 Physionet File stdb_313 Waveform Segment

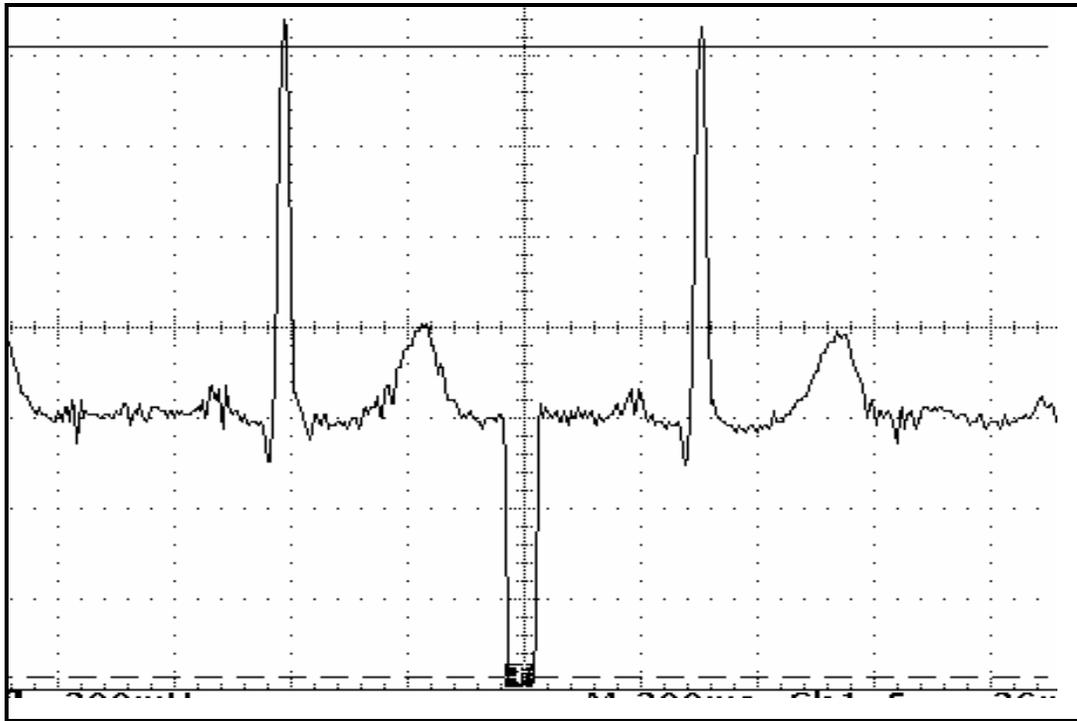


Figure 27 File stdb_313 Snapshot taken on Oscilloscope

All of the data files stored on the prototype simulator (7 data files were stored on the simulator) were test on the oscilloscope and compared with the file's plots available on the Physionet web. The visual comparison showed that all of the simulator's waveforms match the Physionet plots perfectly. This is not surprising –the plots should match perfectly because they are using the same data. But the comparison proves that the prototype simulator works exactly as expected and it produces authentic human ECG signals.

Strengths of the Authentic ECG Simulator

The first and most important strength of the Authentic ECG Simulator is that it does not create its ECG waveforms mathematically as all other portable simulators currently do. This is important because using mathematical formulas will never produce all of the subtle variations and inconsistencies that can be present in actual human ECG waveforms. In addition, the waveform created by the Authentic ECG Simulator is not static over time. For a full one minute the waveform is dynamically changing. This means the waveform doesn't merely repeat itself in an identical fashion as the current portable simulators do. Also, the low frequency baseline shift; that is due to the patient breathing, is present in the Authentic ECG Simulator's waveform. This is not present in a simulator's ECG waveform that is created mathematically. As already discussed, one of the most important diagnostic capabilities of modern ECG monitors is the ability to detect arrhythmias. However, in order for a manufacturer to claim their monitor has arrhythmia detection capabilities the monitor must be thoroughly tested on actual human ECG signals with the particular arrhythmias. This type of testing can not be done on the currently available portable simulators because they are not producing an authentic human ECG. A second important strength of the Authentic ECG Simulator lies in the fact that it can store a

number of authentic arrhythmia ECG files. Thus it can be used to test a monitor's arrhythmia detection capabilities. Over one hundred ECG signal data files with known arrhythmias are available from the MIT-BIH database files. Depending on the size of the data files and the amount of memory loss due to fragmentation the prototype simulator will hold some where between one hundred and two hundred one minute data files. This is a lot of ECG waves. The currently available portable simulators at most can synthetically produce perhaps up to twelve different arrhythmias. A third strength of the Authentic ECG Simulator lies in the fact that the data files on the unit can be changed. The prototype simulator can hold many files, as previously discussed. However, even when the simulator's memory becomes full data files can be changed (by erasing files and replacing them with different files). This provides an aspect of versatility to the Authentic ECG Simulator not available on the current simulators. A fourth strength of the Authentic ECG Simulator is the fact that it is portable and can be run on a 9 volt battery (rechargeable). Some additional positive features of the Authentic ECG Simulator are: (1), the fact that multiple frequencies can be implemented on the simulator by simply changing the time interval between the digital to analog conversions, and (2), providing for two voltage amplitudes – more amplitude settings could be added as long as the ECG signal does not exceed the DAC12 maximum voltage.

Areas That Require Further Work

There are a few aspects of the prototype simulator that could be improved. Probably the most important improvement that should be pursued is faster and more reliable development software for reading the data from the formatted MIT-BIH data files so it can be stored onto the NAND flash memory. As was mentioned, in the software discussion, the storing of a one minute data

file on to the flash memory can currently take over two hours and can fail if the stack is overrun. The problem is not in the flash memory's programming speed. According to the Samsung data sheet writing one page of data (528 bytes) takes typically only 200 μ s [50]. This means a data file of 82 memory pages (typical size) should take less than a second to be stored onto the NAND flash memory. The problem instead lies in the development software's ability to read any input into the microcontroller. As always getting input is the slowest aspect of program execution. The only way to solve this problem is to find different development software platform for the MSP430 that has faster input capabilities. This could be a viable possibility, as several other vendors do provide software development platforms for the MSP430.

Another aspect of the software that could possibly be improved is to integrate all of the current programs into just one program with different modes of operation. This would require a better user interface such as a digital keypad for input and a larger LCD unit for output. However, both of these would be needed for the full featured simulator anyway. Currently there is one program to erase the NAND flash memory and store data onto the NAND, a second program to verify the data got stored properly by reading and printing it out, and finally a third program that uses the data to create the ECG signal. The second program that verifies the data by reading it should not be necessary after the testing phase of the software development is completed. It is necessary during the development phases, but a final version of the simulator, once debugged, will no longer require this software. That means that only the other two programs would need to be integrated into one. This software application would need to have two modes of operation, one for storing or erasing data and one for running the ECG signal. The only thing that might prohibit this improvement is the capacity of the ROM memory on the MSP430. The issue is whether or

not it would be able to hold all of the programming instructions. The only way to know for sure is to try it. The newly integrated single program would also be larger than the two separate programs currently are because the code for changing modes would need to be added. However, if this improvement could be added it would make the simulator more user friendly. It should be noted that even with this improvement loading data files onto the simulator will still requires that the simulator be connected to a personal computer, with the applicable software necessary, and an internet connection, so the Physionet web site can be accessed.

APPENDIX:
PERMISSION TO USE IMAGES

Permission to use image in Figure 1, Plot (b)

Dear Paul

Thank you for your enquiry.

Yes, no problem, you can use our waveforms for your paper. It would be nice, if possible, to cite Serviciencia as the source.

I would be interested in to read your paper in future, if posible, also.

Best regards

Alberto R. Marino
General Manager
E-mail: marino@serviciencia.es

Serviciencia, S. L.
Isabel II, 22
Polígono Industrial "Sector 23" - Nave 70
45210 YUNCOS
(Toledo)
SPAIN
Tel: (+34) 925 536154 - Fax: (+34) 925 537644
E-mail: serviciencia@serviciencia.es - Internet: www.serviciencia.es

----- Original Message -----

From: Paul Michalek

To: serviciencia@serviciencia.es

Sent: Sunday, May 07, 2006 11:07 PM

Subject: Permission to use graphic

I would like to use the graphic of the ECG waveform on your website (<http://www.serviciencia.es/fc3-in.htm>) in a research paper I am writing. I have to get your permission in order to use it in my paper.

Thank You,
Paul Michalek

Permission to use image in Figure 3

http://en.wikipedia.org/wiki/Cardiac_action_potential#Phase_4.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts (Subject to disclaimers).

Permission to use image in Figure 4, 6, 7

Paul,

You have my permission to use the figures indicated in your e-mail (below) in your thesis. Best wishes as you complete your thesis.

Richard Klabunde, Ph.D.
www.cvphysiology.com

Paul Michalek wrote:
Richard E. Klabunde, Ph.D.

I am writing a thesis on an ECG Simulator. I would like to use some of the images of the ECG Segment breakdown on your webpage.
<http://www.cvphysiology.com/Arrhythmias/A009.htm>.

If you can give me your permission I would greatly appreciate it.

Thanks
Paul Michalek

Permission to use image in Figures 5 and 8

From: http://en.wikipedia.org/wiki/Image:ECG_001.jpg

I, the creator of this work, hereby release it into the public domain. This applies worldwide. In case this is not legally possible, I grant any entity the right to use this work for any purpose, without any conditions, unless such conditions are required by law.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

Subject to disclaimers.

Permission to use images in Figures 9, 24, 26, and 28

> studentid207@aol.com
> 06-14-06
> Paul Michalek

> Dear Sir or Madam:
>
> This letter is to ask for your permission to use images from your website. I am completing a master's degree at the University of Central Florida a degree of Master of Science in the Department of Computer Engineering. I would like your permission to use the images on your website (address below) in my thesis.
> <http://www.physionet.org/cgi-bin/chart>
>
> Please send an email to studentid207@aol.com . Thank you for your attention in this matter.
> Sincerely,
>
> PERMISSION GRANTED FOR THE USE REQUESTED ABOVE:
> By: _____
> Date: _____
> _____

Dear Paul,

You may use any materials from PhysioNet with proper credit. Cite the URL(s) for the materials you use, and also cite the Circulation article about PhysioNet (see <http://www.physionet.org/citations.shtml> for instructions).

When your thesis is complete, please send me a note with a citation for it (title, your full name, school and department, city/state, date).

Best wishes for success in your project!

George

George B. Moody

Harvard-MIT Division of Health Sciences and Technology

MIT Room E25-505A

Cambridge, MA 02139 USA

Permission to use images in Figures 13, 16, 19 and 20

Hi Paul,

Thank you for asking for permission. You have my permission to use the four images listed below for your Thesis. Please include a tag that states "Image used with permission from www.sparkfun.com".

Good luck with the thesis review and let me know if you need anything else,

-Nathan

Paul Michalek wrote:

> Nathan:

>

> As per our phone conversation. I need to get written permission to
> use images from your website in my Thesis paper for the University of
> Cental Florida. I will of course reference your website as the
source

> of the images and give your compay full credit for any text or image
I

> use. If you can email a reply granting permission to use the images
> at the URLs listed I would greatly appreciatiate it.

>

> 1. The board I purchased:

> http://www.sparkfun.com/commerce/product_info.php?products_id=58

>

> 2. The schematic of the board:

> <http://www.olimex.com/dev/images/msp430-169stk-sch.gif>

>

> 3. The JAG interface:

> http://www.sparkfun.com/commerce/product_info.php?products_id=15

>

> 4. The LCD Display on the board:

> http://www.sparkfun.com/commerce/product_info.php?products_id=255

>

> Thank You Very Much,

> Paul Michalek

>

Permission to use images in Figures 15 and Tables 4 and 6

Hello Paul,

This permission is given in the disclaimer in all the TI datasheets and user's guides:
Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices.
Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.
I hope this helps. If you have any other questions, please let me know.

Regards,

Jim Fell

MSP430 & Catalog TMS470 Applications
Texas Instruments
Semiconductor Technical Support
http://www-k.ext.ti.com/sc/technical_support/pic/americas.htm

-----Original Message-----

From: pjmich3131@yahoo.com

Sent: 6/21/2006 02:53:22 PM

To: support@ti.com

Subject: Jim Fell - Permission to use image / regarding our phone conversation

Jim Fell,

I talked to you a day or 2 ago and you said I could use any images from the Texas Instruments data sheet in my thesis paper. Can you please respond to this email. I have to have something in writing. I will of course reference Texas Instruments properly as the source and will not modify any tables or figures (as per our conversation).

Thank You Very Much,
Paul Michalek

Thank you for your interest in Texas Instruments. We grant the permission you request in your e-mail below.

On each copy, please provide the following credit:

Courtesy Texas Instruments

Regards,

Larry Bassuk
Copyright Counsel
Texas Instruments Incorporated

Office 972-917-5458
Fax 972-917-4418
Cell 214-458-5975

From: Paul Michalek [mailto:pjmich3131@yahoo.com]
Sent: Friday, June 16, 2006 4:08 PM
To: copyrightcounsel@list.ti.com - Copyright Legal Counsel
Subject: Permission to use image

Dear Sir,

I am using the TI MSP430 microcontroller in my Thesis for my Master's degree at University of Central Florida. I would like to ask your permission to use several images from the TI User's Manual for the MSP430. If it would be okay please email me. I would of course reference Texas Instruments where ever the images are used.

thanks
Paul Michalek

Note: All images not listed in this appendix were created by the author of this thesis, Paul J. Michalek.

REFERENCES

- [1] Leading Causes of Death — Statistics, Cardiovascular Diseases, Source on the Internet: American Heart Association, Available at:
<http://www.americanheart.org/downloadable/heart/1136822094308LeadingCauses06.pdf>.
- [2] Lawrence S. Cohen, M.D., Yale University School of Medicine Heart Book, Heart Disease Symptoms, Chapter 9,., Available at:
<http://info.med.yale.edu/library/heartbk/9.pdf>.
- [3] A-Z Health Guide from WebMD: Medical Tests, Electrocardiogram, June 09, 2006, Source on the Internet: WebMD, available at:
http://www.webmd.com/hw/heart_disease/hw213248.asp.
- [4] *Heart and Circulatory, Tachycardia*, Harvard Medical School for IntelliHealth, Source on the Internet: Aetna IntelliHealth, Available at:
<http://www.intelihealth.com/IH/ihtIH/WSIHW000/8059/23704/266770.html?d=dmthHealthAZ>.
- [5] Dale Dubin, M.D., Understanding the Heart's Electrical Messages, Source on the internet: Essential EKG (ECG) Information For Physicians, January, 2006, Available at:
http://themdsite.com/life_preserver.cfm.
- [6] Frank G. Yanowitz, M.D., The Standard 12 Lead ECG, Source on the Internet: Alan E. Lindsay, ECG Learning Center, Available at:
http://medlib.med.utah.edu/kw/ecg/ecg_outline/Lesson1/index.html.
- [7] Dale Dubin, M.D., Understanding the Heart's Electrical Messages, Source on the internet: Essential EKG (ECG) Information For Physicians, Physicians,., January, 2006, Available at: <http://www.themdsite.com/physicians.cfm>
- [8] Webster, John G., Medical Instrumentation Application and Design, Third ed. , John Wiley & Sons, 1998.
- [9] US Food and Drug Administration, " Medical Devices," Available at:
<http://www.fda.gov/oc/ohrt/irbs/devices.html#risk>.
- [10] Best Research Practices, Device Class and Regulatory Controls, Source on the Internet: Medical College of Georgia, 2005, Available at:
http://www.mcg.edu/research/ohrp/BestPractices/Device_Class.htm.
- [11] Cardiac Monitor Guidance (including Cardiotachometer and Rate Alarm), Guidance for Industry; November 5, 1998, U.S. Department of Health and Human Services Food and

- Drug Administration Center for Devices and Radiological Health, Available at:
<http://www.fda.gov/cdrh/ode/cmonitor.pdf>.
- [12] Wayne R. Crosby, Source on the Internet: Equipment Requirements for Validating, A Patient Monitoring System, 1995 Nelson Publishing, Available at:
<http://www.evaluationengineering.com/archive/articles/0595medi.htm>.
- [13] Ivaylo I Christov, Real time electrocardiogram QRS detection using combined adaptive threshold, Source on the Internet: BioMedical Engineering OnLine 2004, Available at:
<http://www.biomedical-engineering-online.com/content/3/1/28>.
- [14] D. Tsunami, J. McNames, M. Aboy, M.S. Ellenby, A New Resource for Independent and Blinded Assessment of QRS Detection Algorithms, Biomedical Signal Processing Laboratory, Electrical and Computer Engineering, Portland State University, Portland OR, USA, Complex Systems Laboratory, Pediatrics, Oregon Health & Science University, Portland OR, USA, Available on the Internet at:
<http://bsp.pdx.edu/Publications/2003/TsunamiEMBS.pdf>.
- [15] MIT-BIH Database, Source on the Internet: MIT-BIH Database and Software Catalog, Massachusetts Institute of Technology, 2005, Available at: <http://ecg.mit.edu/dbinfo.html>.
- [16] European ST-T Database, Source on the Internet: Registries & Databases, European ST-T Database, Available at:
http://www.escardio.org/knowledge/ehs/registries/european_st.htm.
- [17] SERVICIENCIA, S.L. SPAIN, Source on the Internet: ECG Simulator, Ferronato - FC3D, Available at: <http://www.serviciencia.es/fc3-in.htm>.
- [18] Dale Technology, ECG/Arrhythmia Simulator - Dale EHS10, 5200 Convair Drive, Carson City, NV 89706, Source on the Internet: daletch.com, Available at:
http://www.daletch.com/main/product_info.php-cPath=24&products_id=42.htm.
- [19] Dale Technology, ECG/Arrhythmia Simulator - DALE13, 5200 Convair Drive, Carson City, NV 89706, Source on the Internet: daletch.com, Available at:
http://www.daletch.com/main/product_info.php-cPath=24&products_id=43.htm.
- [20] NETECH MiniSim 1000, Netech Corporation, 60 Bethpage Drive, Hicksville, NY, Source on the Internet: netech.org, Available at:
<http://www.netech.org/productdetail.asp?id=51>.
- [21] Bio-Tek, Lionheart 3 Multi-Parameter Simulator, Fluke Corporation, Source on the Internet: AKW Medical, Inc., Available at:
<http://www.healthcaremall.com/akwmed/ecgsim.htm>.

- [22] MULTI-PRO 2000, Electrical Safety Analyzer / 12 Lead ECG Simulator, Netech Corporation, 60 Bethpage Drive, Hicksville, NY, Source on the Internet: netech.org, Available at: <http://www.netech.org/productdetail.asp?id=57>.
- [23] DELTA 3000 - A three in one unit with Defibrillator Analyzer / Pacemaker Analyzer and ECG Simulator, Netech Corporation, 60 Bethpage Drive, Hicksville, NY 11801, Source on the Internet: netech.com, Available at: <http://www.gonetech.com/delta3000.htm>.
- [24] M311 ECG Simulator, Fogg System Company, Inc. 15592 East Batavia Drive; Aurora, CO 80011, Source on the Internet: allwave.com, Available at: <http://www.allwave.com/pc-309-6-m311-ecg-simulator.aspx>.
- [25] Cardiovascular System I: The Beating Heart, Source on the Internet: BioCoach Activity, Available at: http://www.phschool.com/science/biology_place/biocoach/cardio1/intro.html.
- [26] Heart Electrical Activity, Source on the Internet: HeartSite.com, Available at: http://www.heartsite.com/html/electrical_activity.html.
- [27] J.W. Clark, "The Origin of Biopotentials", Medical Instrumentation: Application and Design, 3rd edition, edited by J.G. Webster, John Wiley and Sons, Inc., New York, NY Chap. 4, pp. 123-124, 1998.
- [28] Cardiac action potential, Source on the Internet: Wikipedia, Available at: http://en.wikipedia.org/wiki/Cardiac_action_potential#Phase_4.
- [29] Richard E. Klabunde, Electrocardiogram, Source on the Internet: Cardiovascular Physiology Concepts, Available at: <http://www.cvphysiology.com/Arrhythmias/A009.htm>.
- [30] The QRS Complex, The EKG Waveform, Source on the Internet: Electrocardiology and Cardiac Arrhythmias, Available at: <http://sprojects.mmi.mcgill.ca/cardiophysio/EKGQRScomplex.htm>.
- [31] EKG or Electrocardiogram, Source on the Internet: HeartSite.com, Available at: <http://www.heartsite.com/html/ekg.html>.
- [32] The Cardiac Prevention and Rehabilitation Centre; Source on the Internet: St. Michael's Hospital, Toronto, Ontario, Available at: http://www.stmichaelshospital.com/images/cardiac_images/figure2.jpg.
- [33] Stress Test Intro, Source on the Internet: HeartSite.com, Available at: http://www.heartsite.com/html/stress_test.html.

- [34] Cardiac Arrhythmias, Introduction to Cardiac Arrhythmias, Source on the Internet: Electrocardiology and Cardiac Arrhythmias, Available at: <http://sprojects.mmi.mcgill.ca/cardiophysio/arrythmiasintro.htm>.
- [35] Richard E. Klabunde, Arrhythmias, Source on the Internet: Cardiovascular Physiology Concepts, Available at: <http://www.cvphysiology.com/Arrhythmias/A008.htm>.
- [36] Margaret Wahl, THE HEART IS A MUSCLE, TOO, Source on the Internet: Quest, vol. 6, No.7, June 1999, Available at: <http://www.mdausa.org/publications/Quest/q63cardiac.html>.
- [37] Richard E. Klabunde, Abnormal Rhythms - Definitions, Source on the Internet: Cardiovascular Physiology Concepts, Available at: <http://www.cvphysiology.com/Arrhythmias/A012.htm>.
- [38] An Introduction to the PhysioBank Archives, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/physiobank-intro.shtml>.
- [39] MIT-BIH Arrhythmia Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/mitdb>.
- [40] Creighton University Ventricular Tachyarrhythmia Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/cudb/>.
- [41] MIT-BIH Noise Stress Test Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/nstdb/>.
- [42] MIT-BIH ST Change Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/stdb/>.
- [43] MIT-BIH Malignant Ventricular Arrhythmia Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/vfdb/>.
- [44] MIT-BIH Atrial Fibrillation/Flutter Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/archives.shtml>.
- [45] MIT-BIH ECG Compression Test Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/cdb/>.
- [46] MIT-BIH Supraventricular Arrhythmia Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/svdb/>.
- [47] MIT-BIH Long-Term Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/ltdb/>.

- [48] MIT-BIH Normal Sinus Rhythm Database, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/physiobank/database/nsrdb/>.
- [49] Chart-O-Matic: View signals and annotations, Source on the Internet: PhysioBank, Available at: <http://www.physionet.org/cgi-bin/chart>.
- [50] 8M x 8 Bit NAND Flash Memory, Revision No.0.9, May24,2004, Samsung Electronics, Source on the Internet: datasheetcatalog.com, Available at: http://www.datasheetcatalog.com/datasheets_pdf/K/9/F/6/K9F6408Q0C-H.shtml.
- [51] MIT-BIH Arrhythmia Database Directory, Harvard-MIT Division of Health Sciences and Technology Biomedical Engineering Center, Source on the Internet: physionet.org, Available at: <http://www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm>.
- [52] Ganier, C. Digital-to-Analog Converter on TI MSP430, Source on the Internet: Connexions Web site. Available at: <http://cnx.org/content/m11856/1.2/>.
- [53] MSP430F169 Product Folder, Texas Instruments, Source on the Internet: focus.ti.com, Available at: <http://focus.ti.com/docs/prod/folders/print/msp430f169.html>.
- [54] MSP430-169STK MPS430F169 STARTERKIT DEVELOPMENT BOARD, Development Boards and Tools, Source on the Internet: Olimex.com, Available at: <http://www.olimex.com/dev/index.html>.
- [55] Evaluation Board for MSP430F169, Development Tools/MSP, Source on the Internet: sparkfun.com, Available at: http://www.sparkfun.com/commerce/product_info.php?products_id=58.
- [56] WHY MSP430?, Source on the Internet: MSP430, Available at: <http://elmicro.com/msp430web/whymsp.htm>.
- [57] Mixed Signal Microcontroller (Rev. D) (msp430f169.pdf, 1187 KB) 31 Mar 2005, Source on the Internet: focus.ti.com, Available at: <http://focus.ti.com/docs/prod/folders/print/msp430f169.html>.
- [58] Bill Giovino, TI MSP430 16-bit Low Power Flash Microcontrollers Get Feature & Product Expansion, Source on the Internet: Microcontroller.com, Available at: http://www.microcontroller.com/news/ti_msp430_50newdevices.asp.
- [59] C/C++ compiler and debugger tools for TI MSP430, Source on the Internet: iar.com, Available at: <http://www.iar.com/>.
- [60] D/A Converters Definition Of Terms, National Semiconductor, Source on the Internet: www.national.com, Available at: <http://www.national.com/appinfo/adc/files/ms101157.pdf>.

- [61] Digital-to-analog converter, Source on the Internet: Wikipedia, The Free Encyclopedia, Available at: http://en.wikipedia.org/wiki/Digital_to_analog.
- [62] MSP430x1xx Family User's Guide (Rev. F), Texas Instruments, Source on the Internet: focus.ti.com, Available at: <http://focus.ti.com/lit/ug/slau049f/slau049f.pdf>.
- [63] MSP430-JTAG, JTAG For Programming and Flash Emulation, Source on the Internet: olimex.com, Available at: <http://www.olimex.com/dev/msp-jtag.html>.
- [64] MSP430-JTAG, JTAG For Programming and Flash Emulation, Source on the Internet: olimex.com, Available at: <http://www.olimex.com/dev/images/msp430-jtag.gif>.
- [65] Basic 16x2 Character LCD, Source on the Internet: sparkfun.com, Available at: http://www.sparkfun.com/commerce/product_info.php?products_id=255.
- [66] HD44780U (LCD-II), Dot Matrix Liquid Crystal Display Controller/Driver, Hitachi, Source on the Internet: sparkfun.com, Available at: <http://www.sparkfun.com/datasheets/LCD/HD44780.pdf>.
- [67] MSP430 IAR C/C++ Compiler Reference Guide for Texas Instruments' MSP430 Microcontroller Family, Source on the Internet: iar.com., Available at: <ftp://ftp.iar.se/WWWfiles/msp430/guides/oc430.pdf>.