Design Progress Report

"Development of a Rate-Responsive, Mixed-Signal Neuron Emulator for Voltage Clamp Testing"

Angela Phongsavan, Stephen Sladen
University of Rhode Island, Dept. of Electrical, Computer, and Biomedical Engineering,
Kingston, RI 02881, USA

Abstract

The purpose of this project is to create a neuron emulator to test with a digital clamp. An intermediate step in our design process will be developing a software technique to increase the baseline voltage, which will cause an increase in the firing rate of the action potentials. The PIC18F452 is used to generate clock signals to control the switching of the MC14066, a quad analog switch. An engineering prototype has been built, generating a 110mV magnitude action potential, ranging from -90mV to 20mV. Upon inheriting this capstone project, the new engineering prototype has now been developed to output a 600mV magnitude, ranging from -500 mV to 100mV. Currently, a functional prototype is in the process of being developed in order to make the device more practical for users.

Introduction

The purpose of this design project is to develop a mixed signal neuron emulator to test a digital clamp using a single electrode voltage clamp setting. The significance of the neuron emulator will be to propagate action potentials. The term action potential can be defined as when specific ions such as sodium or potassium pass through their respective channels across the electrochemical gradient of the cell membrane. A very small voltage potential is produced when this occurs, this is the action potential we will be modeling. The task at hand will involve using the PIC18f452 microprocessor operating under the C++ programming language.

Preliminary results show that the neuron emulator has been constructed to achieve successful action potentials which are nearly glitch-free and have a low signal-to-noise-ratio (SNR). The ultimate goal of our team is to implement a baseline voltage, and cascade the baseline upwards over some time interval. Upon reaching a predetermined value of threshold, action potentials will begin to fire. A proportional linear relationship exists here between the baseline voltage and the firing rate of the action potentials. In other words, as the magnitude of the baseline voltage increases, so will the rate of action potentials being fired.

Origin of Problem

In the past, electrophysiologists have used two electrodes to measure and record the electrical activity across the cell membrane of a neuron. The previous method of obtaining a voltage clamp was performed by inserting two electrodes into the cell. This process has proved to be difficult and normally results in damages to the neuron. The neuron emulator was a project that originated as the result of a team of undergraduate and graduate students that sought to model the passive and active properties of a live neuron. With an applied input stimuli, the

device can reach a predetermined threshold value and allow the cell membrane to depolarize and propagate an action potential. Upon repolarization, the input of the stimuli increases due to a developed algorithm that increases the baseline which results in a faster firing rate of action potentials after some time. Current methods involve the possibility of retrieving the same information using a single electrode setting. This introduces us to the digital clamp, capable of achieving patch clamp techniques. Techniques include real-time switching of voltage measurements while injecting current to successfully simulate the neuron and its electrical properties.

Preliminary Results

Initial work included reconstructing a basic neuron emulator circuit which outputted action potentials that ranged from -20mV and 90mV. The schematic can be seen in Figure 1. Before the neuron emulator was realized in hardware, it was first tested in simulation using MultiSim. Two RC circuits were used to model the resting membrane potential and an action potential. These action potentials were controlled by square waves which determined when each RC circuit was charged and when the action potentials were generated.

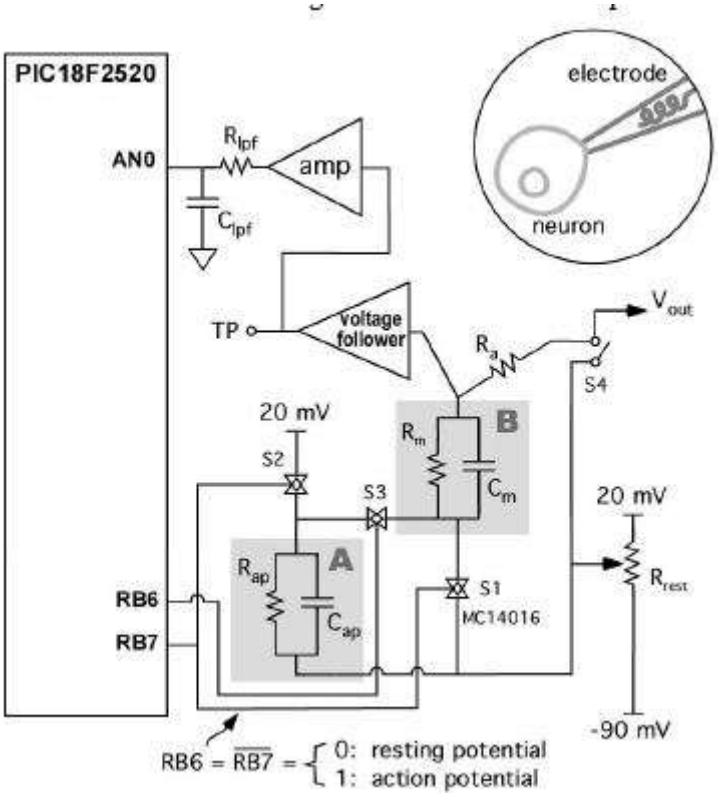

Figure 1: Schematic of the neuron emulator.

The circuit was created in the SPICE simulation environment, MultiSim. Not all the components were found in the program so minor changes had to be made. RB6 and RB7 in the model were opposing clock waves generated by the PIC microprocessor. Square wave inputs were put in place of the clock signals. The student version of MultiSim did not contain the MC14016 analog switch. Three voltage controlled switches were used for the three switches needed in the circuit. A constant -90mV source was used instead of a potentiometer which would vary the resting membrane potential. Figures 2 and 3 display the results of the final simulation.

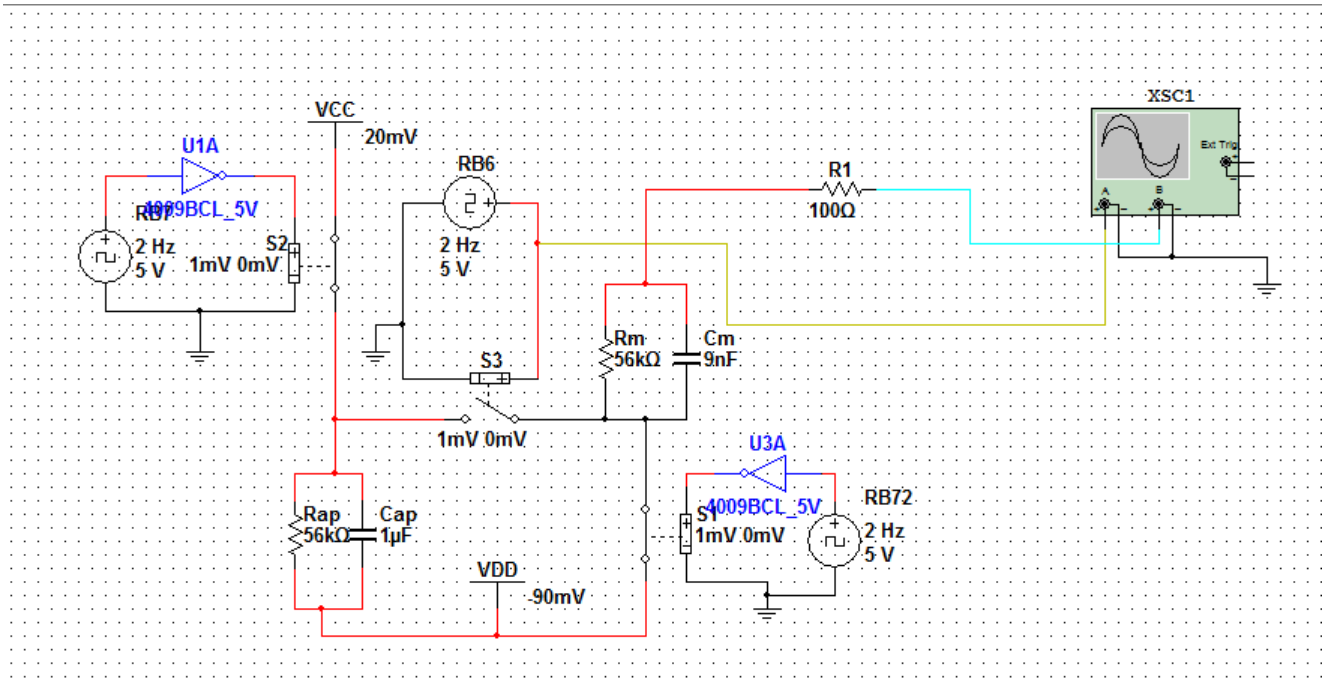

Figure 2: The neuron emulator simulated in MultiSim.

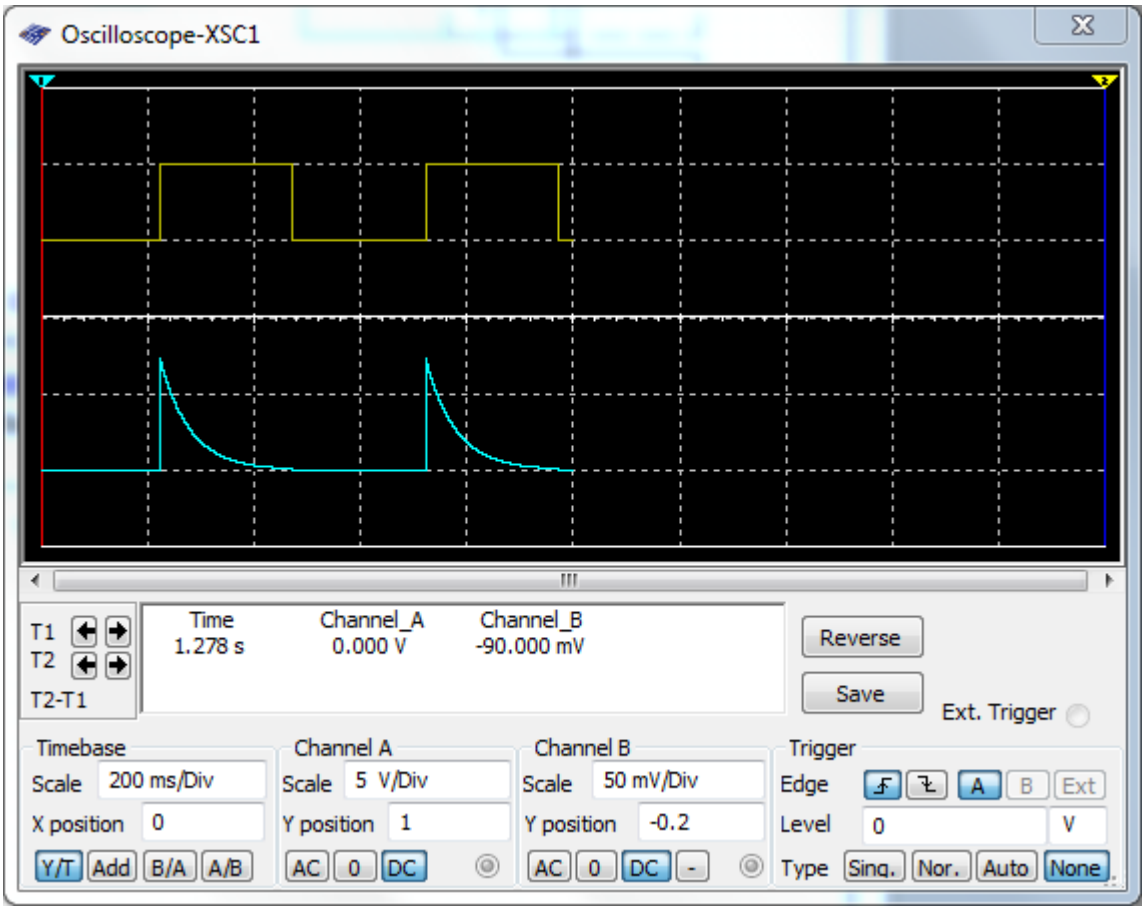

Figure 3: The square waves are displayed on channel 1 (top) and the action potentials seen at the membrane RC circuit on channel 2 (bottom).

The PIC18F52 was used to generate the square waves of opposing magnitudes and these controlled analog switches on the MC14016. These square waves are generated by ports RC0, RC1, and RC2, which are bidirectional I/O ports. RC0 is an opposing magnitude of RC1 and RC2, where (RC1 = RC2) meaning when RC0 is high (1), RC1 and RC2 are low, (0). While RC0 is low, the circuit is simulating the membrane potential at rest. Figure 4 displays the three switches under operating conditions.

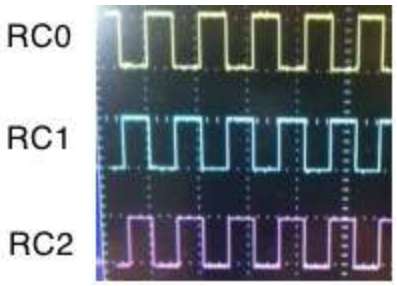

Figure 4: The three square waves inputs controlling the switches in the circuit that are generated by the PIC from bidirectional ports RC0 (top), RC1 (middle), and RC2 (bottom) are shown on the oscilloscope.

During this time, the RC circuit which consists of a resistor and a capacitor combination, of the action potential is only connected to 20mV and a -90mV ground reference. The 20mV and -90mV sources are created from voltage divisions of resistors using a 5V voltage regulator, the LM7805, and -9V from the LMC7660, a voltage inverter. This causes the action potential circuit to possess a magnitude of 110 mV. Once RC0 is high, the 20mV source is disconnected from the RC circuit of the action potential and it is then connected to the membrane RC circuit. The membrane potential circuit is fed through a voltage follower. The voltage follower is achieved using a LMC6001 op-amp chip, an ultra low-input current op-amp. The action potentials are seen at the output of the voltage follower.

The action potentials resulting from the initial circuit were both noisy and contained unwanted artifacts found at the switching of RC0 going from logic 1 to 0. Figure 5 displays the initial output of the voltage follower (channel 2) and its alignment in time with the rising edge of RC0 (channel 1). The artifact can be seen on the falling edge of RC0.

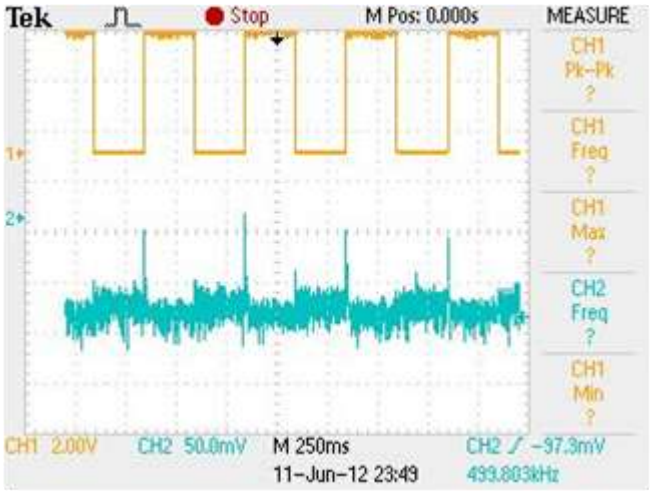

Figure 5: Channel 1 displays the square waves while Channel 2 displays the output of the voltage follower.

It was believed the analog switches primarily caused the artifacts. A variety of different methods were used to attempt to eliminate the artifacts, all of which were unsuccessful. It was discovered later that one of the voltage division sources contained the artifact that disrupted the desired output. A capacitor was also placed at the -90mV reference point which served as virtual ground. The result of this diminished the artifact and reduced some of the noise. Upon discovering a solution to noise, the outputs of the sources were tied down to ground through a capacitor (coupling). The integrated circuit chips that received direct sources were also coupled with a capacitor which reduced the overall noise of the output.

The action potentials are sent through the PIC in order to perform an echo operation to ensure the signal is being read correctly by the microprocessor. A digital-to-analog converter was added to the existing circuit for the PIC to convert the digital action potentials, converted by the echo program upon entering the PIC, back to an analog signal. The echo program was tested with positive value signals and was found to be working correctly. The action potentials (which ranges from -90mV to 20mV) were placed into the PIC but were not outputted correctly. The reference points (pins 4 and 5) on the PIC were altered in order to zoom in on the small signal. A small negative voltage was set at V_{ref}- but the PIC still did not output the correct signal.

Realistic Constraints

Economic

The neuron emulator is a relatively inexpensive device. According to the bill of materials, most materials are composed of passive components (i.e. resistor, capacitors). The rest of the materials, integrated circuit chips, still remain inexpensive. The only remotely expensive components would be the PIC18F4525, where the cost of each microprocessor is slightly under $5, and the LMC6001BIN op-amp, which retails for about $12.30. Therefore, the entire design project could be assembled under a budget of $30.

Ethical

A responsible engineer will take it upon him/her self to ensure the safety of the public and those who immediately interact with developed devices are also protected. As engineers who are on their way to entering industry, would practice a well placed code of ethics. In our design process, we considered moral decisions that benefit the health and safety of any user who could come into contact with the device.

Health and Safety

The neuron emulator is an instrument that is powered by a 9V battery. Being a low voltage DC powered device, the risk of accidental shock is minimized. Further safety precautions were taken by providing the prototype additional means of earth grounds. All of the components comply with the IDC 60601-1 which classifies them to be a Class 1, Type B applied parts. The

IC components also comply with the RoHS standard by not including materials that are hazardous to our health such as lead, mercury, etc.

Manufacturability

The original engineering prototype was first made on a breadboard. Time constructing this is miniscule. When taking this into a functional prototype, we will face other constraints. The amount of patience and the level of precision are two main concerns here. This is when we will be creating the infamous "black box" so that the device can be moved easily and demonstrated as necessary. Using an epoxy protoboard, the soldering holes are extremely small, therefore, the ability to solder well will also be addressed. Component placement will also be of concern so that errors can be minimized. Overall, the neuron emulator is surely a device that could be manufactured with ease due to the level of safety taken in relation to component considerations and the inexpensive bill of materials.

Engineering Standards

With safety being a top priority for the devices biomedical engineers develop, ethical engineers came to an agreement and set forth specific safety regulations to protect the public against harm and malpractice from insufficient designs in medical devices. The IEC (International Electrotechnical Committee) has set forth a series of technical standards for medical device equipment. The most general standard many countries choose to follow is the IDC 60601-1. Recently, a new process was adapted where one must comply and fill out an ordinance form with the ISO 14971 standard which is known as the "International Standard for Application of Risk Management to Medical Devices." When incorporating a risk management section into an experiment, one is complying with 3rd revision. One beneficial aspect of complying with the 3rd (2005) revision is the option to include new technologies and methodologies to develop new devices that were not recognized by the limitations and parameters set forth by the 2nd revision. Underwriters Lab (UL) allows the designer to choose between the 2nd and 3rd revision to the standard.

Underwriters Labs, mostly known as the safety consultant, they often provide services that validate, test, inspect, audit, and advise to manufacturers, retailers, and consumers. UL was approved to perform these safety tests for the public by the US federal agency known as OSHA (Occupational Safety and Health Administration), which identifies UL as a "Nationally Recognized Testing Laboratory.

One last engineering standard that should be accounted for is the RoHS (Restriction of Hazardous Substances.) This directive was adopted so that lead-free components could be restricted from being used in electronic devices and equipment. Any RoHS compliant component is known as a "lead-free device," and is tested for six substances: Lead (Pb), Cadmium (Cd), Mercury (Hg), Hexavalent chromium (Hex-Cr), Polybrominated biphenyls (PBB), and Polybrominated diphenyl ethers (PBDE), the last two are plastics found in flame retardants. By

today's standards, the maximum permitted concentrations of any these substances is 1000 ppm (parts per million.) In relation to our neuron emulator capstone design, all of the integrated circuit chips, (i.e. LMC7660, LM324, LMC6001, DAC0800, MC14066, and the PIC18f452) are all RoHS compliant according to manufacturers' specification sheets. The neuron emulator is also IDC 60601-1 compliant by identifying the device to be a Type-B applied part due to its specific component requirements to protect users against electrical shock. It can be further classified to be a Class I type device where extra protection has been taken into account by added earth protection (i.e. ground). European Conformity (EC) can be further made about the device under the ISO 14021 standard where claims made are accurate, as well as not being misleading.

Bill of Materials

4 MHz ceramic resonator
capacitors (F):

- (3) 0.01µ
- (4) 0.1µ
- (5) 1µ
- 4.7µ
- 220µ
- (2) 47µ
- (4) 100µ
- 220µ
- 9n

DAC0800
ISCP-3 Header
LM324
LM7805
LMC7660
LMC6001
MC14066
PIC18F4525
push button
resistors (Ω):

- 100
- 270
- 470
- 680
- (2) 1k
- (2) 4.7k

- (8) 10k
- 18k
- (3) 2 k
- 47k
- (2) 56k
- 10k potentiometer

Design Alternative

The design process of the neuron emulator was began with an idea that would allow performance of a voltage clamp on a simulated neuron without disrupting the cell membrane upon examination. Design components were chosen with very specific specifications in mind. The engineering prototype was designed on a breadboard using a 9V battery input. Except for small details, the overall circuit is run using 5 volts, the purpose being the PIC input voltage maximum. This also adds a safety factor using low voltage, therefore reducing the risk of shock to the user. The actual microprocessor we chosen for this experiment was the PIC new 18 series, the 18f4525. With past knowledge of the PIC from our Biomeasurement Laboratory in which we design biological signals, we chose to use this. The baseline voltage, known as the resting membrane potential was sought out by selecting to include a voltage inverter. The magnitude was then chosen to range from -500mV to +100mV. The alternative to replace the chosen resistors with potentiometers to achieve the required voltage divisions was a discussion amongst team members. Ultimately the resistors remained, and as an addition, capacitors were coupled in order to reduce noisy spikes from the output. For the PIC to be able to recognize the very small signal we are sending it, an amplifier is needed. The LM324 has four low-power internal op-amps that have high voltage gains. It operates under a single power supply, using a range of voltages and satisfying our +/- 5V voltage rails. Deciding on the amplifier application we should construct was an issue of concern. Different attempts were executed such as the combination of inverting and non-inverting amps. However, we were not successful in raising the baseline without biasing the amplifier. When a potentiometer was applied to the amplifier, the baseline was able to be adjusted by limiting the amount of current being pulled to ground. Creating a functional prototype or a black box was thought to be the next appropriate step. Concerns arose about what materials we would use for the soldering board. The option of printed circuit boards (PCB's) was denied because of the possibility of making a mistake when designing the board and wasting money. A functional prototype is currently in the process of being developed on a epoxy fiberglass protoboard. A switch was chosen to be included so that the user can manually turn on/off the device.

Project Management

The overall workload of the project was distributed into three components amongst two

team members. Angela Phongsavan chose to accept the responsibility of being the project manager, who oversees the progress of the project and is ambitiously determined to meet the deadlines set forth by realistic constraints. The hardware engineer role was assigned to Stephen Sladen, who has a broad knowledge of system components, as well as being proficient in linear and nonlinear circuit theory. Angela Phongsavan volunteered to take up the role of software engineer due to her already existing background with the C++ operating language based program MPLab. Angela also has a previous history involving the neuron emulator design project, working with Prof. Ying Sun during the summer of 2012, during which she was responsible in creating a stock code to work off of this semester as well as design RC specifications of the action potential.

Methods

Upon investigating older designs and publications concerning the neuron emulator, new specifications were developed that needed to be implemented into our new design. A value was selected for the magnitude of our action potentials to be 600mV, ranging from -500mV to 100mV. This was achieved by using resistors and applying basic voltage divisions. The PIC microprocessor also has complications reading negative voltages. This indicates some amplifier type will be needed to raise the baseline. Using an LM324 quad op-amp switch, two amplifiers were constructed. The first consists of an inverting summing amplifier which possesses a stage gain of 2. A DC bias was constructed amongst the summing amplifier by inserting a potentiometer to allow us to shift the baseline into a positive value so that the PIC is able to read the data. The next op amp in series is an inverting amp, which possesses a gain of four. Using linear system techniques, the two amplifier gains are cascaded together and multiplied, final result yielding a gain of 8. This satisfies the conditions in which the magnitude of the action potential to fall in the range of 0-5V. The microprocessor then sends a digitized signal to the digital to analog converter (DAC). When constructing all of the new hardware improvements onto a new breadboard, we encountered a slight glitch. On the falling edge of the action potential, a discontinuous edge existed approximately 50mV above the resting membrane potential. After making minor RC modifications and going over time constant values helped minimize the edge. A block diagram of the device is shown in Figure 6. The current program in the PIC is echo. The action potentials can be observed at the output of the inverting amp as seen in Figure 7. Figure 8 displays the analog output from the DAC.

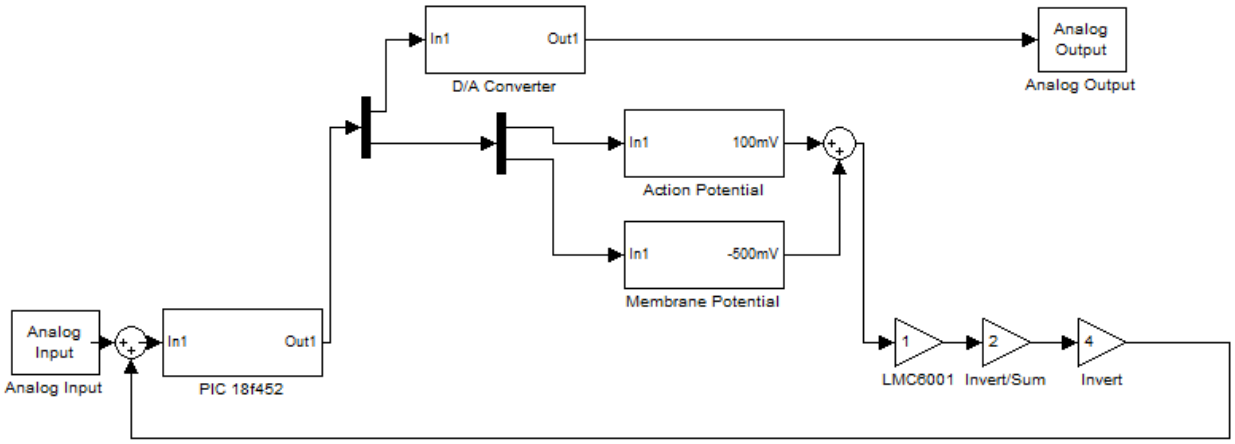

Figure 6: Block diagram of the current neuron emulator.

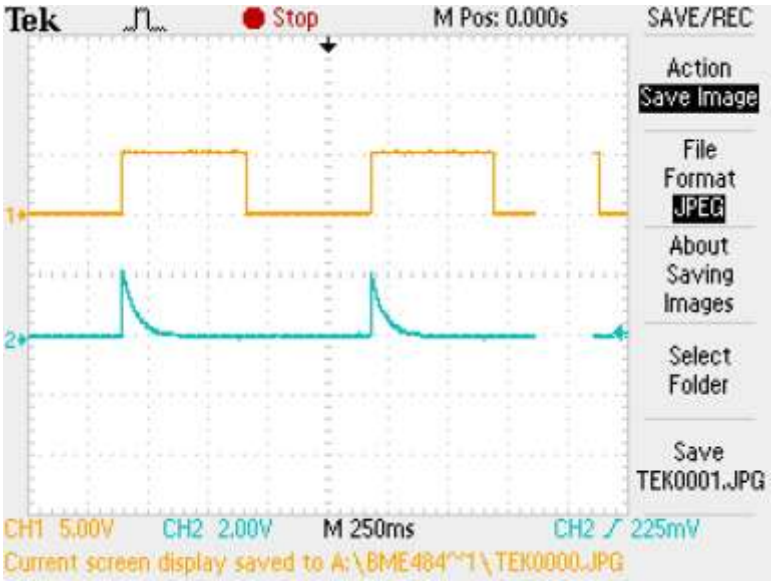

Figure 7: Channel 1 displays the square wave outputs from RC0. Channel 2 shows the output of the two stage gains which is fed to the input of the PIC (pin 2).

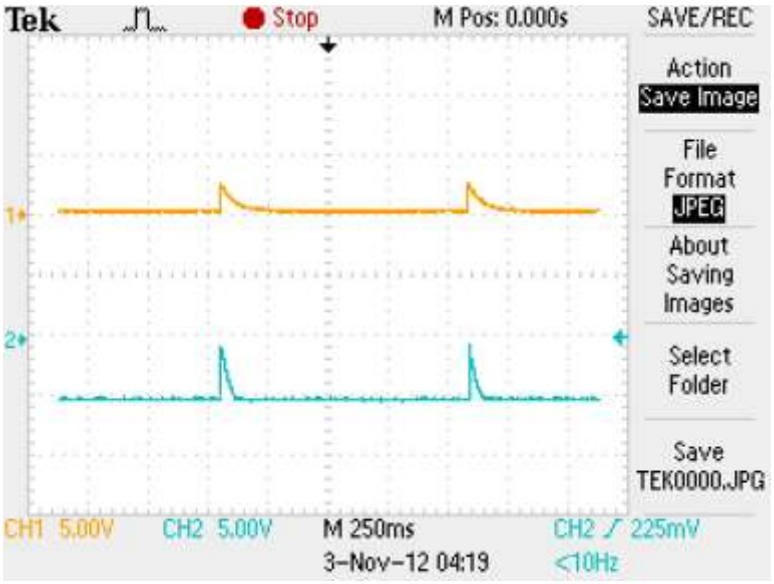

Figure 8: Channel 1 exhibits the input to the PIC while channel 2 is the output of the DAC.

References

[1] R. Rieger, Y. Sun, and Y.C. Wu, “Implementation of a Neuron Emulator for Patch-Clamp Settings”
[2] G. Ausfresser, and Y. Sun, “Neuron Emulation, Instrumentation, and Communication for a Neuroscience Instrument” 38th Annual Bioengineering Conference.” Philadelphia, PA. March 16th- 18th, 2012.
[3] Y.C. Wu, J.Y. Chen, R. Rieger, and Y. Sun, A neuron emulator for single-electrode settings. in Proc. IEEE 37th Annual Northeast Bioengineering Conference, 2011.
[4] J. Wu, Y. Sun, J. DiCecco, and R. Hill. A universal instrument for biological measurements, stimulation, clamping, and processing. 31st Annual Northeast Bioengineering Conference, Hoboken, NJ, April 2-3, 2005.
[5] Ying Sun, PhD. & Eugene Chabot, PhD.
[6] Brian Ramos, Graduate Student
[7] Datasheet, PIC18F452, Microchip Technology Inc., 2002.
[8] Datasheet, SerLED, Serial enabled LED, <www.sparkfun.com>, 2004.
[9] Datasheet, MC14066B, Semiconductor Components Industries, LLC, 2011.
[10] J. G. Webster, Medical Instrumentation-Application and Design, 3rd ed., Wiley, 1998.
[11] PIC Sample Code in C, http://www.microchipc.com/, 2011.
[12] eBook, C Programming Tutorial for PIC Microcontroller Application, <www.ebooksquad.com/2009/06/28/cprogramming-tutorial-for-pic-microcontroller-application.html>, 2011.
[13] Datasheet, DAC0800, National Semiconductors, <datasheetcatalog.com>, 1995.

Appendices

Appendix A - C code

```
/********************* Specify the chip that we are using *************************/
#pragma chip PIC18f4525

/************************* Define prototype functions ***************************/
void _highPriorityInt(void);
void delay_ms(unsigned char x);

/****************************** Global variables ********************************/
unsigned char output;

#pragma origin 0x8
interrupt  highPriorityInterrupt(void)
{
        _highPriorityInt();      // 8 code words available including call and RETFIE
        // Restore W, STATUS and BSR from shadow registers:
        #pragma fastMode
}

unsigned char ReadADC() /********* start A/D, read from an A/D channel *************/
{
unsigned char ADC_VALUE;
        GO = 1;                  // Start the AD conversion
        while(!ADIF) continue;   // Wait until AD conversion is complete
        ADC_VALUE = ADRESH;      // Return the highest 8 bits of the 10-bit AD conversion
        return ADC_VALUE;
}

void _highPriorityInt(void) /***** high priority interrupt service routune ********/
{
checkflags:
        if(TMR0IF == 1) {                // When there is a timer0 overflow, this loop runs
                TMR0IE = 0;              // Disable interrupt
                TMR0IF = 0;              // Reset timer 0 interrupt flag to 0
                TMR0H = 0xFC;            // Reset timer count: high-order and low-order bytes...FC17 for
1ms
                TMR0L = 0x17;            // $FFFF - $F64F = $09B0 = 2480 (decimal) => ~2.5 ms
                output = ReadADC();
                PORTD  = output;         // Output to the D/A via the parallel port D
                TMR0IE = 1;              // Re-arm by enabling the Timer0 interrupt
        }
}

void SetupADC(unsigned char channel) /***** configure A/D and set the channel ******/
{
        TRISA = 0b.1111.1111;    // Set all of PORTA are inputs
        // ADCON1 Setup
        // bit 7: Left justify result of AD (Lowest 6bits of ADRESL are 0's)
        // bit 6: Set to Fosc/8
        // bit 5-4: Unimplemented
        // bit 3-0: Configuration of 8 AD ports (Set all 8 inputs to Analog)
        ADCON1  = 0b.0000.1111;
        // ADCON0 Setup
        // bit 7,6 = 1,0: Set to Fosc/8
        // bits 5-3 = Channel select
        // bit 2: GO Bit (Starts Conversion when = 1)
        // bit 1: Unimplemented
        // bit 0: AD Power On
        ADCON0 = (channel << 3) + 0b.0100.0001;
        ADIE = 0;                // Turn off the AD interrupt
        ADIF = 0;                // Reset the AD interrupt flag
}
```

```
void delay_ms(unsigned char x) /* generate a delay for x ms, assuming 4 MHz clock **/
{
        unsigned char y;
                for(;x > 0; x--) for(y=0; y< 165;y++);
}

/************************ main program ******************************/

void main()
{
        TRISC = 0x00;    // Set all port B pins as outputs
        PORTC = 0x06;   //
        TRISD = 0x00;    // Set all port D pins as outputs
        T0CON = 0x88;
        INTCON = 0xA0;
        SetupADC(0);
        while(1) {
                PORTC.0 = !PORTC.0;      // toggle RC0 to turn LED on and off
                PORTC.1 = !PORTC.1; // toggle RC1 on and off
                delay_ms(250);
                delay_ms(250);
                PORTC.2 = !PORTC.1;
        }
}
```

Appendix B - Figures

Appendix C - Formulas Used

Equation 1: Voltage Division

$$V_{out} := \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

Equation 2: Gain of a Summing Amplifier

$$\frac{V_{out}}{V_1 + V_2} = \frac{-R_f}{\frac{(R_1 \cdot R_2)}{R_1 + R_2}}$$

Equation 3: Gain of an Inverting Amplifier

$$\frac{V_{out}}{V_{in}} = \frac{-R_f}{R_{in}}$$

Equation 4: Gain of a Non-Inverting Amplifier

$$\frac{V_{out}}{V_{in}} = 1 + \frac{R_f}{R_{in}}$$