A DUAL MODE PULSED ELECTRO-MAGNETIC CELL STIMULATOR

A THESIS IN
Electrical Engineering

Presented to the Faculty of the University
of Missouri-Kansas City in partial fulfillment of
the requirements for the degree

MASTERS OF SCIENCE

By

HATEM IBRAHIM RIZK

B.S.E.E., University of Missouri, Kansas City-Columbia, 2000

Kansas City, Missouri
2012

© 2012

HATEM IBRAHIM RIZK

ALL RIGHTS RESERVED

A DUAL MODE PULSED ELECTRO-MAGNETIC CELL STIMULATOR

Hatem Ibrahim Rizk, Candidate for the Master of Science in Electrical Engineering

University of Missouri-Kansas City, 2012

ABSTRACT

This thesis presents the design and test of a dual-modality cell stimulator. The stimulator generates pulsing electric and magnetic fields at programmable rates and intensities. The target application is the stimulation of bone and muscle cells. While electric and magnetic stimulators have been reported before, this is the first device that combines both modalities. The ability of the dual stimulation to target bone and muscle tissue simultaneously has the potential to improve the therapeutic treatment of osteoporosis and sarcopenia. The device is fully programmable and easy to use and can run from a battery or a power supply. In-vitro tests show a 4% increase in protein synthesis 24 hours after the stimulation. These levels are comparable to heat shock stimulation.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering have examined a thesis titled "A Dual Mode Pulsed Electro-Magnetic Cell Stimulator," presented by Hatem Ibrahim Rizk, candidate for the Master of Science degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Walter D. Leon Salas, Ph.D., Committee Chair

School of Computing and Engineering

Ghulam Chaudhry, Ph.D.

School of Computing and Engineering

Reza Derakhshani, Ph.D.

School of Computing and Engineering

CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my appreciation to Dr. Walter D. Leon Salas, Dr. Ghulam Chaudhry, and Dr. Reza Derakhshani for serving as members of my masters committee.

I am especially indebted to professor Leon Salas, who served as chairman of my masters committee and as my thesis advisor. His professional guidance, extensive knowledge, and scholarly critique were an invaluable contribution to this thesis. I appreciate his dedication to his students, teaching, and research. Without his advice, assistance, and encouragement, this work would never be possible.

I would also like to express my appreciation to Dr. Chaudhry, professor and chair of the Computer Science and Electrical Engineering Department, for his continuous scholarly guidance and advice throughout my studies at the University of Missouri in Kansas City.

My gratitude is also extended to Dr. Marco Brotto, Program Director of the Muscle Biology Research Group (MUBIG) and Associate Professor of Nursing and Medicine, for his support, encouragement, and his contribution to my intellectual buildup, especially in the area of biological research. His knowledge and expertise in Muscle Biochemistry, Muscle Fatigue, Physiology, Biophysics, was instrumental in the research, development, and testing of the device presented in my thesis.

Special thanks are extended to the University of Missouri - Center of Excellence in the Study of Dental and Musculoskeletal Tissues (CEMT), for providing the grant that made the research work of this study possible.

Finally, I would like to extend my deepest appreciation to my mother and father, Dr. Soheir Abdelrahman and Dr. Ibrahim Rizk, who have provided me with a constant source of inspiration and motivation. I also thank my wife, Jehan Elshayeb and children, Ibrahim, Noor, and Yousef, for their support and patience during this time. I dedicate this thesis to all of them.

CHAPTER 1

INTRODUCTION

1.1 Motivation of Work

Pulsed electrical stimulation has been extensively employed in muscle stimulation particularly in patients with neuromuscular impairments [1] with the aim of restoring full or partial mobility of limbs. This type of stimulation is commonly known as functional electrical stimulation or FES. Several electrical stimulators have been reported in the literature and some are available commercially [1, 6-12]. Another application of electrical stimulation has been in electro-tactile stimulation to evocate tactile sensation in the skin by applying a local electric current through the skin [4].

Magnetic stimulation has also been shown to produce effects on living organisms. For instance, pulsed magnetic stimulation has been employed in brain stimulation resulting in detectable physiological and behavioral effects both in humans and in animals [15]. Another use of magnetic stimulation is for treatment of bone conditions such as non-union fractures, failed joint fusions and congenital pseudo-arthroses with success rates of 70% to 95% in double-blind studies [16].

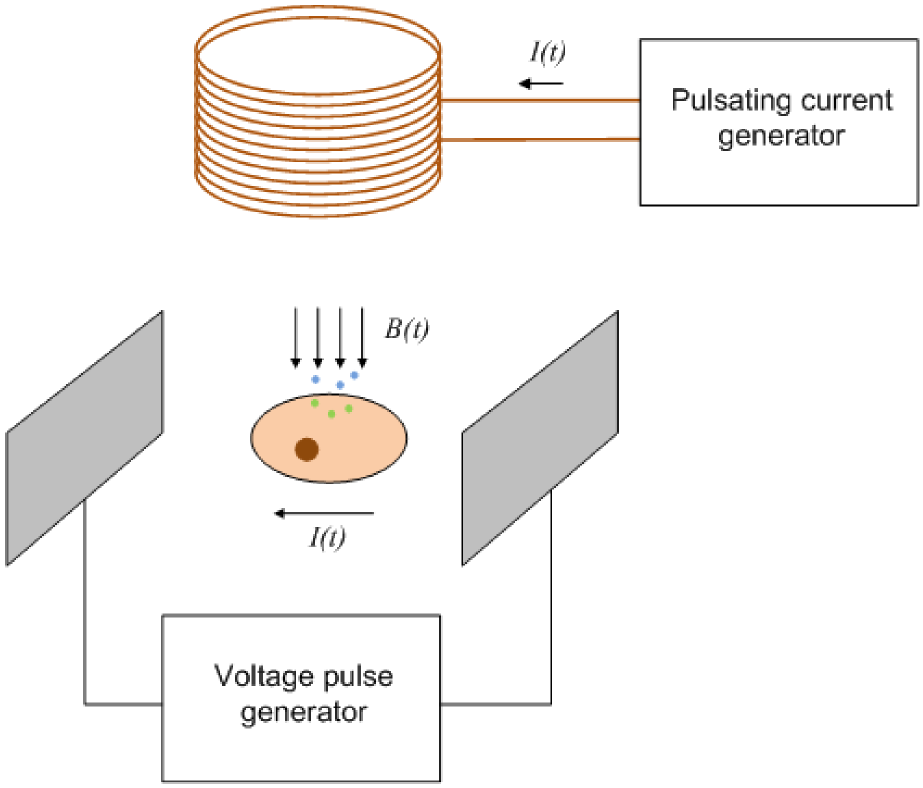


Figure 1: Dual electro and magnetic cell stimulation.

A device to stimulate bone and muscle cell growth and potentially for treatment of bone and muscle injuries is presented. The device, called EM-Stim, is illustrated in Figure 1 above and generates electric and magnetic pulses at programmable intervals. In a human or animal body, muscles and bones are intimately interrelated and the loss of activity in one of them affects the other. This interrelation is especially evident in persons with bone fractures. While the bone is healing, the muscles loose mass due to lack of exercise. Furthermore, when skeletal muscles are not exercised, bone mass density decreases. In these situations, muscle mass can be partially maintained if externally stimulated by applying repetitive electric pulses. Magnetic pulses have proven positive effects on bone healing. The EM-Stim has been designed to generate electric pulses of different frequencies and amplitudes to stimulate muscle growth. It also generates magnetic pulses to induce bone healing. This dual stimulation is a unique feature of the EM-Stim and makes it a promising device in the treatment of bone fractures or muscle injuries. Besides this clinical application, the EM-Stim is being used to study the crosstalk at the cellular level between muscle and bone.

This device will be used to better understand the interplay between bones and muscles. Ultimately, our goal is test this device in animals and humans to fully realize its applications on musculoskeletal injuries and diseases.

1.2 Literature Review

A programmable electronic stimulator for functional electric stimulation (FES) was presented in [10]. The stimulator is programmable, has four output channels and is battery operated. The stimulator is able to generate mono-phasic electric pulses of up to 150 V by employing a flyback power converter. The output stage is an optically isolated push-pull circuit. It also has available a synchronization input and output for cascading with other devices. A specially developed computer program running on Windows operating system is required to program the parameters of the stimulator. Programming is performed through a serial port.

A programmable ramp waveform generator for pulsed electromagnetic field (PEMF) stimulation was designed in [11]. The waveform generator is implemented around the ATMEGA 168 microcontroller which generates pulses with selectable duty cycle that are integrated by an integrator circuit to produce ramp waveforms. The slope of the ramp waveform is controlled by the microcontroller through a digital potentiometer. The slope is set according to the desired magnetic field settings. This generator is intended to work with a coil driver to produce a time-varying magnetic field. Two coils of radius 4.25 cm and 6 cm and a number of turns of 120 and 230 respectively were built to cover the targeted field intensity range (0.5 mT to 4.5 mT).

A high-efficiency circuit for electrical muscle stimulation was reported in [1]. The circuit is based on a boost converter topology in which the reference voltage is a square waveform instead of a constant voltage. By virtue of the feedback loop of the boost converter, a high voltage waveform that follows the shape of the low-voltage square reference voltage is generated. This set up allows reaching efficiencies of up to 98.6%.

In [1] a multichannel direct-synthesized electrical stimulator is described. It introduces an element-envelope method for flexible waveform generation. The method is implemented with the digital signal processor TMS320C32. The signal processor also handles the serial communication with a host computer. The generated waveforms are applied to a constant-current output stage. The employed constant-current source is based on the Holland architecture with a Wilson mirror and is able to provide a linear voltage-to-current conversion with high-voltage compliance. A customized Windows-based program was developed to communicate with the stimulator.

Two circuits for FES are described in [20]. The first circuit is based on an oscillator that generates a train of pulses with controlled amplitude, frequency and pulse width. These pulses are stepped up to the required voltage by a transformer. The second circuit is based on a resonant converter. The resonant converter is composed of two transistors, a capacitor and an inductor. The authors conclude that a stimulator based on the resonant converter results in a more compact circuit since no transformer is needed.

A high-voltage constant current stimulator for electro-tactile stimulation is described in [2]. The stimulator is based on an improved Holland current pump topology in a bridge configuration to allow high impedances and small currents. A compliance voltage of almost 800 V was achieved using commercially available high-voltage opamps (Apex PA-85A). A high power supply of +/- 430 V was obtained by stacking two 200 V and two 230 V floating supplies. This setup created a common mode latch-up problem. The latch-up problem was averted using high-voltage relays that kept the supplies disconnected until 1 second after power up.

Another programmable electric stimulator for skeletal muscle cardiac assist is reported in [21]. It is built around the MC68HC811 microcontroller which sends the control signals to the analog electronics to generate the desired pulse sequence. It also interfaces with a custom Windows-based program for interaction with the user and to rapidly develop new stimulation sequences. The analog electronics is essentially an R-2R digital-to-analog converter implemented with commercially available op-amps. A boost DC-DC converter based on the MAX633 regulator was employed to generate +/- 15 V to power the analog part from a 5 V supply.

The programmable stimulator subject of this thesis was developed to facilitate both magnetic and electrical therapy simultaneously. This allows for studying the joint effects of both types of treatment on muscles as well as the bone structure they are attached to, especially in light of cross-talk relations which studies have shown.

While some stimulators, generate mono-phasic electrical signals, the EM-STIM device, generates a bi-phasic signal which helps reduce static charge build-up observed in some patients treated with mono-phasic signals. This was achieved by integrating an H-Bridge into the design at the output stage of the device where electric applicator probes are attached. The H-Bridge is controlled by the onboard programmable microcontroller. This allows precise control of output signal characteristics and ability to modify on the fly by adjusting program parameters via the command line prompt while connected to a computer or by relying on visual indicators displayed in the on board screen and using the on board push buttons for input. Previous stimulator applications relied on hard wired designs such as resonators that did not offer the level of flexibility within EM-STIM to control signal shaping dynamically. The program coded onboard the microcontroller also allow it to control a high-current transistor to apply and electric current on attached coils to generate magnetic fields at pulsed intervals depending on the desired application sequence and in between applying electric pulse trains through the electric probes that induce currents onto therapy subjects. For the combined therapy application offered by EM-STIM, two types of signals need to be generated within the EM-STIM circuit: A high voltage signal for electrical therapy needed to induce low electric current through material that typically has high resistance properties, such as the skin of a patient, or fluids containing myocites and osteocites in a research flask. The other type of signal is a high current signal that can be coupled onto a coil to generate the magnetic fields for magnetic therapy. These two types of signals are generated using separate Buck and Boost circuits implemented within the design of EM-STIM. Both operate from the attached portable 12 V battery and are able to generate up to 10 A of current needed for magnetic therapy well as up to 60 V separately for electric stimulation respectively. In the current EM-STIM circuit, output current into the coil has been limited to 0.7 A to prevent overheating of the coil.

CHAPTER 2

HARDWARE DESIGN

2.1 Introduction

This chapter discusses the overall design of the Electro-Magnetic stimulator referred to here as EM-STIM and the technical requirements influencing its design. We start by listing the main application of EM-STIM, and then proceed to discuss the high-level synthesis of the EM-STIM design and delve further into the design of each sub-component and the driving factors behind its selection, expected operation and performance metrics.

2.2 Background

The application requirements were gathered during preliminary meetings between the UMKC School of Computing and Engineering – Integrated Circuits and Systems Research Lab (ICSR) and the UMKC School of Nursing – Muscle Biology Research Group Lab (MUBIG). This was followed by a six month immersion period in which I received hands-on training on state of the art biological research techniques and taking part in daily experiments. These experiments involved photometer measurements of fluorescence in cells, Calcium Imaging, and numerous molecular and cell biological studies that involved reverse transcriptase-PCR (RT-PCR) , protein electrophoresis, Western Blot, protein mass analysis, and the applications of muscle fiber extraction techniques. I also assisted in maintaining C2C12 muscle cell cultures for experiments and development of different cell growth media for various experiments. It is during this emersion period where I was able to further understand the biological laboratory research environment and hone in the requirements of the EM-STIM in accordance to the lab research needs and based on enhancements to what was found in an extensive survey of existing literature involving electrical and magnetic stimulation in a biological laboratory environment and commercial products on the market to date.

2.3 Application Requirements

This primary aim behind developing the EM-STIM product is to empower researchers in any biological research laboratory environment to conduct extensive studies on the effects of applying simultaneous Electric and Magnetic pulses on biological tissue and to better understand how they respond to this combined application to seek therapies that can be validated and used in the future to alleviate suffering and improve the quality of life for humanity. Thus, in addition to the dual stimulation requirement, and keeping in mind the device will be used heavily in research and the iterative nature of these studies, we needed to build a device that is highly configurable in a lab environment and gives the researcher a lot of control and flexibility over a wide range parameters affecting the stimulation output of the product. The device also needed to be user friendly, safe, and easy to operate by a non-engineer in a lab environment, where we also observed the long term nature of some of these studies. During the studies researchers attempt to assess impacts of stimulation on the differentiation of cells and any enhancements to their Myogenic program and this could extend over a period of several days, even weeks, shall this product or another based on concepts in this design be used in future vivo clinical experimentation. Therefore, it was important to develop a design with minimal power consumption requirements, portable in nature, and can support long term studies and can be developed in the future to support en vivo clinical validation studies. In the analysis of different media cultures and their contents it was observed that it is quite possible to have different impedance conditions depending on the type of media used and its content. This is also found to be the case with muscle tissue depending on the amount of fat content, the impedance varied [31]. Thus, building a feedback system to monitor the test environment and report conditions to a main processing unit was essential for safety as well as to help regulate our stimulation output and gives have predictable output and thus higher statistical significance when assessing biological response in experiments involving EM-STIM. Lastly, a rapid and cost effective prototype of the design was requested by our lab collaborator to ensure a quick commencement of in vitro biological testing procedure a part of the funding process and extending current research to an in vivo environment. In the next section we delve more into the design process and how it was derived based on the application requirements that were listed in this section.

2.4 Design Synthesis and Implementation

A quick analysis of our application requirements enables us to deduce in general terms the main subcomponents of the circuit. Once we were able to define those main subcomponents, we were then able to logically group them into three main functions as shown in Figure 2. The output stage of EM-STIM is where stimulation signals are conditioned before application through external probes. In the case of in vitro testing, magnetic fields are generated within an external coil that is attached below or above a flask containing cultured cells. At the same time electrical stimulation currents are applied via external electrodes embedded in the culturing medium. An H-Bridge and a Power MOSFET are used to shape electric and magnetic stimulation signals prior to injection into the external stimulation probes.

The power system contains main power supply source for EM-STIM which is a 12 V NiMH battery pack. It also encompasses power electronics circuitry that performs various voltage conversion functions needed for each type of stimulation prior to injection into the output stage. It is in this part of the circuit where raw stimulation signals are generated before being injected into the output stage for pulse shaping. Electrical stimulation requires a high DC voltage signal with low current drives and this is fulfilled by a boost DC-to-DC voltage converter. On the other hand, magnetic stimulation requires high current drive and low voltage signals and this is fulfilled by a buck DC-to-DC voltage converter. Low drop-out voltage regulators allow constant and stable and constant voltage supply to multiple ICs in the design regardless of the state of the battery charge.

The final subsystem is the control system built within a low power programmable microcontroller that manages the output stage to produce highly customizable stimulations patterns. These patterns can vary in amplitude, frequency, and the type of therapy, weather it is electric in nature, magnetic, or perhaps a combination of both. The control system also processes feedback from sensor circuits in the output stage including monitoring the output current injected at the out put stage and the current charge status of the battery pack. It is possible in the future to modify EM-STIM to adapt the output signals based on real-time therapy conditions; an example would be to adjust the applied voltage based on the impedance of the stimulation medium to achieve a constant stimulation current. Configuration management is enabled via an external connection to a PC.

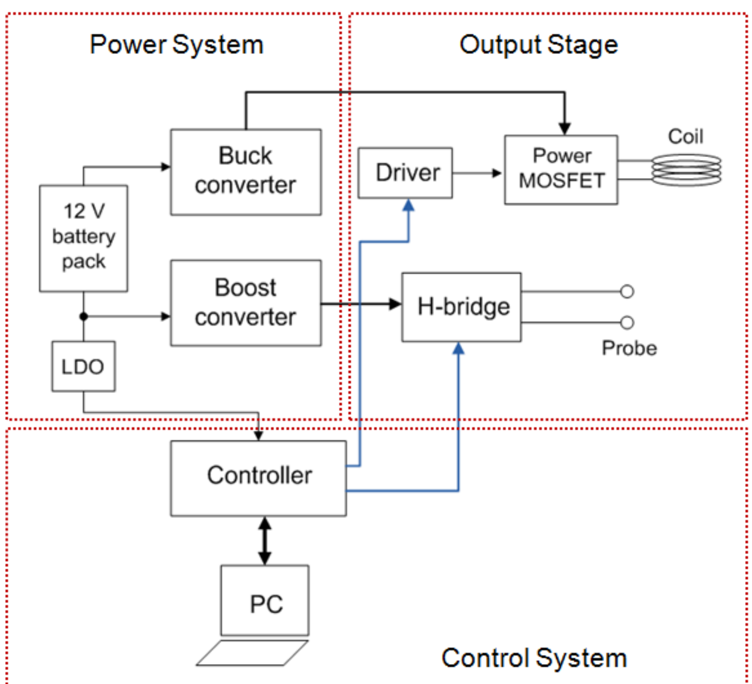


Figure 2: Block diagram of the dual-mode cell stimulator.

2.4.1 Output Stage

2.4.1.1 H-Bridge

A sub-circuit of the output stage, the H-Bridge is utilized in conjunction with the MCU to generate electrical pulse trains for final application through externally connected electrodes. It can also be used to invert electrical stimulation pulses and shut off electrical stimulation all together as needed in the current mode of operation. The H-Bridge receives its raw input from a boost DC-to-DC voltage converter in the form of a high voltage DC signal and manipulates it based on control signals transmitted over MCU output pins which are physically connected to it. The conditioned signal is then transferred over to the external probes.

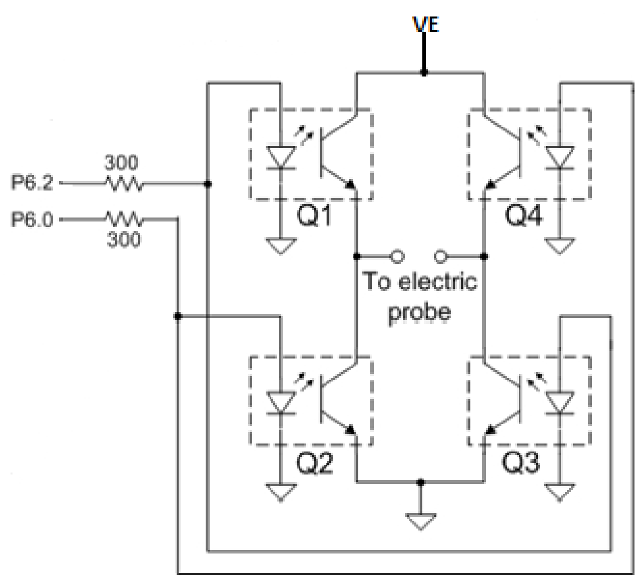


Figure 3: Schematic diagram of the H-bridge sub-circuit.

Two pairs of OPTEK Technology OPIA601 optocouplers are connected to form the basic structure of the H-Bridge as shown in Figure 3. The first pair (Q1, Q4) is connected to the raw electrical stimulation signal at the source. The second pair (Q2, Q3) is connected at the sink to the common ground. The point of interconnection between the sinks of (Q1, Q4) and the sources of (Q2, Q3) is where external electrical stimulation probes are connected. The gates of the optocouplers are connected to output pins on the MCU through which the firmware can control the states of the optocouplers and influence the direction of current flow passing through the external probes. For example, transmitting a low voltage signal to the gates of (Q1, Q3) while simultaneously transmitting a high signal to gates of (Q2, Q4) will shutoff (Q1, Q3) and enable (Q2, Q4) which effectively allows the current to flow in the direction from Q4 to Q2. Reversing the control signal status leads to reversal of current (from Q1 to Q3) and thus creates DC voltage inversion. Transmitting a low signal to all gates shuts off all optocouplers and stops the current flow through the stimulation probes.

In this particular implementation we are using OPTEK OPIA601 optocouplers. Using optocouplers as oppose to transistors offers a high level of protection to the MCU from overvoltage damage. This could occur if the MCU pins were to come into contact with high voltage signals from the boost voltage converter due to any hardware failure. This specific model provides up to 5000 V of isolation. Figure 4 below shows the prototype implementation of the H-Bridge circuit and basic pin connectivity. The blue and white jacketed wiring provides connectivity to the external probes, while the red and black wires connect the boost voltage converter output and to the system ground consecutively. In addition, it also shows the brown and yellow lines connecting to pairs of optocouplers that need to be activated or de-activated at the same time at any given moment, based the firmware control flow and upon receiving a high or low signal from the MCU pins. As can be seen in the implementation photo in Figure 4, each pair of optocouplers are connected to one output pin on port 6 of the MSP430FG4618. A 150 Ohm current limiting resistor is installed at the input of each optocoupler gate and is used with the output of the MCI pin to source a constant current of 12 mA in each gate when the output of the pin is raised to high. This particular MSP430 MCU allows for a maximum of 48 mA combined output for all outputs which we are able to comply with in our current configuration as we only draw a maximum of 24 mA at any given time. This particular optocoupler was selected due to its high Current Transfer Ratio (CRT) which is specified with a maximum value of 9000%. The CRT defines the ratio of the collector current in the photodiode portion of the optocoupler in relation to the input diode current. In our current design, we limit our input current to 12 mA by choosing a choosing a diode input resistance of 150 Ohms. Thus with this drive level, we are able to achieve the maximum specified collector current of up to 150 mA. This is also the maximum stimulation DC current E-STIM is designed to apply to electrical stimulation probes at the maximum rating of 60 V which is the maximum voltage output of the boost voltage converter and further below the 300 V Collector-Emitter rating of the optocoupler.

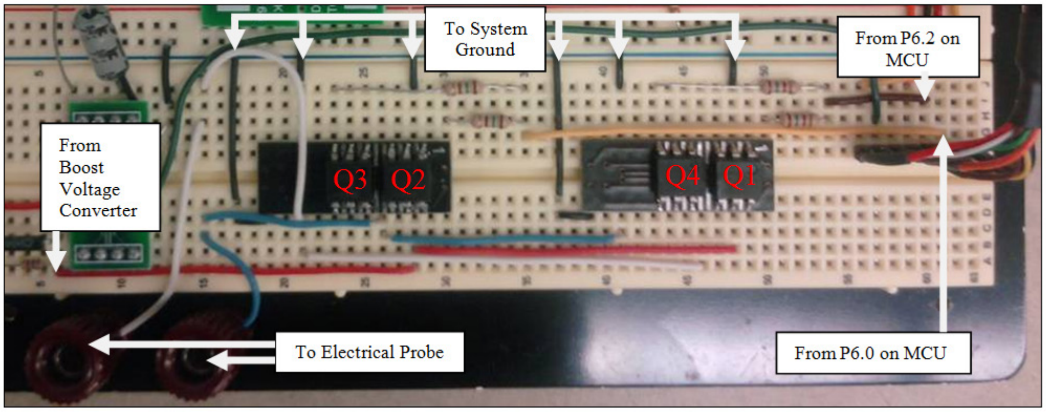


Figure 4: Prototype implementation of the H-bridge sub-circuit of the output stage.

2.4.1.2 Coil Design

The EM-STIM is designed to use an external coil to generate the magnetic flux used in Pulsed Electro-Magnetic treatment. A review of the literature on pulsed magnetic stimulation reveals that a wide range of magnetic field intensities have been employed for bone healing ranging from 0.034 μT to 15 mT [13,14]. The maximum current level that is needed to generate such magnetic fields can be calculated from Ampere's Law. Considering a coil of N turns and radius R, the magnetic field B at a distance r from the center of the coil is given by:

$$B = \frac{\mu_0 I \cdot N \cdot R^2}{2(R^2 + r^2)^{3/2}} \quad (1)$$

where μ_0 is the permeability of free space and is equal to $4\pi\times10^{-7}$ H/m. To obtain a magnetic field of 15 mT [14][13] at a distance of 5 mm with a coil of 3cm of radius and 100 turns, a current of 7.5 A is needed. Considering a DC coil resistance of 0.2 Ω, a voltage of 1.7 V will be needed to produce a current of 8.4 A through the coil. Thus, a low voltage source with high current capabilities is sufficient to generate needed flux for magnetic therapy and the selection of a high efficiency buck voltage converter for integration in the design. Driving the coil with a constant current of 8.4 A would yield a fixed magnetic intensity of 15 mT at 1 cm from the center of the coil: This magnetic flux intensity is constant and cannot be adjusted without varying the amount of current sunk into the coil. High-current regulation would typically require complex circuitry and may not have the same linear effect on magnetic intensity. In the EM-STIM design, we take advantage of Pulse Width Modulation (PWM) to adjust the duty cycle of the high DC current signal before injection in to an external coil. By doing so, we are able to dynamically adjust the average DC current value and thus alter the resultant intensity of the magnetic flux produced by the external coil. This can be done on the fly by simply adjusting software parameters in the EM-STIM firmware. In the final implementation of the coil implementation, enameled copper wire gauge #30 AWG is used to build an external coil for magnetic stimulation probe that can be placed above or below stimulation flasks. To avoid over-heating concerns, the buck voltage converter current was limited to 0.7 A which yields a calculated flux intensity of 1.4 mT at a distance of 5 mm from the center of the coil according to the magnetic flux equation above.

2.4.1.3 Power MOSFET and Driver

A Vishay IRF640S Power MOSFET is used at the output stage of the EM-STIM circuit and in series with the external coil. This arrangement allows us to control the manner in which current is injected onto the coil by switching it based on output from within the MCU firmware. Due to the high speed nature of the PWM control signals generated by the MCU we needed to select a MOSFET with fast switching capabilities. We also needed a MOSFET that can support the high currents generated by the buck voltage converter and be capable of injecting those high currents into an external coil. In analyzing the different Power MOSFETs commercially available, it was determined that the IRF640S IC with fast switching capabilities, high power support, and low resistance is an optimal choice for integration onto the EM-STIM circuit. The fast switching capability allows it to react efficiently to our PWM signal used to modulate the field intensity produced by the coil. More importantly is the fact that it supports a continuous drain current of up to 18 A, more than enough to support our current prototype target output of 0.7 A. It can also support higher current drives needed for higher level of magnetic stimulation with future enhancements to heat control, such as with using higher gauge coils and added external cooling. In addition, with minimal modification, EM-STIM can support current drive needs of a large number of stimulation currents if needed in a research study. Figure 5 shows the final implementation of this magnetic stimulation output stage sub-circuit.

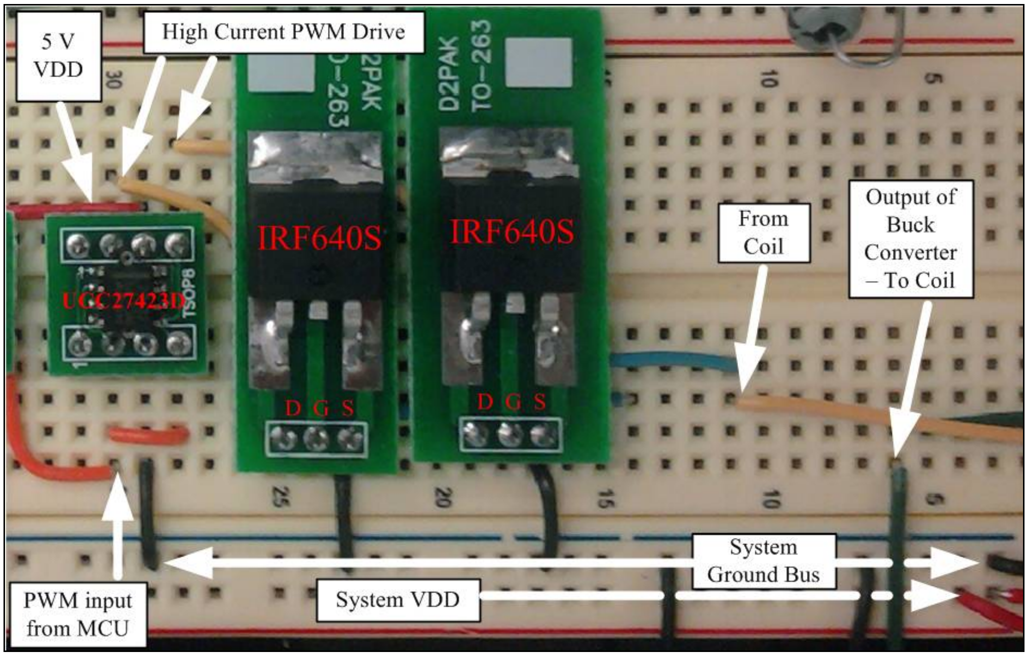


Figure 5: Prototype implementation of magnetic stimulation output stage.

Although our aim is to use PWM pulse trains generated at the MCU output port as input to our Power MOSFET modulating the high-current signal injected into the coil, we are not able to do so due to the fact that the MSP430 MCU output pins are not designed to sink (nor source) high currents into external peripherals and thus cannot provide enough current to drive into the gate of the IRF640S IC. Therefore, we need to rely on a driver circuit that can take the MCU PWM output and apply it with the needed high current drive at the MOSFET gate. These capabilities are found in the TI UCC27423D high-speed low-side MOSFET driver which has a fast response time and can sink up to 4 A of current into at low supply voltage levels, in our case 5 VDC supplied by the onboard 5V LDO.

2.4.1.4 Current Feedback Monitoring

EM-STIM is designed to generate bi-phasic electrical stimulation pulses having a variable voltage of up to 60 V. The stimulating current is constant and can reach a maximum of 150 mA. Nevertheless, it will vary depending on the impedance measured across pairs of electrical stimulation probes connected to the device. Due to the variable impedance properties associated with different muscle tissues and cell growth media, a feedback can help monitor applied currents to ensure safe and proper operation. It can also be used as part of a future enhancement and along with other circuit modification to dynamically adjust applied voltages to achieve target constant current rates. The current-sense amplifier LTC6102 is employed to monitor the current through the H-bridge. The LTC6102 monitors current via the sense resistor Rsense of 0.05 Ω. The small voltage developed across the sense resistor is converted by the LTC6102 to an output current at pin OUT that flows through ROUT (5.1 k Ω) creating an output voltage proportional to the current through the bridge. This set up allows a small sense signal on a large common mode voltage to be translated to a ground-referred signal. The output voltage from the current-sense amplifier is read by the microcontroller's internal ADC as shown in Figure 6 which shows the actual connectivity and implementation of the current sense amplifier.

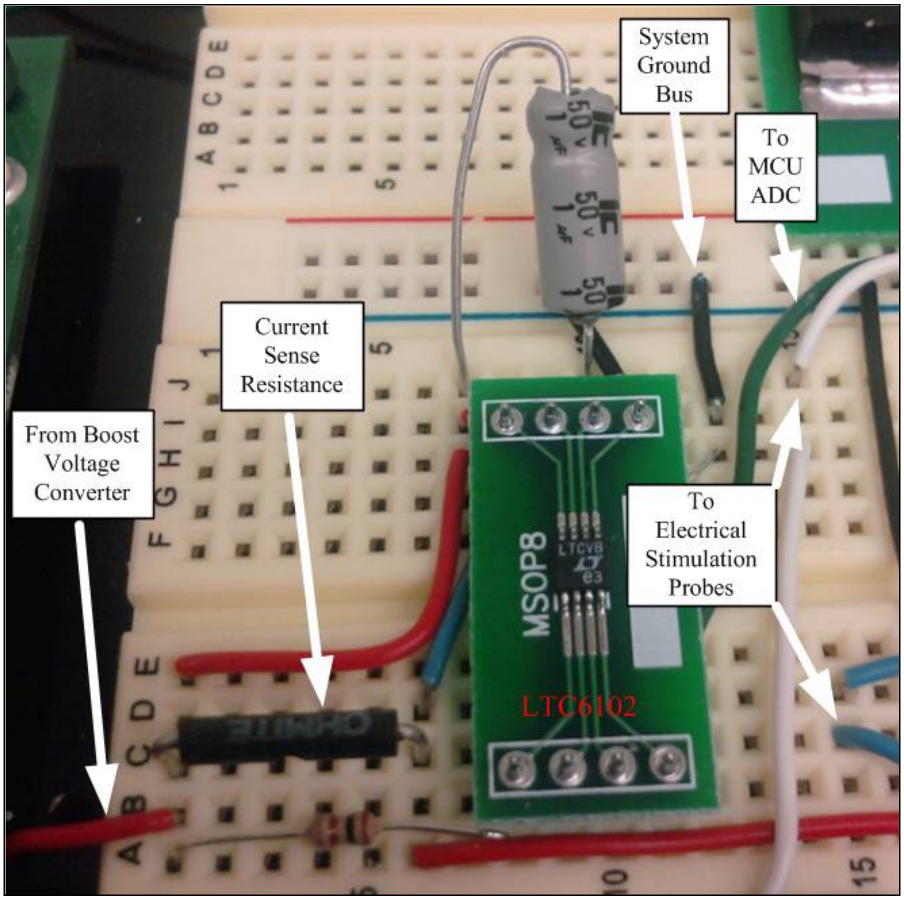


Figure 6: Prototype implementation of current sense feedback circuit.

The battery voltage is also monitored via a resistive voltage divider whose output voltage is fed to an input channel of the microcontroller's internal ADC. The current voltage and current levels are optimized for in-vitro stimulation and thus the current feedback mechanism is limited to monitoring the stimulating current and detecting malfunctions or short circuit conditions. Stimulator output stages can be revised to shutdown when monitored currents are found to exceed a specific level for safe operation. The microcontroller can easily be programmed in the future to adjust boost voltage levels.

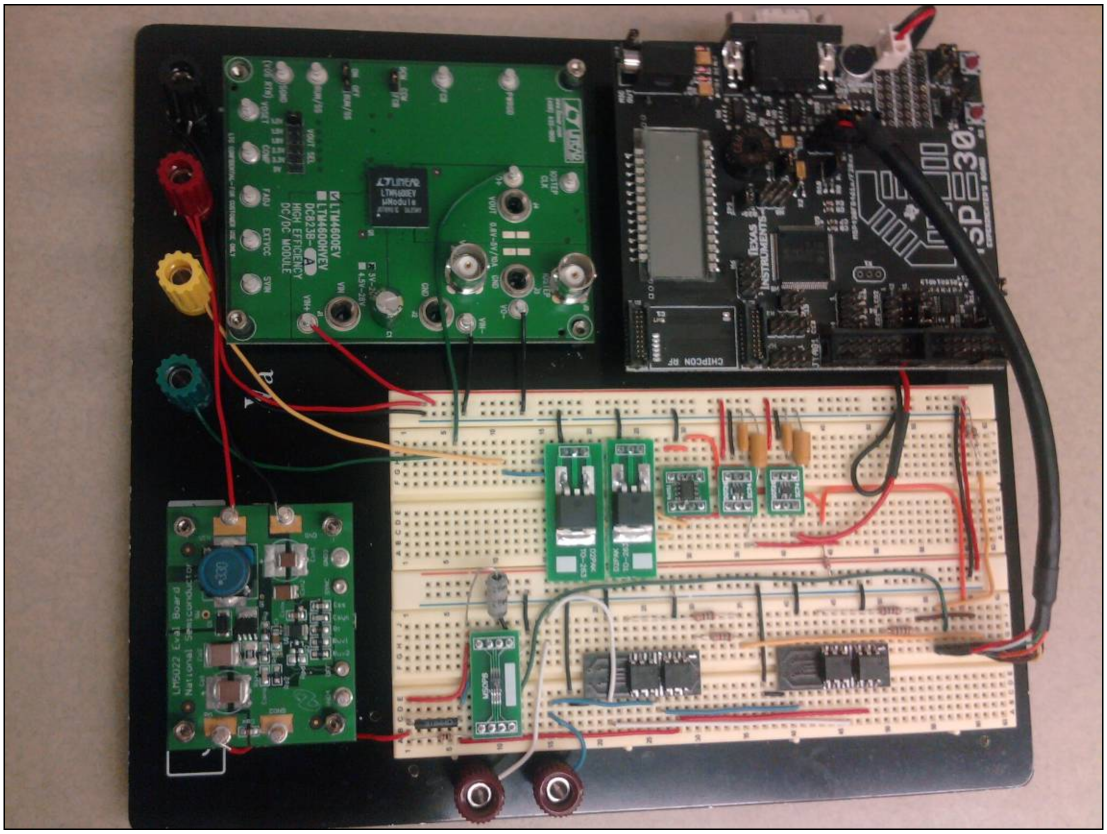


Figure 7: Prototype implementation of the EM-STIM device.

2.4.2 POWER SYSTEM

In selecting the primary power source for the EM-STIM device, consideration was made the safe operation of the device especially during any future use of the device while performing en vivo studies. It was also important to have an increased level of mobility. For these reasons, EM-STIM was designed to be fully portable and isolated from the power grid during its operation.

2.4.2.1 NiMH Battery Pack

EM-STIM relies on a 12 V Nickel-Metal Hydride (NiMH) rechargeable battery pack to power its peripherals during stimulation sessions. The maximum capacity of this battery pack is 5000 mAh. The battery pack is mounted below the prototype board and is connected to a bus line providing a 12 V power source to all peripherals.

2.4.2.2 Smart Charger

An off-the-shelf rapid charger is used to charge the EM-STIM battery pack. The charger accepts a 120-240 VAC input signal and sets proper charging voltage automatically. It includes built in circuitry to protect the battery from high-current, short circuit and reverse polarity conditions. An external temperature sensor monitors battery temperature and provides additional protection from overcharge conditions. The temperature sensor should be mounted to the side of the battery at all time.

2.4.2.3 Low Dropout Regulators (LDOs)

Low Dropout Regulators (LDOs) are integrated in the design of the EM-STIM to regulate the supply voltage for all Integrated Circuits (ICs). In general, LDOs are used to provide a stable power supply voltage independent of load impedance, input-voltage variations, temperature, and time [30]. LDOs have the ability to maintain required system voltage independently of the state of battery charge and with small differences between the system and load voltages. A linear regulator typically consists of, a reference voltage, a voltage divider, a series pass transistor, and a feedback amplifier. The feedback amplifier along with the series pass transistor act as a voltage controlled current source that will vary its current based on feedback from amplifier which constantly compares the output voltage to the reference voltage and adjusts its current which drives the gate of the series pass transistor thus adjusting the voltage across it. This mechanism enables us to create a controlled voltage drop as we are able to create a variable resistor within the regulating device that adjusts itself continuously to maintain constant voltage across the load regardless of its impedance and the state of charge of the battery pack.

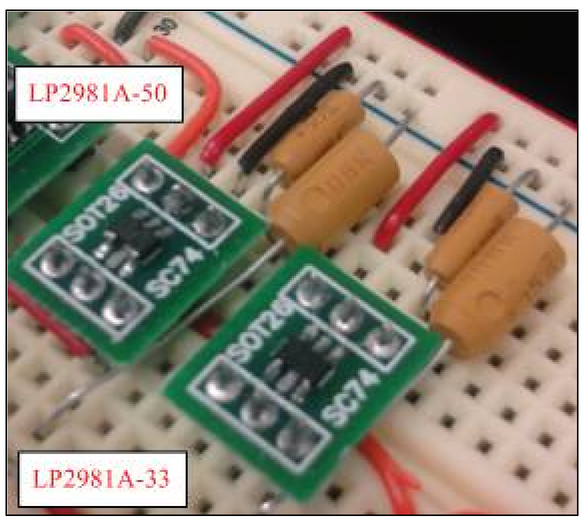


Figure 8: Prototype implementation of 3.3 V and 5 V LDOs.

Two separate LDOs (Figure 8) are implemented in the EM-Stim circuit. The first LDO regulates a 12 V DC signal coming from the battery pack and provides a constant 3.3 V required for operation of the MCU and associated circuitry. The second LDO regulates the battery voltage to a 5 V DC signal required for operation of the MOSFET driver IC.

2.4.2.4 DC-DC Buck Power Converter

Magnetic stimulation requires a voltage source of relatively low output voltage but must be able to provide a high current drive. These two requirements are achieved with a high-efficiency buck converter that translates the nominal battery voltage of 12 V to 1.8 V be able to support a maximum output current of 10 A. This buck converter was built around the LTM4600 (Figure 11). The LTM4600 is a high efficiency, high density switch-mode step-down power module that operates over an input voltage range of 4.5 V to 20 V and supports an output voltage range of 0.6 V to 5 V with a maximum output current of 10 A [26]. The output voltage is set via an external resistor relative to the input voltage. The package integrates all necessary components including input and output filters. Only bulk input and output capacitors are needed yielding a simple PCB layout. In the initial prototype, we include an LTM4600EV evaluation board for a quick implementation of the buck voltage converter as shown in Figures 7 and 9. The output voltage selector was set to 1.8 V on the implementation board which is in order to achieve a maximum magnetic intensity level of 1.4 mT as calculated in section 2.4.1.2.

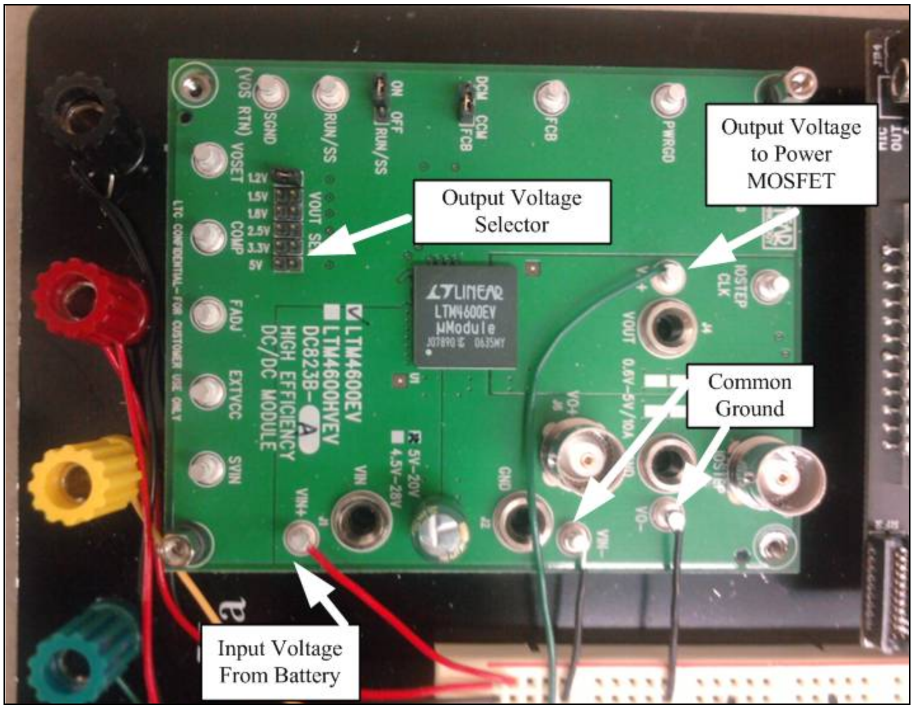


Figure 9: Implementation of a buck converter using an LTM4600 evaluation board.

2.4.2.5 DC-DC Boost Power Converter

The needs for electrical stimulation are different from magnetic stimulation in that it requires a high voltage output (between 40 V to 60 V) and relatively low output currents (0 to 150 mA). A separate boost converter performs the function of stepping up the battery voltage from 12 V and up to 60 V applied to external platinum probes attached to the EM-STIM H-Bridge. The boost voltage converter was implemented with the integrated circuit LM5022 (Figure 11). The LM5022 is a high voltage low-side N-channel MOSFET regulator [27]. Output voltage regulation is based on current-mode control, which eases the design of loop compensation while providing inherent input voltage feed-forward. It includes a startup regulator that operates over a wide input range of 6 V to 60 V and a PWM controller designed for high speed capability including an oscillator frequency range up to 2 MHz. An LM5022 evaluation board was used in the EM-STIM prototype for a quick implementation of the boost voltage converter as shown in Figure 10.

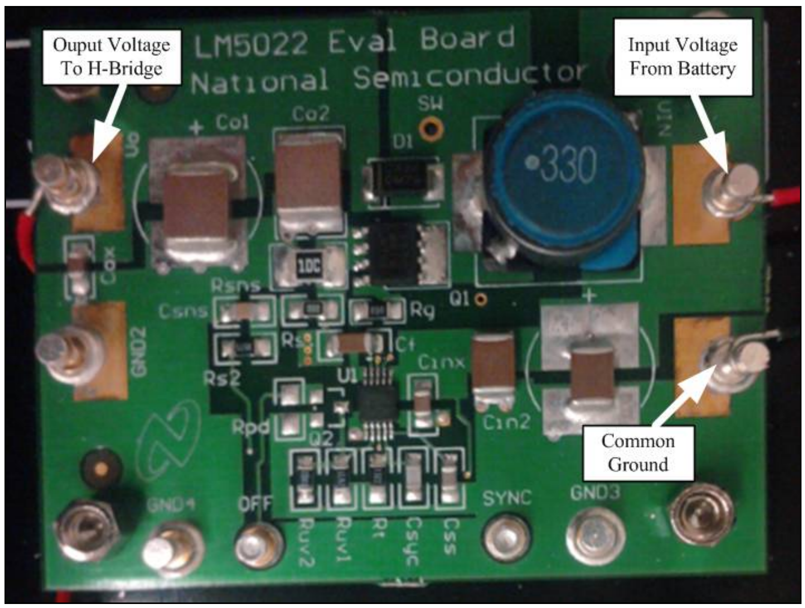


Figure 10: Implementation of a boost converter using an LM5022 evaluation board.

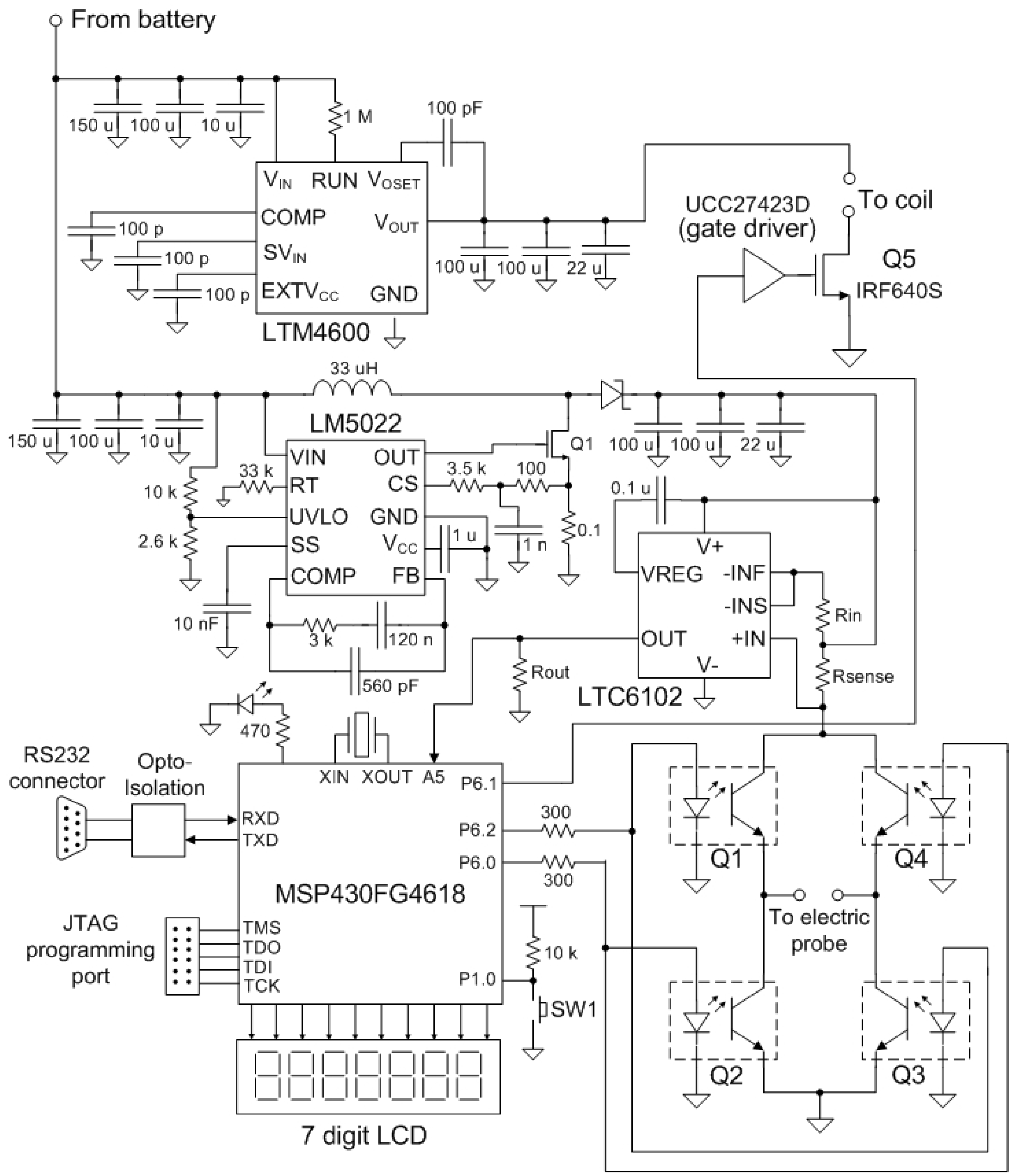


Figure 11: Detailed schematic diagram of the dual mode stimulator.

2.4.3 Control System

EM-STIM relies on a Texas Instruments MSP430 microcontroller for wave pattern generation, power management, LCD control functions, sensory functions, as well user driven interaction through built peripheral controllers such as UART and LCD controllers among others. Features built within this microcontroller allow for reduced power consumption making it an excellent candidate for portable electronic applications.

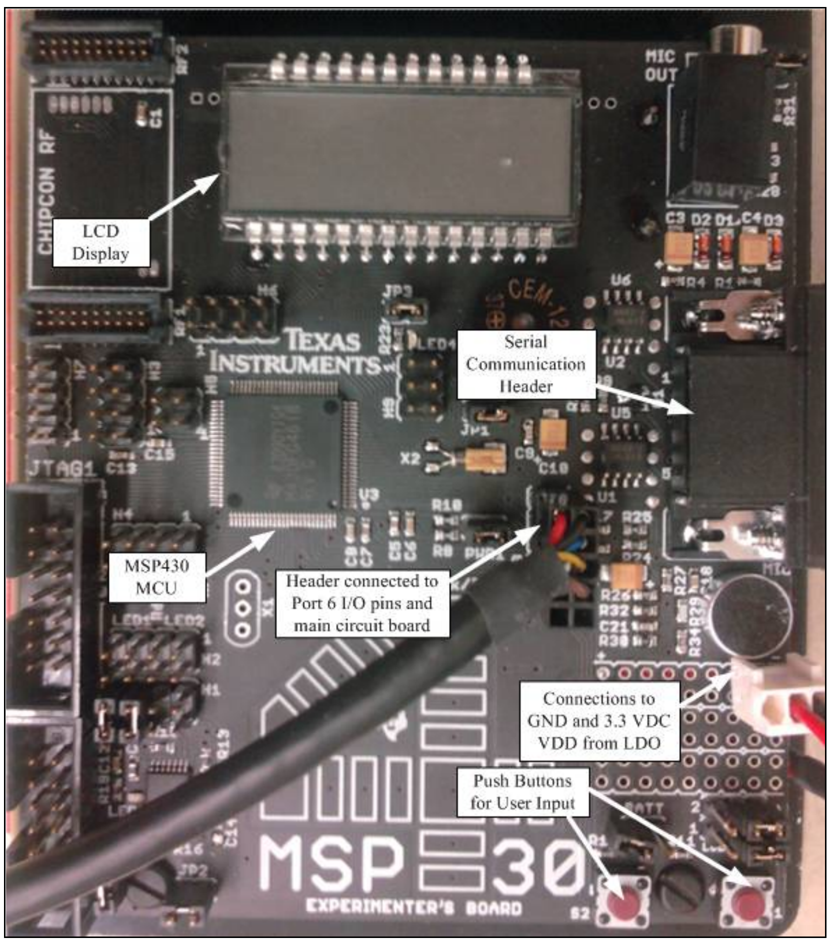


Figure 12: Implementation of the control system using evaluation board.

Also, built in timers and PWM capabilities allow for precise and efficient pulse train generation. An onboard LCD controller allows for quick and efficient interfacing with an LCD display with minimal added components. In addition, built in Analog-to-Digital Converters can be used in feedback control applications. These features made it in excellent choice for integration into the EM-STIM design to reduce power consumption, produce precise outputs, and minimize physical size of the device. The EM-STIM prototype includes an MSP430FG4618 Experimenter's Board in its initial implementation as shown in Figure 12.

2.4.3.1 Analog-to-Digital Converter

Included in the MSP430FG439 microcontroller is a high-performance 12-bit analog-to-digital converter (ADC) module [28]. EM-STIM relies on this built-in ADC to monitor voltage across the battery pack, as well as current running through the H-Bridge. A built in timer will trigger the ADC to sample voltage from pins 6.7 and 6.5 of the microcontroller, which are physically connected to the battery pack and the output pin of the current sense amplifier. The resultant data is then stored in two internal memory registers and used by other of functions in the EM-STIM firmware.

2.4.3.2 LCD Controller

EM-STIM utilizes an LCD Controller to display useful information on its operational status and during interaction with operators. This LCD Controller is built within the MSP430FG439 microcontroller device and creates segment voltages automatically based on values stored in an internal display memory. An integer representing battery voltage sampled by the ADC is continuously displayed on an LCD display built into the EM-STIM prototype board (Figure 12). Also, battery indicators built into the LCD display are updated based on measured battery voltage levels which are compared to five different voltage levels to show a visual indicator of available battery charge

2.4.3.3 Serial Communications Interface

A universal asynchronous receive/transmit (UART) peripheral interface provides a mean for EM-STIM users to fully interact with the device and drive stimulation studies through a user friendly menu-driven command line interface that can run within any standard terminal emulation program running on a connected PC. The UART interface hardware is embedded within the MSP430FG439 MCU and is utilized by its firmware to interface with any connected PC through its serial communications port. The microcontroller will detect the connection automatically and will in turn, transmit menu information to be displayed to the user within the terminal emulation program. In the event that user input through the keyboard is detected, the emulation program will transmit that input over the serial communications interface to a buffer within the EM-STIM MCU and an Interrupt service request (ISR) is generated. Once an ISR is generated, the user input will be read from the buffer, decoded, and a sequence of instructions will be executed based on the selected user options and the associated menu commands at the time user input was recorded. The menu options include the ability to display system settings, modify stimulation parameters such as number of pulses and rest time, and the ability to perform troubleshooting actions in a test mode configuration. Troubleshooting actions include: turning on and off select pairs of optocouplers forming the h-bridge, inducing current through the coil, and display current sense amplifier measurements.

2.4.3.4 Low Power Mode

EM-STIM relies on low power features in the MSP430 microcontroller family that make it suitable for portable electronic applications. The MSP430FG439 operates in 6 different software selectable power modes based on the need to balance ultra-low power consumption, speed and data throughputs, and the minimization of individual peripheral power consumption. Figure 13 shows the approximate current draw of the MCU in different power modes. EM-STIM operates in Low Power Mode 0 (LPM0) while generating pulse trains during stimulations and the CPU is activated only when the current timer expires signaling a new state. In this mode, the microcontroller is able to scan all ports in low power modes and wake-up with minimal delay to respond to any interrupt events. This feature reduces current consumption and thereby increases the number of hours of stimulation per battery charge. In this mode, the MCU current consumption is limited to 55 μA.

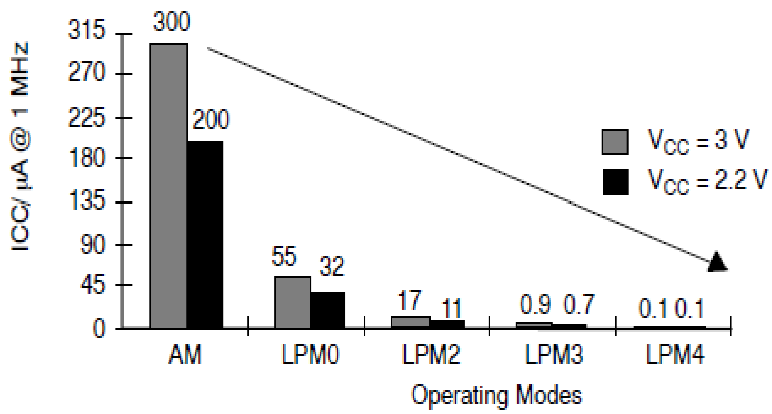


Figure 13: MSP430 microcontroller power consumption per operating mode

2.4.3.5 Timers

Timing of pulse trains generated is made precise by the use of timers built into the microcontroller. The MSP430FG439 comes with 4 different types of timers and support PWM outputs and interval timing. Stimulus control patters are generated within the Timer-A interrupt service procedure. The timer is set to operate in the Up mode and while operating in this mode, it will count up to the value stored in the TACCR0 register at which point an interrupt event is generated and the state machine transitions into a new state. The TACCR0 register is set within the firmware when a new state is reached based on the pulse timing parameters.

2.5 Mass Production Cost Analysis

It is estimated that the EM-STIM device will cost approximately $400 a piece in a minimum production run of 100 units (Figure 14). This cost figure was derived based survey of major electronic supplier pricing databases and estimated manufacturing production costs. This includes a miniaturized PCB board production costs, heavy duty case, and external stimulation probes. It is believed that this cost can be reduced by negotiating material costs and substituting some components with lower cost equivalents.

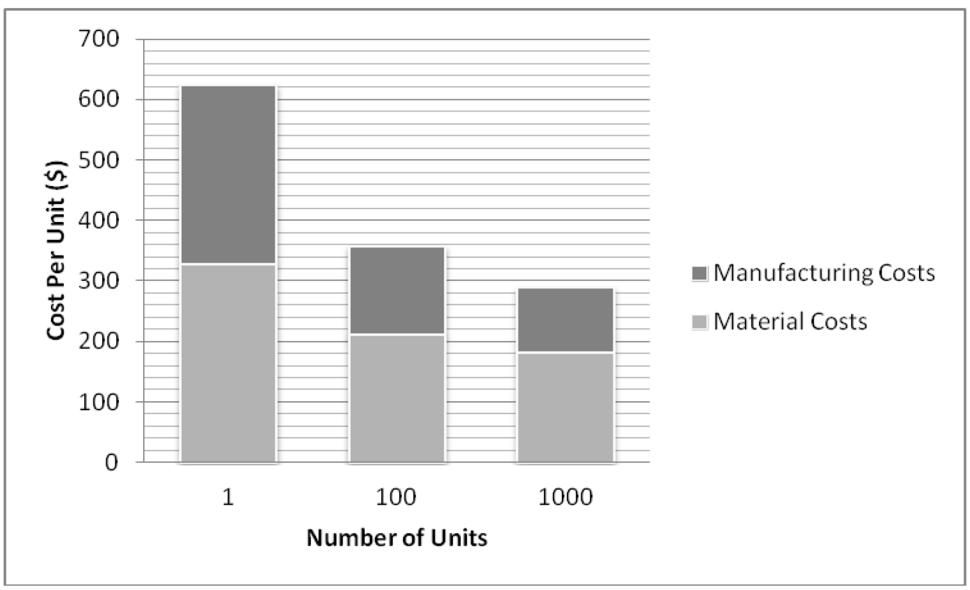


Figure 14: EM-STIM cost per unit relative to production quantity

An initial startup would require sales and marketing resources as well engineering and technical support. A seed investment o f $850,000.00 would enable a startup to manufacture 100 units at $40,000 and hire an engineering scientist and a sales and marketing manager for a period of up to two years and cover running expenses during that time for a corporate office. These funds would also cover initial product development costs associated with the EM-STIM prototype which amounted to $40,000. This cost figure also includes $200,000 to cover test equipment needs. Further clinical and lab testing would require additional funds to be secured by a venture capitalist firm or through a cooperative agreement with a research institution open to a revenue sharing agreement.

CHAPTER 3

FIRMWARE DESIGN

3.1 Waveform Generation

EM-STIM relies on a state-machine algorithm to generate trains of electrical and magnetic stimulation (Figure 15).

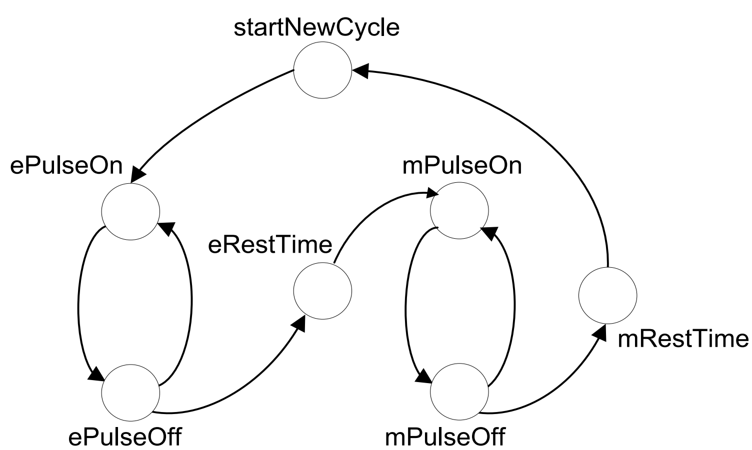


Figure 15: State machine diagram.

The algorithm makes use of several parameters to control the timing of each pulse train including: pulse repetition, pulse duration, time between pulses and rest period. These parameters are set by the user according to any desired stimulation pattern and can be applied independently with respect both types of stimulation, electric and magnetic. The microcontroller continuously cycles through the different states once a stimulation session is started. The pulse parameters control the length of time the microcontroller will linger in any of these states, when to advance to the next state, and which state to advance to. Once it has advanced to a new state, it executes a series of commands to transmit a high or low signal on the output pins connected to the output stages of the boost and buck converters. In the case of electrical stimulation, pairs of optocouplers forming the H-Bridge the output stage of the boost converter are activated and/or disabled to control the current flow direction or halt it altogether. Similarly, output pins from the microcontroller are used to enable MOSFET current drivers to couple and modulate high current signals from the buck converter through a high power MOSFET and into an external coil to produce magnetic fields. The magnetic intensity is varied by adjusting the duty cycle of the modulating signal.

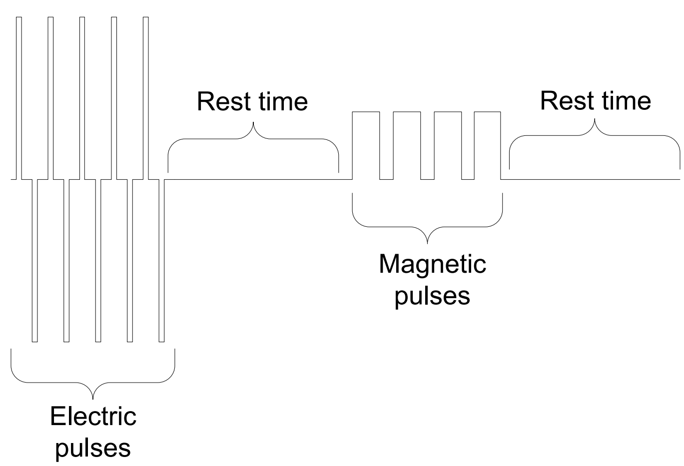


Figure 16: Illustration of EM-STIM output waveforms.

An internal watch dog timer is activated each time the system transitions into a new state. The value of this timer is set for any given state based the pulse parameters (Figure 16). For example, during a train of electrical stimulation pulses, once an ePulseOn state is reached, the watch dog timer is set a value equal to *pulse duration* and a pair of optocoupleres in the H-Bridge are enabled by setting the output pins connected to their gates to a high voltage or 3.3 V. This causes a high voltage signal with constant current of up to 150 mA to be coupled to external stimulating probes. The system will remain in that state until an interrupt is triggered by the system when time equal to *pulse duration* has passed. At this point of time, the system can advance to the next state, which can be either ePulseOff or eRestTime, depending on how many pulses were generated thus far in a train of electrical pulses. A built in counter keeps track of this count and at the end of each pulse, a comparison is made with respect to the *pulse repetition* parameter. If the two are equal, the system advances to the eRestTime state and the internal watch dog timer is set to duration equal to *rest time* otherwise the system will advance to the ePulseOff state with the watch dog timer set to *time between pulses*. In both of those cases, the microcontroller will set the output pins to a low signal until either timer expires and so on.

3.2 Control Flow

The EM-STIM firmware relies on a 16-bit timer, Timer_A, within the on board MSP430 MCU for timing control to generate pulse trains. This is done by setting the TACCR0 register, used by Timer_A, to a value that represents a time period. Timer_A will start counter that will trigger an Interrupt Service Request (ISR) once it reaches the value set in TACCR0. It is in the internal state machine built within the firmware that we use this timer in the fashion to control pulse width, duty cycle, rest periods between pulse trains specific to either magnetic or electric stimulation or both for that matter in the case of dual stimulation. As we enter a new state, we set a time counter based on the stimulation parameters controlled by the user and set the next state. The firmware contains command sequences that pertain to each specific state. For example, as we enter the ePulseOff state, an integer representing a time period of no electric pulse activity, or time between pulses, will be set, all bridges on the H-bridge will shutdown and the next state will be set. The CPU will sleep for the set time period until expiration, at which point, it will wake-up on an ISR raised by Timer_A to enter the next state and execute further commands and procedures as set in the firmware. Figures 17 and 18 show the flow chart diagrams of the state machine and Timer_A ISR function.

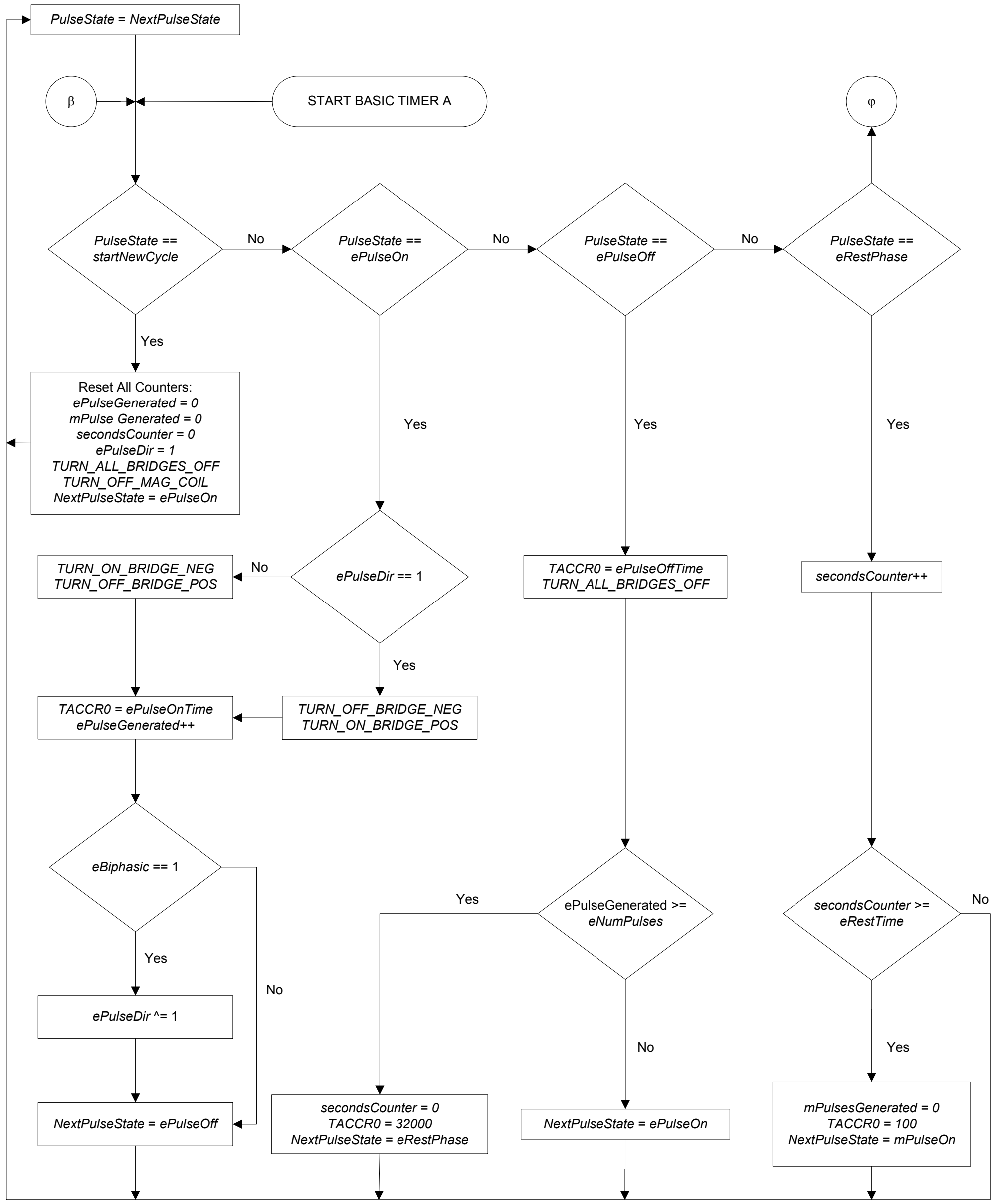


Figure 17: EM-STIM state machine control flow diagram - Part A.

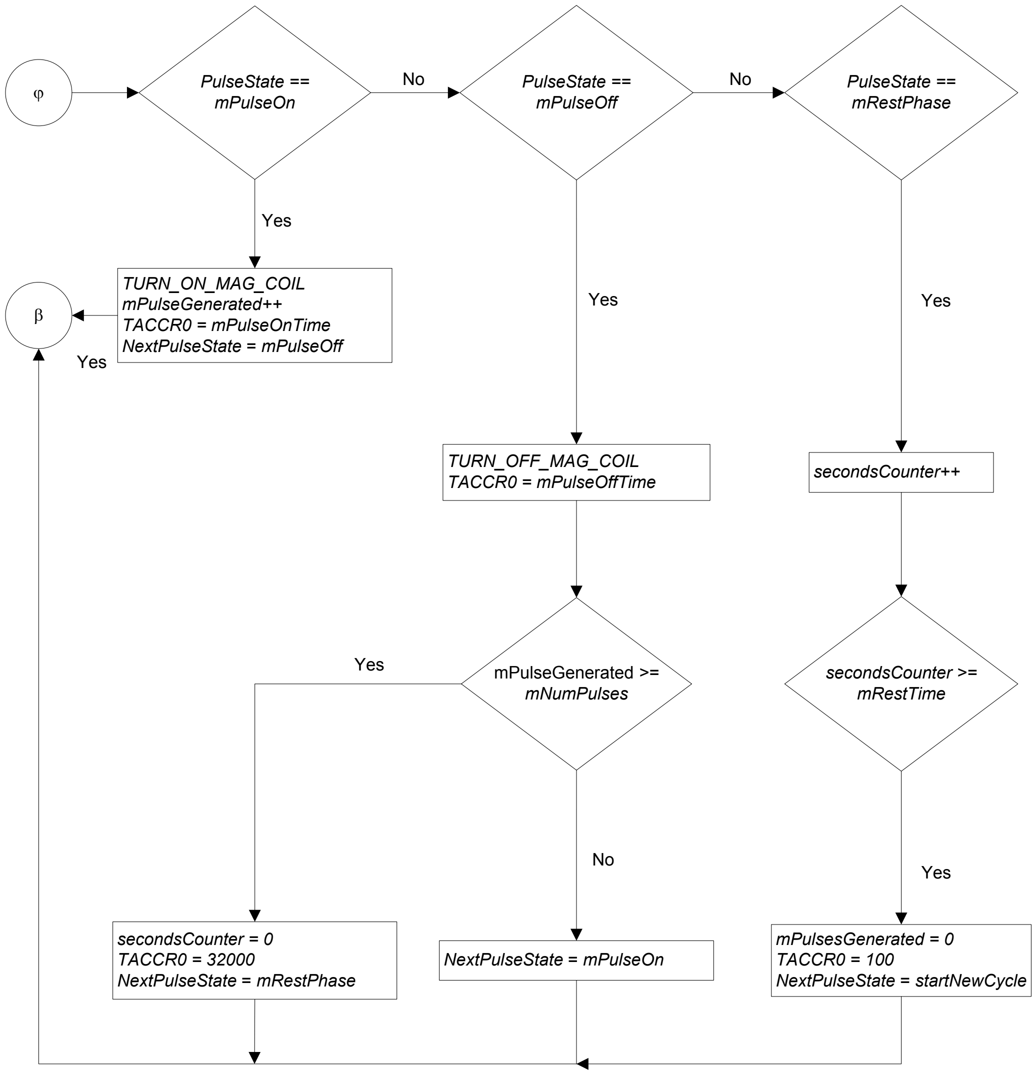


Figure 18: EM-STIM state machine control flow diagram - Part B.

CHAPTER 4

DEVICE OPERATION

4.1 Device Setup

4.1.1 Battery and Power Supply

For proper operation of EM-STIM, the attached battery pack needs to be charged for a minimum of 3 hours when using the device for the first time or when using a new battery pack. The battery is charged using a universal smart plugged in any standard 120-240 V AC power outlet with proper grounding. The device can be used while connected to external power while in operation for extended stimulation conditions. The charger comes with a temperature sensor that needs to be attached to the battery pack at all times. The smart charger uses this sensor to measure the battery temperature and ceases charging if it detects overheating conditions. Once the battery pack is charged or while connected to an AC source, the device can be powered up by pressing the attached power switch to an on status. The device will detect the incoming power and will go through initialization procedures, then wait for user input at the command line prompt.

4.1.2 Serial Interface

The EM-STIM prototype relies on an RS-232 9-pin interface for communications with a PC in order to facilitate interaction with the device operator using a command line menu-driven GUI. The device uses this connection to display current device status, modify test parameters, and to activate troubleshooting procedures built within EM-STIM based on input from the operator. A physical connection is established by connecting the EM-STIM RS-232 connector with a PCs RS-232 COM port using a standard 9-pin DB serial cable. This is to be followed by provisioning a modem connection in a terminal emulation program on the PC. The modem connection will be used to dial into EM-STIM which will display menu with command line prompt in the terminal emulation GUI application and await input from the operator though the PC keyboard. Below are the instructions on how to create a modem connection with EM-STIM using the Microsoft Windows HyperTerminal. The operator has the option of using any other commonly available terminal emulation program as the port settings below are adhered to.

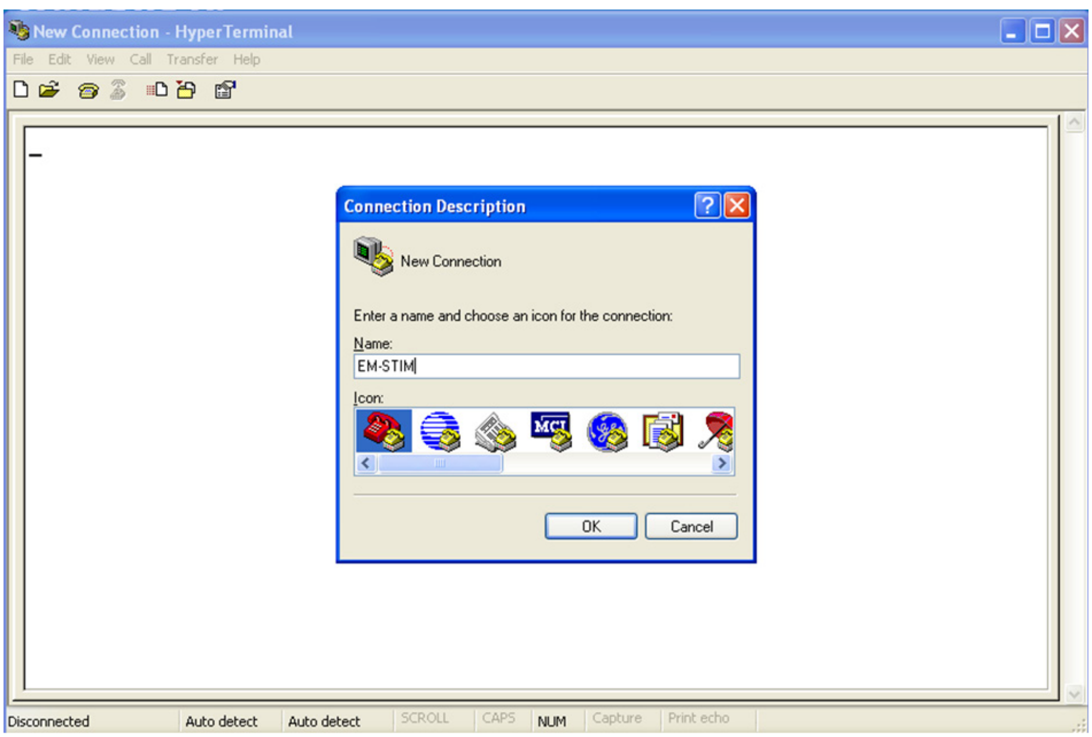


Figure 19: HyperTerminal new connection setup - connection description dialog.

1. From the **File** menu, the user should click on the **New Connection** menu option, this will display the **Connection Description** dialog in which the user should specify the connection name and click on the **OK** button (Figure 19).

Figure 20: HyperTerminal new connection setup – connect to dialog.

2. A new **Connect To** dialog will be loaded on the screen in which the user should specify the option to connect using **COM1** then click on the **OK** button (Figure 20):

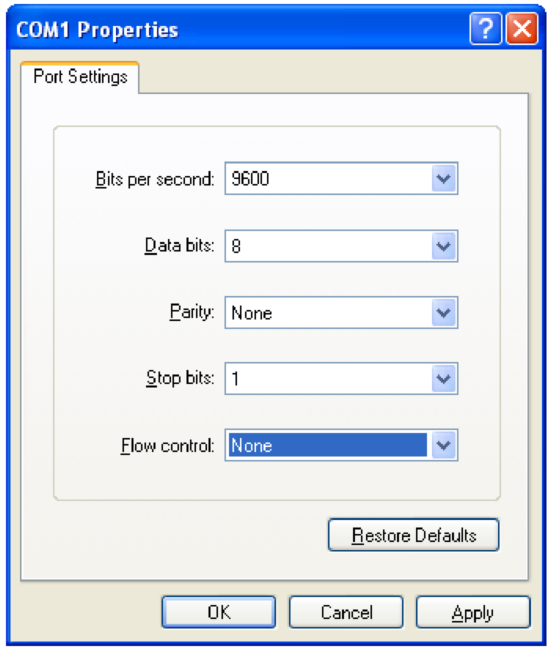


Figure 21: HyperTerminal new connection setup – COM1 Port settings dialog

3. A new **COM1 Properties** dialog will be loaded on the screen in, this should be populated according to the above figure followed by clicking on **OK** button. Once the connection has been configured, it can be saved and referred to later on to dial into EM-STIM without the need to re-enter the information (Figure 21).

4.2 Menu Navigation

Upon detecting a serial connection and on power up, EM-STIM will display the Main Menu, shown in Figure 22, with five sequential options followed by a command prompt. The user can select any of these options by entering the corresponding menu number in the command prompt and pressing the enter key.

```
----Welcome to UMKC EStim ----

*** Main Menu ***
1: Pulse Stimulation
2: Magnetic Stimulation
3: Display Settings
4: Restore default settings
5: Test (for technical personnel only)
>>
```

Figure 22: Main menu.

The first option allows the operator to modify electrical pulse stimulation parameters such as number of electrical pulses and the rest time between pulse trains. It also gives the user the option to toggle biphasic pulse generation. Figure 23 below shows the Electrical Stimulation sub-menu.

```
*** Electrical Stimulation Menu ***
Press 'q' to increase number of pulses
Press 'a' to decrease number of pulses
Press 'w' to increase rest time
Press 's' to decrease rest time
Press 't' to toggle biphasic
Press 'M' to return to main menu
```

Figure 23: Pulse stimulation menu.

The second option on the Main Menu allows the operator to modify magnetic stimulation parameters such as magnetic stimulation time and intensity levels. Figure 24 below shows the Magnetic Stimulation sub-menu and a sample command line output response.

```
*** Magnetic Stimulation Menu ***
Press 'p' to increase magnetic stimulation time
Press 'l' to decrease magnetic stimulation time
Press 'o' to increase magnetic stimulation level
Press 'k' to decrease magnetic stimulation level
Press 'M' to return to main menu
>>Magnetic ON time = 56 ms
Magnetic ON time = 62 ms
Magnetic ON time = 68 ms
Magnetic ON time = 75 ms
Magnetic ON time = 81 ms
Magnetic ON time = 87 ms
Magnetic ON time = 93 ms
Magnetic level = 15%
Magnetic level = 15%
Magnetic level = 14%
Magnetic level = 14%
Magnetic level = 13%
Magnetic level = 13%
Magnetic level = 12%
Magnetic level = 12%
Magnetic level = 11%
Magnetic level = 11%
Magnetic level = 10%
```

Figure 24: Magnetic stimulation menu and output response.

The third option on the Main Menu allows the displays the current parameter settings (Figure 25) and system status while option four allows the user to revert the settings to the default values.

```
>>3
*** System Settings ***
Pulses = 10
Electric pulse ON time = 5 ms
Electric pulse OFF time = 45 ms
Rest time = 150 sec
Biphasic ON
Battery voltage = 3.33 V
Temperature = 40 C
Probe current = 0.0 mA
Magnetic ON time = 500 ms
Magnetic level = 12%
```

Figure 25: Display settings system output

Lastly, selecting option five on the Main Menu allows the operator to enter a test menu where different parts of the output stage can be manipulated to troubleshoot any potential faults within EM-STIM. As shown Figure 26, the user has the option to turn off select branches of the H-Bridge or all together, turn of the magnetic coil, and to read the current sense amplifier output.

```
>>5
*** Test Menu ***
Press 'f' to turn off electric bridge
Press 'r' to turn on RIGHT bridge branch
Press 'l' to turn on LEFT bridge branch
Press 'm' to turn on magnetic coil
Press 'c' to read the output of the current sense amplifier
Press 'M' to return to main menu
Probe current = 0.6 mA
```

Figure 26: Test menu.

CHAPTER 5

MEASUREMENTS AND DATA

5.1 Application Measurements

The electric and magnetic outputs of the EM-Stim were measured individually to gauge their performance against the original technical requirements. A gaussmeter (Model 421 Gaussmeter, LakeShore, Westerville, OH) was used to measure the strength of the magnetic field output from the coil at different distances from the center of the coil and at different excitation voltages. The coil was built to have 100 turns and using enameled copper wire #30 AWG. The probe was placed at distances of 5 mm, 14 mm and 20 mm from the coil. Figure 27 depicts the intensity of the measured magnetic field. The intensity was varied by changing the duty cycle of the pulse-width modulated signal that drives the power MOSFET resulting in a linear relationship between the field intensity and the duty cycle. As shown in diagram 27, the intensity varies linearly with the duty cycle and is inversely proportional to the distance from the coil. Operators of EM-STIM have the ability to set the duty cycle from within the command menu using the serial interface. The maximum measured magnetic field is 1.6 mT for an excitation voltage of 1.8 V (output voltage of the LTM4600 buck converter). Higher magnetic fields can be obtained by setting higher excitation voltages but that leads to higher currents through the coil, raising its temperature to levels that can be considered unsafe. Thus, the DC current through the coil is capped at around 0.7 A with and excitation voltage of 1.8 V. The rise in temperature can be controlled by using a thicker coil wire and by cooling the coil [28].

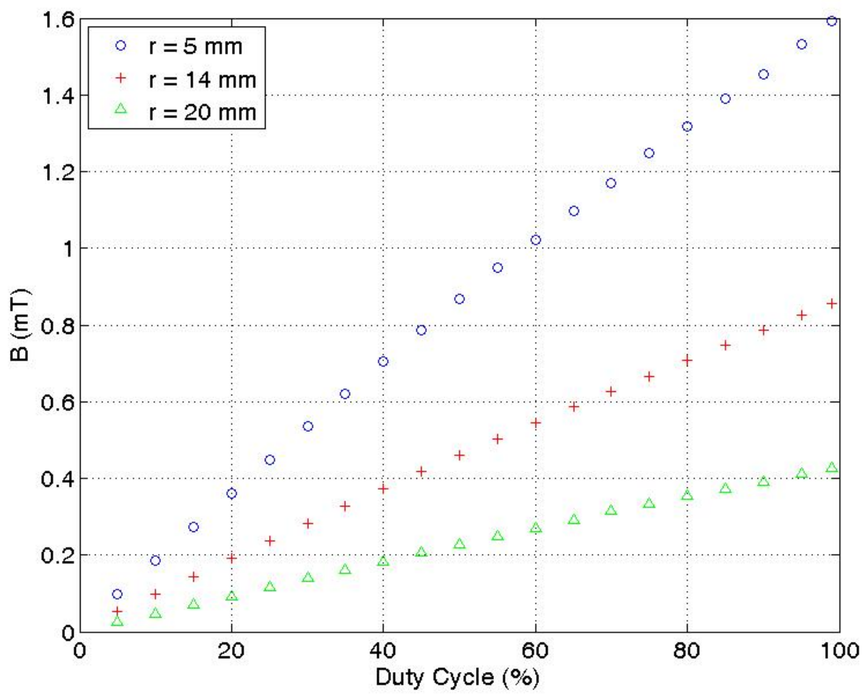


Figure 27: Measured intensity of the magnetic field produced by the coil.

The electric stimulation output was tested by submerging two carbon electrodes in a cell culture media. The type of media used was Dubelcco's modified Eagle media with high glucose (Mediatech) supplemented with 10% fetal bovine serum (Fisher), 10µg/ml penicillin and 100µg/ml streptomycin (Invitrogen). A multiwell C-Dish (IonOptix, Milton, MA) culture dish assembly was employed for this setup: This C-Dish provides six wells where different cell cultures can be placed and stimulated. The C-Dish electrodes are made of pure carbon which helps reduce build-up of toxic waste products. 10 ml of culture media was placed in each well and the EM-Stim was set to generate a train of 10 pulses on its electric stimulation output. Each pulse had duration of 5 ms and a repetition period of 50ms. The output of the current sense amplifier (LTC6102) was monitored with an oscilloscope (Tektronix MSO 3032). Figure 28 shows the voltage output of the current sense amplifier. From this voltage waveform, the peak current through the culture media can be calculated.

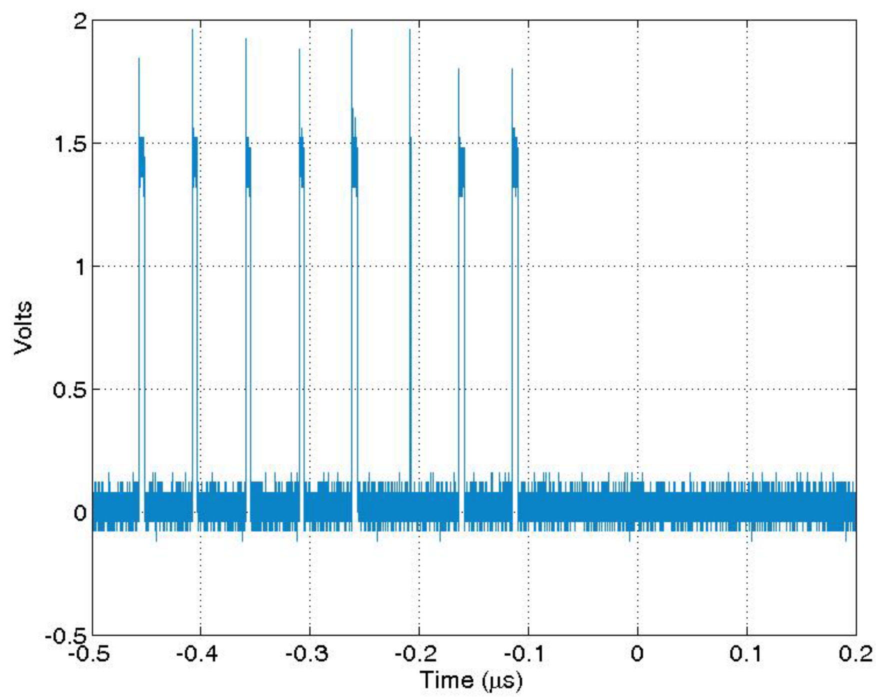


Figure 28: Measured voltage at the output of the current sense amplifier.

The voltage output of the LTC6102 current sense amplifier is given by:

$$V_{out} = V_{sense}\left(\frac{R_{out}}{R_{in}}\right) \tag{2}$$

where $V_{sense} = I_{sense} \times R_{sense}$. Thus,

$$I_{sense} = \frac{V_{out}}{R_{sense}\left(\frac{R_{out}}{R_{in}}\right)} \tag{3}$$

Replacing the values of Rsense = 0.05 Ω, Rout = 10 kΩ and Rin = 47 Ω, we obtain a peak current of 140 mA which is in agreement with the design specifications.

5.2 Biological Test Data

The myogenic cell line C2C12 was used to test the effects of the electric and magnetic pulses. The C2C12 cells were plated and grown to confluence in Dulbecco's Modified Eagle Medium (Mediatech) containing 10% fetal bovine serum (Fisher), 10µg/ml penicillin and 100µg/ml streptomycin (Invitrogen). Cell cultures were maintained at 37° C and under a continuous 5% CO_2 stream appeared confluent after approximately 24 h of incubation. The culture medium was then removed and replaced with DMEM containing 2% horse serum and incubated for 48 h while subjected to Pulsed Electro-Magnetic stimulation. EM-Stim was set to generate a train of 10 pulses on its electric stimulation output. Each pulse had duration of 5 ms and a repetition period of 50ms.Figure 29 (a) shows untreated cultured C2C12 myoblasts. On the first run, we stimulated C2C12 myoblasts and after 24 hours, shown in figure 29 (b), there was a significant increase in myoblast proliferation observed under the microscope that was further substantiated by an increase in protein synthesis of approximately 4%. In a second series of experiments, myoblasts were stimulated for a period of 48 hours, shown in figure 29 (c), under the same conditions where we observed an enhancement of the myogenic program in treated cells compared to control cells under non-stimulated conditions, where, myoblasts started to fuse and form myotubes (muscle cells) quicker and appeared denser under in microscopic observation.

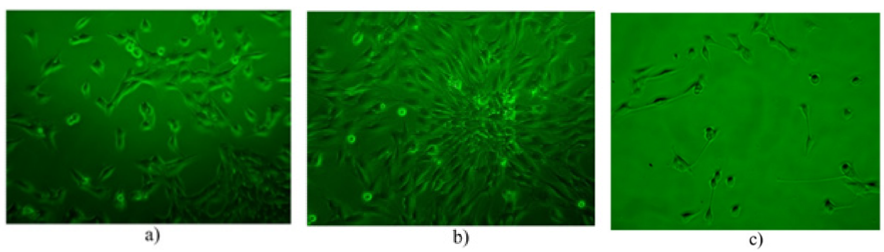


Figure 29: Microscopic images of C2C12 cells before (a), during (b) and at end of stimulation period (c).

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

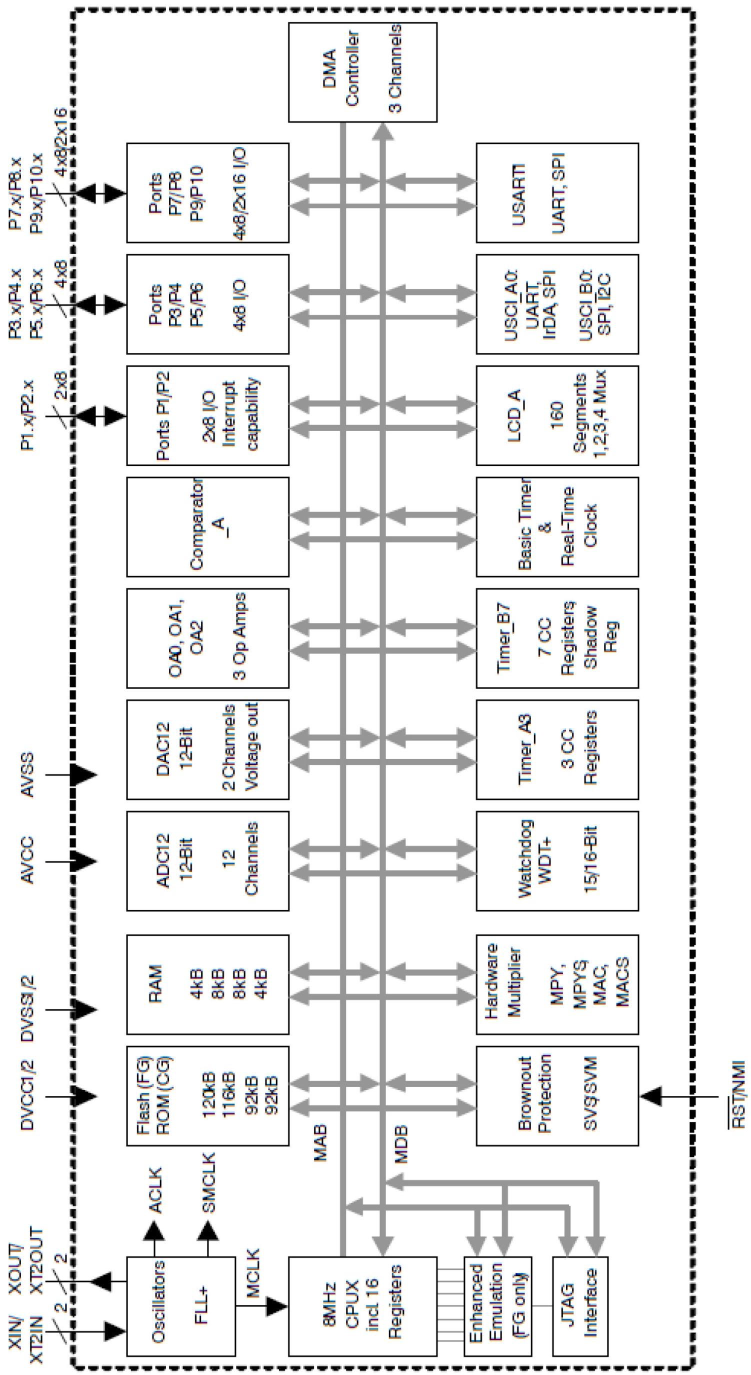
EM-STIM provides a unique tool to study the effects of concurrent electrical and magnetic stimulation on muscle and bone cells. The design is compact yet provides a highly configurable research tool that can be used to generate an unlimited combination of waveform patterns involving pulsed electric and magnetic pulse trains with varying degrees of intensities and frequencies. Initial studies on C2C12 cell cultures showed an increase of 4% in cell mass and enhancement of the myogenic program which together substantiates that under in vitro conditions the EM-Stim is able to promote proliferation and differentiation of C2C12 myoblast cells.

6.2 Future Work

Planned enhancements to the EM-STIM device include integrating additional stimulation channels into the circuit board to allow for electrical and magnetic stimulation of multiple cell flasks in a simultaneous fashion. This would involve adding H-bridges and high power MOSFETs. Other enhancements include building a complete feedback loop to control the boost voltage converter circuit based on variable impendence conditions in the cell growth media or in vivo studies involving treated flesh. Adding wireless data connectivity may allow for real time transmit of operational status, as well as, enable the control of EM-STIM remotely while conducting stimulation studies.

Our future goals include measuring the effects of stimulation over extended periods of time. In addition to that, we seek to study the effects of stimulation on bone cells by including osteoblasts along side myoblasts in multi-welled flasks and attempt to assess their interaction under combined electric and magnetic stimulation conditions and perhaps move to in vivo experimentation in primary muscle cells from animals and humans.

Appendix A MSP430FG439 Functional Diagram



Appendix B Firmware C Code

```
//*****************************************************************
***
// UMKC EM-Stim Device - Electric and Magnetic Cell Stimulator
//*****************************************************************
***
#include <Board.h>
#include <cstring>
#include <ctype.h>
#include <LCD.h>

// Function Declarations
void initBasicTimer(void);
void initPortPins(void);
void initDAC12(void);
void displayVoltage(int, unsigned int, double);
void initUART(void);
void initTimerB(void);
void initTimerA(void);
void initADC12(void);
void initVariables(void);

void TXString(char*, int);
```

```
void TXCharacter(char);
void TXeConfig(void);
void TXmConfig(void);
void TXVoltage(unsigned int, double);
void TXCurrent(unsigned int);
void processInput(void);
void displayMenu(TMenu);
void displayMenuOption(TMenu, int);
//void displaySampleAvg(volatile unsigned int *);
void TXValue(char *, unsigned int, char *);
void ScanAnalogInputs(void);

enum TState {TMainMenu, TPulseMenu, TMagMenu, TSettingsMenu, TTestMenu}
State;
enum TPulseState {startNewCycle, ePulseOn, ePulseOff, eRestPhase, mPulseOn,
mPulseOff, mRestPhase} PulseState, NextPulseState;
unsigned int ePulseOnTime, ePulseOffTime, mPulseOnTime, mPulseOffTime,
eNumPulses, mNumPulses;
unsigned int ePulsesGenerated, secondsCounter, mPulsesGenerated, eRestTime,
mRestTime;
unsigned char eBiphasic, ePulseDir;
```

```
char splash[] = {"\r\n----Welcome to UMKC EM-Stim ----\r\n\r\n"};
char msg1[]   = {"\r\nUnder construction...\r\n>>"};
char msg2[]   = {"Pulses = "};
char msg3[]   = {"Rest time = "};
char msg4[]   = {"Magnetic level = "};
char msg5[]   = {"Magnetic ON time = "};
char msg6[]   = {"%\r\n"};
char msg7[]   = {"\r\n"};
char msg8[]   = {"Biphasic ON\r\n"};
char msg9[]   = {"Biphasic OFF\r\n"};
char msg10[]  = {" ms\r\n"};
char msg11[]  = {"\r\n*** System Settings ***\r\n"};
char msg12[]  = {" sec\r\n"};
char msg13[]  = {"Electric pulse ON time = "};
char msg14[]  = {"Electric pulse OFF time = "};
char msg15[]  = {"..... default settings restored\r\n"};
char msg16[]  = {"Battery voltage = "};
char msg17[]  = {" V\r\n"};
char msg18[]  = {"Temperature = "};
char msg19[]  = {" C\r\n"};
char msg20[]  = {"Probe current = "};
char msg21[]  = {" mA\r\n"};
```

```
static const char *Menu1[] = { "\r\n*** Main Menu ***\r\n",
                        "1: Pulse Stimulation\r\n",
                        "2: Magnetic Stimulation\r\n",
                                "3: Display Settings\r\n",
                        "4: Restore default settings\r\n",
                        "5: Test (for technical personnel only)\r\n",
                                ">>"
                        };

static const char *Menu2[] = { "\r\n*** Electrical Stimulation Menu ***\r\n",
                        "Press 'q' to increase number of pulses\r\n",
                        "Press 'a' to decrease number of pulses\r\n",
                                "Press 'w' to increase rest time\r\n",
                                "Press 's' to decrease rest time\r\n",
                        "Press 't' to toggle biphasic\r\n",
                                "Press 'M' to return to main menu\r\n",
                        ">>"
                        };

static const char *Menu3[] = { "\r\n*** Magnetic Stimulation Menu ***\r\n",
                        "Press 'p' to increase magnetic stimulation time\r\n",
                        "Press 'l' to decrease magnetic stimulation time\r\n",
                                "Press 'o' to increase magnetic stimulation level\r\n",
```

```
                "Press 'k' to decrease magnetic stimulation level\r\n",
                "Press 'M' to return to main menu\r\n",
        ">>"
        };

static const char *Menu4[] = { "\r\n*** Settings Menu ***\r\n",
        "Press 'e' to display electric pulse settings\r\n",
        "Press 'm' to display magnetic pulse settings\r\n",
                "Press 'M' to return to main menu\r\n",
        ">>"
        };

static const char *Menu5[] = { "\r\n*** Test Menu ***\r\n",
        "Press 'f' to turn off electric bridge\r\n",
        "Press 'r' to turn on RIGHT bridge branch\r\n",
                "Press 'l' to turn on LEFT bridge branch\r\n",
        "Press 'm' to turn on magnetic coil\r\n",
        "Press 'c' to read the output of the current sense amplifier\r\n",
        "Press 'M' to return to main menu\r\n",
        ">>"
        };
```

```
TMenu mainMenu={Menu1, 7}, pulseMenu={Menu2, 8}, magMenu={Menu3, 7},
setMenu={Menu4, 4}, tstMenu={Menu5, 7};

unsigned int SampleIndex, BattVolt, ADCOut, Temperature, CurrentSense;
char blink_battery, PendingFlag;

void main(void)
{
 unsigned int i;

 WDTCTL = WDTPW + WDTHOLD;              // Stop WDT
 FLL_CTL0 |= XCAP18PF;                  // Set load cap for 32k xtal
 do {                                   // Wait for xtal to stabilize
  IFG1 &= ~OFIFG;                       // Clear OSCFault flag
  for (i = 0x47FF; i > 0; i--);         // Time for flag to set
 } while ((IFG1 & OFIFG));              // OSCFault flag still set?

 initPortPins();                        // Initialize port pins
 initBasicTimer();                      // Initialize basic timer
 initLCD_A();                           // Initialize LCD_A
 initDAC12();
 initUART();
 initTimerB();
```

```
  initADC12();
  initVariables();
  initTimerA();

  dispAllLCDSegs();
  for(i=0; i<60000; i++);
  clrLCD();

  TXString((char *) splash, sizeof splash); //transmits welcome message
  displayMenu(mainMenu);

  for(;;)
  {
//    ScanAnalogInputs();
    _BIS_SR(LPM3_bits + GIE);              // LPM3, enable interrupts
  }
}

//****** This function process characters received through the UART and displays
menus ******
void processInput(void) {
switch(State)
{
```

```
 case TMainMenu:                              //main menu is active
  if (UCA0RXBUF=='1') {                        //if '1' is pressed, goes to Pulse Menu
        UCA0TXBUF = UCA0RXBUF;                 //echo character
        displayMenu(pulseMenu);
        State = TPulseMenu;
  }
  else if (UCA0RXBUF=='2') {                   //if '2' is pressed, goes to Magnetic
Menu
        UCA0TXBUF = UCA0RXBUF;                 //echo character
        displayMenu(magMenu);
        State = TMagMenu;
  }
  else if (UCA0RXBUF=='3') {                   //if '3' is pressed, goes to Test Menu
        UCA0TXBUF = UCA0RXBUF;                 //echo character
        TXString((char *) msg11, strlen(msg11)); //transmits welcome message
        TXeConfig();
        TXmConfig();
  }
  else if (UCA0RXBUF=='4') {                   //if '4' is pressed, restore default settings
        initVariables();
        TXString((char *) msg15, strlen(msg15));
  }
  else if (UCA0RXBUF=='5') {                   //if '4' is pressed, restore default settings
```

```
        UCA0TXBUF = UCA0RXBUF;
        TACTL &= ~0x0030;                //stops timerA to avoid conflicts
        TURN_ALL_BRIDGES_OFF;            //turn off bridge and coil
        TURN_OFF_MAG_COIL;
        displayMenu(tstMenu);
        State = TTestMenu;
    }
    else
     displayMenu(mainMenu);
    break;

  case TPulseMenu:
    if (UCA0RXBUF=='M') {                //returns to main menu
      displayMenu(mainMenu);
      State = TMainMenu;
    } else if (UCA0RXBUF=='q') {         //increase number of electrical pulses
     if (eNumPulses < MAX_PULSES_PER_BURST)
      eNumPulses++;
     TXValue(msg2, eNumPulses, msg7);
    } else if (UCA0RXBUF=='a') {         //decrease number of electrical pulses
     if (eNumPulses > 1)
      eNumPulses--;
     TXValue(msg2, eNumPulses, msg7);
```

```
  } else if (UCA0RXBUF=='w') {                    //increases rest time
    if (eRestTime < MAX_REST_TIME) {
      eRestTime++;
      mRestTime++;
    }
    TXValue(msg3, eRestTime, msg12);              //displays the rest time
  } else if (UCA0RXBUF=='s') {                    //decreases rest time
    if (eRestTime > MIN_REST_TIME) {
      eRestTime--;
      mRestTime--;
    }
    TXValue(msg3, eRestTime, msg12);              //displays the rest time
  } else if (UCA0RXBUF=='t') {                    //toggles biphasic pulse mode
generation
      eBiphasic ^= 1;
      if(eBiphasic)
        TXString((char *) msg8, sizeof msg8);
      else
        TXString((char *) msg9, sizeof msg9);
  } else
  displayMenu(pulseMenu);


  break;
```

```
case TMagMenu:

   if (UCA0RXBUF=='M') {                          //returns to main menu
      displayMenu(mainMenu);
      State = TMainMenu;
   } else if (UCA0RXBUF=='p') {                   //increases mag. stimulation time
     if (mNumPulses < MAX_MAG_PULSES)
       mNumPulses++;
     TXValue(msg5, mNumPulses*(mPulseOnTime + mPulseOffTime)/32, msg10);
//transmits the total mag. ON time
   } else if (UCA0RXBUF=='l') {                   //decreases mag. stimulation time
     if (mNumPulses > 1)
       mNumPulses--;
     TXValue(msg5, mNumPulses*(mPulseOnTime + mPulseOffTime)/32, msg10);
//transmits the total mag. ON time
   } else if (UCA0RXBUF=='o') {                   //increases mag. stimulation level
     if (mPulseOffTime>1) {
        mPulseOnTime++;
        mPulseOffTime--;
     }
     TXValue(msg4, 100*mPulseOnTime/(mPulseOnTime + mPulseOffTime),
msg6); //displays the on time %
```

```
  } else if (UCA0RXBUF=='k') {                    //decreases mag. stimulation level
    if (mPulseOnTime>1) {
      mPulseOnTime--;
      mPulseOffTime++;
    }
   TXValue(msg4, 100*mPulseOnTime/(mPulseOnTime + mPulseOffTime),
msg6);
  } else
  displayMenu(magMenu);

  break;

case TTestMenu:
  if (UCA0RXBUF=='M') {                          //returns to main menu
    displayMenu(mainMenu);
    TACTL |= 0x0030;                            //resume timer
    State = TMainMenu;
  } else if (UCA0RXBUF=='f') {                   //turns off bridge and coil
     TURN_ALL_BRIDGES_OFF;
     TURN_OFF_MAG_COIL;
  } else if (UCA0RXBUF=='r') {
     TURN_OFF_BRIDGE_NEG;
     TURN_ON_BRIDGE_POS;
```

```
  } else if (UCA0RXBUF=='l') {
      TURN_OFF_BRIDGE_POS;
      TURN_ON_BRIDGE_NEG;
  } else if (UCA0RXBUF=='m') {
      TURN_ON_MAG_COIL;
  }  else if (UCA0RXBUF=='c') {
      TXCurrent(CurrentSense);
  } else
  displayMenu(tstMenu);


}


}


//*********Transmits a menu to the terminal ************
void displayMenu(TMenu menu) {
 unsigned int i;
 for (i=0; i<menu.MenuLength; i++)
    TXString((char *)menu.Menu[i], strlen(menu.Menu[i]));
}


//***** Transmits the value of an integer variable with leading and trailing texts
*****
```

```
void TXValue(char *msg_lead, unsigned int value, char *msg_trail) {

  TXString((char *)msg_lead, strlen(msg_lead));

  if (value >= 1000)
    TXCharacter('0' + ((value/1000)%10));
  if (value >= 100)
    TXCharacter('0' + ((value/100)%10));
  if (value >= 10)
    TXCharacter('0' + ((value/10)%10));
  TXCharacter('0' + (value%10));

  TXString((char *)msg_trail, strlen(msg_trail));
}

//********* ADC12 interrupt service *************
#pragma vector = ADC12_VECTOR
__interrupt void ADC12_ISR(void)
{
//   PendingFlag = 0;
//   ADCOut = ADC12MEM0;

   BattVolt     = ADC12MEM0;
```

```
   Temperature  = ADC12MEM1;
   CurrentSense = ADC12MEM2;


   ADC12IV   = 0;
   ADC12IFG  = 0;
   ADC12CTL0 &= ~ADC12SC;                  // reset SC signal


   __bic_SR_register_on_exit(LPM0_bits);     // Exit on LPM0


}


//********* Timer_B interrupt service routine ********
#pragma vector=TIMERB0_VECTOR
__interrupt void Timer_B (void)
{


    ScanAnalogInputs();
    __bic_SR_register_on_exit(LPM3_bits);     // Exit LPM3


}


//*********** Basic Timer Interrupt Service Routine *************
#pragma vector=BASICTIMER_VECTOR
```

```
__interrupt void basic_timer_ISR(void)
{
 LPM3_EXIT;
}

#pragma vector = TIMERA0_VECTOR
__interrupt void TimerA(void)
{
//  TACTL &= ~0x0030;                       //MC=00 -- stops timer to avoid races
  if (PulseState == startNewCycle) {       //new pulse generating cycle starts
     ePulsesGenerated = 0;
     mPulsesGenerated = 0;
     secondsCounter   = 0;
     ePulseDir        = 1;
     TURN_ALL_BRIDGES_OFF;
     TURN_OFF_MAG_COIL;
     NextPulseState = ePulseOn;            //sets next state
  } else
  if (PulseState == ePulseOn) {            //this state turns on the electrical stimulation
bridge
     if (ePulseDir == 1) {                 //implements biphasic polarity (ePulseDir=1 ->
forward current, ePulseDir=0 -> backward current)//
        TURN_OFF_BRIDGE_NEG;
```

```
      TURN_ON_BRIDGE_POS;
    } else {
      TURN_OFF_BRIDGE_POS;
      TURN_ON_BRIDGE_NEG;
    }
    TACCR0  = ePulseOnTime;              //sets the ON time
    ePulsesGenerated++;
    if (eBiphasic == 1)
      ePulseDir ^= 1;                   //toggle the pulse direction if biphasic mode is set
    NextPulseState = ePulseOff;

    //read current sense amplifier
//    ADC12MCTL0 = INCH_4;               // select A4 --> connected to LTC6102
//    ADC12CTL0 |= ADC12SC;              // Start sampling/conversion
//    CurrentSensePending = 1;           // signal for the ADC12 interruption routine

  } else
  if (PulseState == ePulseOff) {         //this state implements the off pulse time
    TURN_ALL_BRIDGES_OFF;
    TACCR0  = ePulseOffTime;             //sets the OFF time
    if (ePulsesGenerated >= eNumPulses) {
//if enough pulses have been generated switch to rest phase
      secondsCounter  = 0;
```

```
      TACCR0 = 32000;                    //will interrupt every second from now on
(RestPhase)
      NextPulseState = eRestPhase;
    } else
      NextPulseState = ePulseOn;
  } else
  if (PulseState == eRestPhase) {
    secondsCounter++;                    //count the number of seconds in RestPhase
    if (secondsCounter >= eRestTime) {
     mPulsesGenerated = 0;
     NextPulseState = mPulseOn;          //startNewCycle;
     TACCR0  = 100;                      //sets a default interruption time
    }
  } else
  if (PulseState == mPulseOn) {          //generates a magnetic pulse
    TURN_ON_MAG_COIL;
    mPulsesGenerated++;
    TACCR0  = mPulseOnTime;              //sets the ON time of magnetic pulse
    NextPulseState = mPulseOff;
  } else
  if (PulseState == mPulseOff) {
    TURN_OFF_MAG_COIL;
    TACCR0  = mPulseOffTime;             //sets the OFF time of magnetic pulse
```

```
    if (mPulsesGenerated >= mNumPulses) {
      secondsCounter = 0;
      TACCR0 = 32000;
      NextPulseState = mRestPhase;
    } else
      NextPulseState = mPulseOn;

  } else
  if (PulseState == mRestPhase) {
    secondsCounter++;                    //count the number of seconds in RestPhase
    if (secondsCounter >= mRestTime) {           //300 sec = 5 min
      NextPulseState = startNewCycle;
      TACCR0  = 100;                     //sets a default interruption time
    }
  }

  PulseState = NextPulseState;
//  TACTL |= 0x0030;                        //MC=00 -- resumes timer

  __bic_SR_register_on_exit(LPM0_bits);
// Exit in low power mode LPM3, interrupts enabled
}
```

```
//******** PORT 1 interruption service routine ************
#pragma vector=  PORT1_VECTOR
__interrupt void P1_ISR (void)
{
  unsigned int i;
   i=15;                           //Delay, button debounce
   do (i--);
   while (i !=0);

  if (P1IFG == 0x01 && DAC12_0DAT <= (4095-16)){
    DAC12_0DAT += 16;
    dispArrow(LCD_ARROW_RIGHT);
  }

  if (P1IFG == 0x02 && DAC12_0DAT >= 16) {
    DAC12_0DAT -= 16;
    dispArrow(LCD_ARROW_LEFT);
  }

  P1IFG = 0x00;                        // clear interrupt flags
  P5OUT ^= 0x02;
  __bic_SR_register_on_exit(LPM3_bits);     // Exit LPM3
}
```

```
//******** UART interruption service routine ************
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCIA0RX_ISR (void)
{
  while(!(IFG2&UCA0TXIFG));
  processInput();
  __bic_SR_register_on_exit(LPM3_bits);     // Exit LPM3


}


//************* Initialize basic timer **************
void initBasicTimer(void)
{
  // Basic timer setup
  // Set ticker to 32768/(256*128)
  // Enable BT interrupt
  BTCTL = BT_fCLK2_DIV128 | BT_fCLK2_ACLK_DIV256;
  IE2 |= BTIE;
}


//************ Initialize DAC *****************
void initDAC12(void)
```

```
{
  DAC12_0CTL = DAC12IR + DAC12AMP_5 + DAC12ENC; // Int ref gain 1
  DAC12_0DAT = 2047;
}

void initTimerB(void)
{
 TBCCTL0 = CCIE;                          // TACCR0 interrupt enabled
 TBCCR0  = 16000;                         // interrupt every half second
 TBCTL   = TBSSEL_1 + MC_1;               //clk=ACLK, count up to TACCR0
}

void initTimerA(void) {

 TACCTL0 = CCIE;                          // TACCR0 interrupt enabled
 TACCR0  = 400;                           //default counter value
 TACTL   = TASSEL_1 + MC_1;               //clk=ACLK, count up to TACCR0

}

//************ Displays a voltage on the LCD display ****************
// Displays a 12-bit integer with two-digit precision starting at 'position' on LCD
display
```

```
void displayVoltage(int position, unsigned int volt_int, double Vref)
{
   double volt;
   unsigned int temp;

   volt = (volt_int*Vref*100)/4096;
   temp = (unsigned int)(volt);
   dispChar(position, temp%10);
   dispChar(position+1, (temp/10)%10);
   if (position < 3)
     dispSpecialChar(position+2);
   dispChar(position+2, (temp/100)%10);
   if (temp >= 1000)
     dispChar(position+3, (temp/1000)%10);
}

/*void displaySampleAvg(volatile unsigned int *results)  //obsolete...
{
 unsigned int temp;

 temp = results[0];
 dispChar(4, temp%10);
 dispChar(5, (temp/10)%10);
```

```
 dispChar(6, (temp/100)%10);

 dispChar(7, (temp/1000)%10);


}

*/

//************** Initialize UART **************

void initUART(void)

{

 P2SEL |= 0x030;                         // P2.5,4 = USCI_A0 RXD/TXD

 UCA0CTL1 |= UCSSEL_1;                   // CLK = ACLK

 UCA0BR0 = 0x03;                         // 32k/9600 - 3.41

 UCA0BR1 = 0x00;                         //

 UCA0MCTL = 0x06;                        // Modulation

 UCA0CTL1 &= ~UCSWRST;                   // **Initialize USCI state machine**

 IE2 |= UCA0RXIE;                        // Enable USCI_A0 RX interrupt

}


//*********** Initialize port pins ****************

void initPortPins(void)

{

  P2DIR = PIN2+PIN1;                     // Set P2.2,1 as outputs

  P5DIR = PIN1;                          // Set P5.1 as output

  P2OUT = PIN1;                          // Set P2.1 to 1
```

```
  P5DIR |= 0x02;
  P1IES |= 0x03;
  P1IE  |= 0x03;

  P6DIR = 0x07;                    // Set ports 6.0, 6.1, 6.2 as output pins
  P6OUT = 0x00;                    // write a zero to P6
  P6SEL = 0xF0;                    // P6.7 and P6.5 ADC option select
 // P6.7 (A7) will sample battery voltage, P6.5 (A5) will sample current sense amp.
  TURN_ALL_BRIDGES_OFF;            // Turn All Bridges OFF
  TURN_OFF_MAG_COIL;               // makes sure magnetic coil is off

}

//************** Initialize ADC12 ***************
void initADC12(void)
{
  ADC12CTL1 = SHP + ADC12DIV_3 + CONSEQ_1;    // SMCLK selected
  ADC12CTL0 = SHT0_1 + REF2_5V + REFON + ADC12ON; // Config ADC12

  ADC12MCTL0 = INCH_7;             // Channel A7 - battery voltage
  ADC12MCTL1 = INCH_10;            // Channel A10 - temperature
  ADC12MCTL2 = INCH_5 + EOS;       // Channel A5 - current sense
```

```
  ADC12IE   |= 0x07;                        // Enable ADC12IFG.3
  ADC12CTL0 |= ENC;                         // Config ADC12
  ADC12IFG  = 0;
}

//*** Initialize public variables ***
void initVariables(void) {

  State = TMainMenu;
  SampleIndex = 0;
  blink_battery = 1;

  //variables for the pulse generation machine
  PulseState = startNewCycle;    //default state at startup
  NextPulseState = PulseState;
  ePulseOnTime=160;              //on time for electrical pulses = 1/32*160 = 5ms
  ePulseOffTime=1440;            //off time for electrical pulses = 1/32*1440 = 45ms
  mPulseOnTime=25;               //on time for magnetic pulses --> 25/(25+175) = 12.5%
duty cycle
  mPulseOffTime=175;             //off time for magnetic pulses
  eNumPulses=10;                 //number of electrical pulses to be generated (~500ms
total)
```

```
  mNumPulses=80;                  //number of magnetic pulses to be generated (~6.25
ms/pulse - 500ms total)
  ePulsesGenerated=0;             //number of electric pulses generated so far
  mPulsesGenerated=0;             //number of magnetic pulses generated so far
  secondsCounter=0;               //will count the number of seconds in the RestPhase
  eBiphasic=1;                    //biphasic pulse mode generation on
  ePulseDir=1;                    //pulse direction 1=forward, 0=backward
  eRestTime=150;                  //rest time (in sec) after electric pulse EM-Stimulation
  mRestTime=150;                  //rest time (in sec) after magnetic pulse EM-Stimulation

}

//*** Transmits a string of characters over the serial port ***
void TXString( char* string, int length )
{
  int pointer;
  for( pointer = 0; pointer < length; pointer++)
  {
    UCA0TXBUF = string[pointer];
    while (!(IFG2&UCA0TXIFG));                // USCI_A0 TX buffer ready?
  }
}
```

```
//*** Transmits a single character over the serial port ***
void TXCharacter(char character)
{
  UCA0TXBUF = character;
  while (!(IFG2&UCA0TXIFG));                 // USCI_A0 TX buffer ready?
}

//*** Transmits the configuration of the electrical stimulation over the serial port ***
void TXeConfig(void) {
  unsigned int TempC;

   TXValue(msg2, eNumPulses, msg7);
   TXValue(msg13, ePulseOnTime/32, msg10);
   TXValue(msg14, ePulseOffTime/32, msg10);

   TXValue(msg3, eRestTime, msg12);              //transmits the rest time
   if(eBiphasic)
     TXString((char *) msg8, sizeof msg8);
   else
     TXString((char *) msg9, sizeof msg9);

   TXString((char *) msg16, strlen(msg16));
   TXVoltage(BattVolt, 15.0);
```

```
  TXString((char *) msg17, strlen(msg17));

  TempC = ((Temperature/4 - 673) * 4230) / 1024;

  TXValue(msg18, TempC, msg19);

  TXCurrent(CurrentSense);

}

void TXCurrent(unsigned int current) {
 unsigned int I_mA;
 double tmp;

 tmp = (CurrentSense*33*20*1000.0/(4096*47.0));
 I_mA = (unsigned int) tmp;

 TXString(msg20, strlen(msg20));

 if (I_mA > 100)
   TXCharacter('0' + (I_mA/100)%10);
 TXCharacter('0' + (I_mA/10)%10);
 TXCharacter('.');
```

```
  TXCharacter('0' + I_mA%10);

  TXString(msg21, strlen(msg21));

}

//*** Transmits the configuration of the magnetic stimulation over the serial port ***
void TXmConfig(void) {
   TXValue(msg5, mNumPulses*(mPulseOnTime + mPulseOffTime)/32, msg10);
//transmits the total mag. ON time
   TXValue(msg4, 100*mPulseOnTime/(mPulseOnTime + mPulseOffTime), msg6);
//displays the on time %

}

//*** Receives a 12-bit integer representing a voltage and converts its corresponding
text representation with two decimal digits
void TXVoltage(unsigned int volt_int, double Vref) {

   double volt;
   unsigned int temp;

   volt = (volt_int*Vref*100)/4096;
```

```
   temp = (unsigned int)(volt);

   if (temp >= 1000)
//if the voltage is larger than 10 V we need to add an extra digit to the left
     TXCharacter('0' + (temp/1000)%10);
   TXCharacter('0' + (temp/100)%10);
   TXCharacter('.');
   TXCharacter('0' + (temp/10)%10);
   TXCharacter('0' + temp%10);

}

void ScanAnalogInputs(void) {
  int i;

    clrLCD();
    displayVoltage(0, DAC12_0DAT, 2.5);
//displays on LCD the output voltage of DAC0

    //start conversion on channels A7, A10, and A5
    ADC12CTL0 |= ADC12SC;                    // Start sampling/conversion
    for(i=0; i<1000; i++);                   // small delay to wait for ADC to finish
conversion
```

```
displayVoltage(3, BattVolt, 14.9);      // displays the battery voltage on LCD

if (BattVolt > 3280)                    // displays LCD battery level indicator
  dispPwrLvl(LCD_PWR_LVL_5);
else if (BattVolt > 2460)
  dispPwrLvl(LCD_PWR_LVL_4);
else if (BattVolt > 1640)
  dispPwrLvl(LCD_PWR_LVL_3);
else if (BattVolt > 820) {
  if (blink_battery)
    dispPwrLvl(LCD_PWR_LVL_2);
  else
    dispPwrLvl(LCD_PWR_LVL_0);
  blink_battery ^=1;
}
else {
  if (blink_battery)
    dispPwrLvl(LCD_PWR_LVL_1);
  else
    dispPwrLvl(LCD_PWR_LVL_0);
  blink_battery ^=1;
}
```

}

Appendix C Serial Communications User Interface Menu Output

Serial Communications Output

*** Magnetic Stimulation Menu ***

Press 'p' to increase magnetic stimulation time

Press 'l' to decrease magnetic stimulation time

Press 'o' to increase magnetic stimulation level

Press 'k' to decrease magnetic stimulation level

Press 'M' to return to main menu

>>Magnetic ON time = 56 ms

Magnetic ON time = 62 ms

Magnetic ON time = 68 ms

*** Magnetic Stimulation Menu ***

Press 'p' to increase magnetic stimulation time

Press 'l' to decrease magnetic stimulation time

Press 'o' to increase magnetic stimulation level

Press 'k' to decrease magnetic stimulation level

Press 'M' to return to main menu

>>Magnetic ON time = 306 ms

Magnetic ON time = 300 ms

Magnetic ON time = 293 ms

Magnetic ON time = 287 ms

Magnetic ON time = 281 ms

Magnetic ON time = 275 ms

Magnetic ON time = 268 ms

Magnetic ON time = 262 ms

Magnetic ON time = 256 ms

Magnetic ON time = 250 ms

Magnetic ON time = 243 ms

Magnetic ON time = 237 ms

Magnetic ON time = 231 ms

Magnetic ON time = 225 ms

Magnetic ON time = 218 ms

Magnetic ON time = 212 ms

Magnetic ON time = 206 ms

Magnetic ON time = 200 ms

Magnetic ON time = 193 ms

Magnetic ON time = 187 ms

Magnetic ON time = 181 ms

Magnetic ON time = 175 ms

Magnetic ON time = 168 ms

Magnetic ON time = 162 ms

Magnetic ON time = 156 ms

Magnetic ON time = 150 ms

Magnetic ON time = 143 ms

Magnetic ON time = 137 ms

Magnetic ON time = 131 ms

Magnetic ON time = 125 ms

Magnetic ON time = 118 ms

Magnetic ON time = 112 ms

Magnetic ON time = 106 ms

Magnetic ON time = 100 ms

Magnetic ON time = 93 ms

Magnetic ON time = 87 ms

Magnetic ON time = 81 ms

Magnetic ON time = 75 ms

Magnetic ON time = 68 ms

Magnetic ON time = 62 ms

Magnetic ON time = 56 ms

Magnetic ON time = 50 ms

Magnetic ON time = 43 ms

Magnetic ON time = 37 ms

Magnetic ON time = 31 ms

----Welcome to UMKC EM-Stim ----

*** Main Menu ***

1: Pulse Stimulation

2: Magnetic Stimulation

3: Display Settings

4: Restore default settings

5: Test (for technical personnel only)

>>

----Welcome to UMKC EM-Stim ----

*** Main Menu ***

1: Pulse Stimulation

2: Magnetic Stimulation

3: Display Settings

4: Restore default settings

5: Test (for technical personnel only)

>>3

*** System Settings ***

Pulses = 10

Electric pulse ON time = 5 ms

Electric pulse OFF time = 45 ms

Rest time = 150 sec

Biphasic ON

Battery voltage = 3.33 V

Temperature = 40 C

Probe current = 0.0 mA

Magnetic ON time = 500 ms

Magnetic level = 12%

1

*** Electrical Stimulation Menu ***

Press 'q' to increase number of pulses

Press 'a' to decrease number of pulses

Press 'w' to increase rest time

Press 's' to decrease rest time

Press 't' to toggle biphasic

Press 'M' to return to main menu

>>

*** Main Menu ***

1: Pulse Stimulation

2: Magnetic Stimulation

3: Display Settings

4: Restore default settings

5: Test (for technical personnel only)

>>2

*** Magnetic Stimulation Menu ***

Press 'p' to increase magnetic stimulation time

Press 'l' to decrease magnetic stimulation time

Press 'o' to increase magnetic stimulation level

Press 'k' to decrease magnetic stimulation level

Press 'M' to return to main menu

>>

*** Main Menu ***

1: Pulse Stimulation

2: Magnetic Stimulation

3: Display Settings

4: Restore default settings

5: Test (for technical personnel only)

>>..... default settings restored

*** Main Menu ***

1: Pulse Stimulation

2: Magnetic Stimulation

3: Display Settings

4: Restore default settings

5: Test (for technical personnel only)

>>

*** Main Menu ***

1: Pulse Stimulation

2: Magnetic Stimulation

3: Display Settings

4: Restore default settings

5: Test (for technical personnel only)

>>5

*** Test Menu ***

Press 'f' to turn off electric bridge

Press 'r' to turn on RIGHT bridge branch

Press 'l' to turn on LEFT bridge branch

Press 'm' to turn on magnetic coil

Press 'c' to read the output of the current sense amplifier

Press 'M' to return to main menu

Probe current = 0.6 mA

REFERENCES

[1] Wu, H.-C., Young, S.-T., Kuo, T.-S., *A versatile multichannel direct-synthesized electrical stimulator for FES applications.* IEEE Transactions on Instrumentation and Measurement, 2002. **51**(1): p. 2-9.

[2] Poletto, C.J., Van Doren, C. L., *A high voltage, constant current stimulator for electrocutaneous stimulation through small electrodes.* IEEE Transactions on Biomedical Engineering, 1999. **46**(8): p. 929-936.

[3] Genovese, J.A., Spadaccio, C., Garcia-Rivello, H., Todyoda, Y., Patel, A. N., *Electrostimulated bone marrow human mesenchymal stem cells produce follistatin.* Cytotherapy, 2009. **11**(4): p. 448-456.

[4] Weber, R.J., *Functional neuromuscular stimulation*, in *Rehabilitation Medicine: Principles and Practice*, J.A. DeLisa, Editor. 1993, Lippincott: Philadelphia, PA.

[5] Reilly, J.P., *Applied Bioelectricity: From Electrical Simulations to Electropathology*. 1998, New York: Springer-Verlag.

[6] Fujita, H., Nedachi, T., Kanzaki, M., *Accelerated de novo sarcomere assembly by electric pulse stimulation in C2C12 myotubes.* Experimental cell research, 2007. **313**(9): p. 1853-1865.

[7] IonOptix. *Cell culture pacing*. 2010 [cited 12/07/2010]; Available from: http://www.ionoptix.com/products/chronic_pacing_systems.

[8] LGMedSupply. *Electronic muscle stimulators*. 2010 [cited 12/07/2010]; Available from: http://www.lgmedsupply.com/mustun.html.

[9] Brandel. *Electrical stimulators*. 2010 [cited 12/07/2010]; Available from: http://www.brandel.com/EM-Stimulator.html.

[10] Ilic, M., Vasiljevic, D., Popovic, D. B., *A programmable electronic stimulator for FES systems.* IEEE Transactions on Rehabilitation Engineering, 1994. **2**(4): p. 234-239.

[11] Jahns, M., Durdle, N., Lou, E., Raso, V. J. *A programmable ramp waveform generator for PEMF exposure studies on chondrocytes*. in *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2006. EMBS '06.* 2006.

[12] McNulty, M.J., Fogarty, P. *Design of a highly efficient circuit for electrical muscle stimulation*. in *IEEE Biomedical Circuits and Systems Conference, 2006. BioCAS 2006.* 2006.

[13] Markov, M.S., *Magnetic Field Therapy: A Review.* Electromagnetic Biology and Medicine, 2007. **26**(1): p. 1-23.

[14] Shupak, N.M., *Therapeutic uses of pulsed magnetic-field exposure: a review.* Radio Science Bulletin, 2003(307): p. 9-32.

[15] Thomas, A.W., Drost, D. J., Prato, F. S., *Human subjects exposed to a specific pulsed (200 uT) magnetic fields: effects on normal standing balance.* Neuroscience Letters, 2001. **297**: p. 121-124.

[16] Bassett, C.A.L., *Beneficial effects of electromagnetic fields.* Journal of Cellular Biochemistry, 1993. **51**: p. 387-393.

[17] MagVenture. *Transcranial magnetic stimulation*. 2010 [cited 12/07/2010]; Available from: http://www.magventure.com/.

[18] NeuroStar. *Transcranial magnetic stimulation (TMS)*. 2010 [cited 12/07/2010]; Available from: http://www.neurostartms.com/Home.aspx.
[19] Orthofix. *Cervical-stim is a non-invasive, bone growth stimulator for spine surgery*. 2010 [cited 12/07/2010]; Available from: http://www.orthofix.com/products/spine_cervstim.asp?cid=37.
[20] Cheng, K.W.E., Yan, Lu, Kai-Yu, Tong, Rad, A. B., Chow, D. H. K., Sutanto, D., *Development of a circuit for functional electrical stimulation.* IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2004. **12**(1): p. 43-47.
[21] Cheever, E.A., Thompson, D. R., Cmolik, B. L., Santamore, W. P., George, D. T., *A versatile microprocessor-based multichannel stimulator for skeletal muscle cardiac assist.* IEEE Transactions on Biomedical Engineering, 1998. **45**(1): p. 56-67.
[22] Ruohonen, J., Ravazzani, P., Grandori, F., *Functional magnetic stimulation: theory and coil optimization.* Bioelectrochemistry and Bioenergetics, 1998. **47**: p. 213-219.
[23] Polk, C., *Dosimetry of extremely low frequency magnetic fields.* Bioelectromagnetics Supplement, 1992. **1**: p. 209-235.
[24] Prato, F.S., Carson, J. J. L., Ossenkopp, K. P., Kavaliers, M., *Possible mechanisms by which extremely low frequency magnetic fields affect opioid function.* The Federation of American Societies for Experimental Biology Journal, 1995. **9**: p. 807-814.
[25] Luben, R.A., *Effects of low-energy electromagnetic fields (pulsed and DC) on membrane signal transduction processes in biological systems.* Health Physics, 1991. **61**(1): p. 15-28.
[26] Linear. *LTM4600 - 10A high efficiency DC/DC micro-module*. 2010 [cited 12/07/2010]; Available from: http://www.linear.com/pc/productDetail.jsp?navId=H0,C1,C1003,C1424,P14812.
[27] National. *LM5022 - 60V low side controller for boost and SEPIC*. 2010 [cited 12/07/2010]; Available from: http://www.national.com/pf/LM/LM5022.html#Overview.
[28] Texas Instruments. *MSP430FG439 - 16-Bit Ultra-Low-Power MCU.* 2010 [cited 10/01/2010]; Available from: http://www.ti.com/product/msp430fg439.
[29] Texas Instruments. *Ultra-Low-Power MSP430™ Microcontrollers.* 2010 [cited 10/01/2010]; Available from: http://www.ti.com/lit/sg/slab034v/slab034v.pdf.
[30] Analog Devices. *Ask The Applications Engineer—37 Low-Dropout Regulators.* 2010 [cited 10/01/2010]; Available from: http://www.analog.com/library/analogDialogue/archives/41-05/ldo.pdf.
[31] E. P. Doheny et al., *The effect of subcutaneous fat thickness on the efficacy of transcutaneous electrical stimulation*. in *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008. EMBS '08.* 2008.

VITA

Hatem Rizk was born in State College, Pennsylvania, in the year 1975. He earned his Bachelor's of Science in Electrical Engineering from the University of Missouri, in Columbia, in the year 2000. Early on during the course of his studies, he joined the Department of Surgery/ENT lab at the University of Missouri School of Medicine, in 1995, as a research technical assistant working on development of electronics and software used in the capture, processing, and analysis of brain stimulus waveforms. In the summer of 1997, he interned at Sprint in Overland Park, KS, and later that year transitioned to a full time Software Engineering role, building network design tools for the next four year. Afterwards, he became involved in developing wireless networks at Sprint, as a Radio Frequency Engineer, with emphasis on the design and construction of indoor Distributed Antennas Systems (DAS). In 2009, he earned his professional engineering licensure in the state of Missouri and started offering professional engineering consulting services to various telecom equipment manufacturers and various service providers in his post as Member Technical Staff with Global Technology Associates and as Principal at Alexandria Limited. It is also during this year, when he initiated his graduate research work at the University of Missouri in Kansas City, working in the Muscle Biology Group – MUBIG and the Advanced Circuit Analysis Laboratory. Mr. Rizk currently holds the position of Professional Radio Access Network Engineer at AT&T in Pleasanton, CA, where he is responsible for the design and implementation of state of the art high-capacity multi-technology wireless communication systems.