Wireless 3D System-on-Package (SoP) for MEMS Movable Microelectrode

by

Li Zhou

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2012 by the
Graduate Supervisory Committee:

Jitendran Muthuswamy, Chair
Jemmy Sutanto
Hongyu Yu

ARIZONA STATE UNIVERSITY

May 2012

ABSTRACT

There is a tremendous need for wireless biological signals acquisition for the microelectrode-based neural interface to reduce the mechanical impacts introduced by wire-interconnects system. Long wire connections impede the ability to continuously record the neural signal for chronic application from the rodent's brain. Furthermore, connecting and/or disconnecting Omnetics interconnects often introduces mechanical stress which causes blood vessel to rupture and leads to trauma to the brain tissue. Following the initial implantation trauma, glial tissue formation around the microelectrode and may possibly lead to the microelectrode signal degradation.

The aim of this project is to design, develop, and test a compact and power efficient integrated system (IS) that is able to (a) wirelessly transmit triggering signal from the computer to the signal generator which supplies voltage waveforms that move the MEMS microelectrodes, (b) wirelessly transmit neural data from the brain to the external computer, and (c) provide an electrical interface for a closed loop control to continuously move the microelectrode till a proper quality of neural signal is achieved.

One of the main challenges of this project is the limited data transmission rate of the commercially available wireless system to transmit 400 kbps of digitized neural signals/electrode, which include spikes, local field potential (LFP), and noise. A commercially available Bluetooth module is only capable to transmit at a total of 115 kbps data transfer rate. The approach to this challenge is to digitize the analog neural signal with a lower accuracy ADC to lower the data rate, so that is reasonable to wirelessly transfer neural data of one channel.

In addition, due to the limited space and weight bearing capability to the rodent's head, a compact and power efficient integrated system is needed to reduce the packaged volume and power consumption. 3D SoP technology has been used to stack the PCBs in a 3D form-factor, proper routing designs and techniques are implemented to reduce the electrical routing resistances and the parasitic RC delay. It is expected that this 3D design will reduce the power consumption significantly in comparison to the 2D one.

The progress of this project is divided into three different phases, which can be outlined as follow:

a) Design, develop, and test Bluetooth wireless system to transmit the triggering signal from the computer to the signal generator. The system is designed for three moveable microelectrodes.
b) Design, develop, and test Bluetooth wireless system to wirelessly transmit an amplified (200 gain) neural signal from one single electrode to an external computer.
c) Design, develop, and test a closed loop control system that continuously moves a microelectrode in searching of an acceptable quality of neural spikes.

The outcome of this project can be used not only for the need of neural application but also for a wider and general applications that requires customized signal generations and wireless data transmission.

DEDICATION

To my Mother and Father

ACKNOWLEDGMENTS

This Dissertation would never have been written if not for the help of the following:

Jemmy Sutanto; Jitendran Muthuswamy; Zikai Tee; Sindhu Anand; Shuang Hu.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

APPENDIX Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Advanced integrated circuit technology and new biocompatible materials have enabled the development of sophisticated biomedical implants to record of neural activity. The recorded neural signals are very useful for neuroscientists to study modern systems neuroscience. There is a great need for technologies that enable neuroscientists and clinicians to observe neural activity associated with a task or behavior in real time [1], [2], [3]. Microelectrodes implanted in the brain can record extra-cellular potentials and provide excellent spatial and temporal resolution of neural activity [4]. Hence, implantable microelectrodes are widely used in clinically viable brain-machine interfaces (BMI) for neural signal acquisition. Advanced MEMS technology has the capability to develop large scale neural recording devices with as many as 100 independent electrodes to obtain multiple channels' of neural signal, which creates a significant impact in clinical medicine and the general neuroscience field with many potential applications [1], [3]. With the realization of multichannel recording technology, physiologists can comprehend the operation of individual neurons, understand the signal processing techniques and organization of biological neural networks and even control multiple prosthetic devices at the same time [5].

1.1. Microelectrodes

Most current designs of the microelectrodes for neural data recording are fixed, which means the depths and positions of the microelectrodes in the brain cannot be changed. They can only record activity of neurons in the region they are implanted initially. If the quality of the neural signal in that position is not acceptable for further analysis, there is no good method to adjust the position of the microelectrodes to search for sites with better quality of recordings. MEMS technology enables the creation of miniaturized movable structures. Hence it was conceptualized that microactuators can be fabricated using this technology to move the electrodes [6], [7], [8]. A new actuation scheme for in-plane, bi-directional translation of polysilicon microelectrodes using electrothermal microactuators for brain implant applications has been designed in our laboratory [6], [9], [10], [11]. With movable microelectrodes design, it is very convenient to control the movements of multiple microelectrodes in order to search for acceptable quality of neural signals.

1.2. Electronics to Control Movement of Microelectrodes and Record Neural Activity

In order to control the movement of MEMS a microprogrammed control unit (MCU) provides the control signals to actuate the microelectrodes. Higher voltage and current are necessary to drive the electrothermal microactuators to move the multiple microelectrodes. Hence, the control signals from output pins of MCU have to be amplified and demuxed.

1.3. Pre-Amplifier

The voltages of neural signals in the brain are usually at the level of micron volts, which is in the same level as noise of the outside world. External interfering and noise signals would easily disturb the weak neural signals (<500uV) [11]. In addition, dc offsets of 1 to 2 volts are common across the microelectrodes because of the electrochemical effects at the electrode-neuron interface. Therefore, in order to keep the high enough signal to noise ratio (SNR) and reject the dc offsets at the input of microelectrodes during the neural data acquisition process, it is necessary to use a low noise bioamplifier to amplify the neural data first before processing and transferring it.

1.4. Analog to Digital Converter

In order to enhance the signal to noise ratio (SNR) during the transmission process, make the data transferring easier, allow on-board digital signal processing for compressing neural data and achieve higher bandwidth, it is very necessary to digitize the analog data to digital format with on-board analog to digital converter (ADC) before being transmitted to the outside world [5].

1.5. Need for Wireless Systems

A current limitation of neural signal and control signal transmission is the method of transmission of the signals to and from the data acquisition devices. Right now, for most neural signal recording, wired connections are used to transfer neural data from implanted microelectrode arrays to external instrumentations. Such wired connections are not limited by data transfer rate. Thus, those neural data recording devices must be connected to external equipments with relatively bulky wire bundles and connectors [1], [12]. In addition, external interference and noise would easily couple to the wires. Also, it is very inconvenient to continuously send control signal to movable microelectrodes or record neural activity from the rodent's brain for a relatively long time with the wire-interconnects system. Furthermore, the test animal cannot move freely with the limited length of wire connection. In order to get rid of the trauma to the brain tissue caused due to the connecting and/or disconnecting of wired Omnetics™ interconnects, this type of neural signal acquisition system must communicate with the external equipments and obtain power without the use of wires [11]. With the implantable units that utilize batteries for power and wireless telemetry for transmission of neural signals and control signals, the system avoids those issues associated with hard-wired connection. Although it has to face the problems of fast discharging of the batteries and the requirement of high power consumption [5].

1.6. Wireless System-on-Package for Neural Recording and Control of Microelectrodes

In an effort to create neural recording devices with wireless power and data transfer, microelectromechanical system (MEMS) electrode arrays are being combined with preamplifier and signal generator in 3D stacked PCBs [10], [11]. For now, an internal source (a coin-sized battery, for example) provides power for the implanted module. Ideally, these circuits should be wirelessly powered since rechargeable batteries are relatively large and have limited lifetimes. In addition, the systems will be tested on rodents, so such systems must deal with not only the power constraints, but also the size [13], [14]. The complete packaged system should be very compact. Furthermore, low power operation is essential for any small implanted electronics as elevated temperatures can cause damage to biological tissues [15], [16]. Moreover, by reducing the total length of the routing onboard, the 3D stacked technique consumes less power and decreases the space of the complete system.

1.7. Thesis Overview

Due to limited time and budget, the RN-41 Bluetooth module from Roving Networks is used in the wireless transmission, which has the maximum baud rate of 115,200 bits/s. It is a very robust link both in signal integrity and transmission distance, and the wireless transmission quality has been tested within 15m (maximum distance in the lab). The ultimate goal is to achieve a fully implantable robotic microelectrode system for neural signal acquisition with the capability to manipulate the microelectrodes wirelessly. The main contribution of this thesis is to develop a completely packaged system for enabling this goal. A design to wirelessly transfer one channel of microelectrode recording and control of movement of three microelectrodes in an array has been implemented.

1.8. Thesis Outline

The organization of the thesis is as follows: Chapter 2 presents the complete system design; Chapter 3 shows the results and analysis; Chapter 4 summarizes the thesis and presents the future plan.

CHAPTER 2

SYSTEM DESIGN

2.1. System Introduction

There are two system designs in the project. The first one is the wireless transmission system for one channel to get a proper quality of neural signal with movable microelectrode closed loop control. The second one is manually controlling multiple microelectrodes movement by wirelessly sending triggering signal from an external computer to a microcontroller.

2.2. System Flow

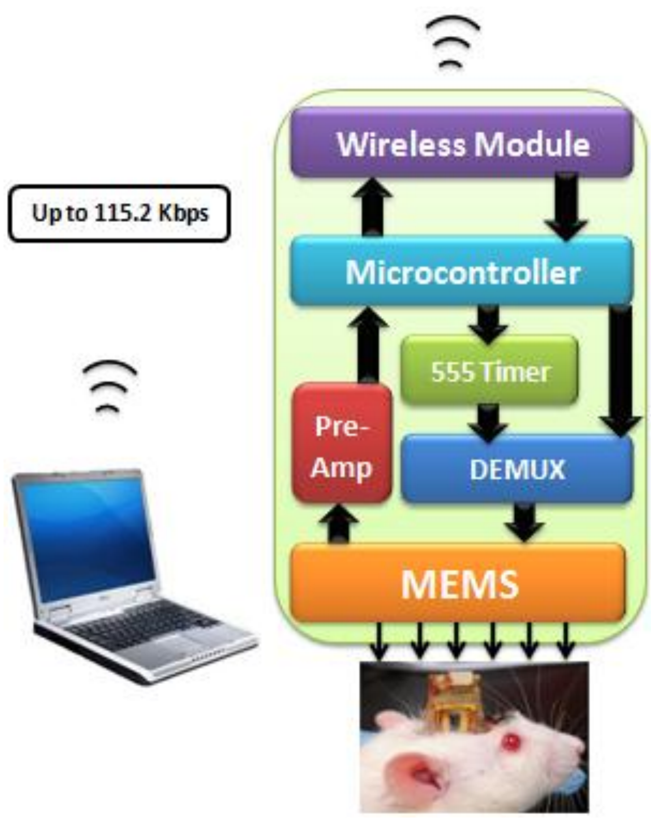

Figure 1: Schematic representing the bi-directional flow of signals. Neural signals recorded from the rodent cortex are transmitted through the pre-amp to a microcontroller and transmitted wirelessly. Control/trigger signals from the computer are wirelessly

transmitted to the microcontroller, which generates the set of waveforms to move the appropriate microelectrode as selected by the demux.

a) System 1 is designed for one microelectrode only due to both the data transfer limitation of the Bluetooth module and the sampling rate of the ADC in the microcontroller. As shown in Figure 1, first, the MEMS microelectrodes acquire the neural data from rodent's brain. Secondly, the preamplifier amplifies the neural data by 200 times. Thirdly, the ADC in the microcontroller samples the input amplified neural data with the frequency of 10 kHz. Fourthly, microcontroller sends the digitized neural signal wirelessly to external computer through Bluetooth module. Fifthly, computer displays the received neural data and stores it for further processing. Meanwhile, the control algorithm implemented on the microcontroller will keep checking the quality of the neural signal. If the signal quality is not acceptable, then the microcontroller will automatically send the control signal to MEMS chip to move the microelectrode until a proper quality of neural signal is achieved. Below is how the close-loop control works.

   i) Microcontroller keeps reading neural data in certain of window of time A (which can be programmed in the microcontroller). If the amplitude of the incoming signal is higher than the threshold, it is defined as a spike. The threshold is set as 3 times the standard deviation above the mean of the incoming data. Since the data sampling rate of the on board ADC is 10 KHz, and normally each spike lasts 1 to 3 millisecond, theoretically each spike could have no more than 30 samples. Therefore, when there is a spike being detected, start to transmit the next 30 points of neural data. After the transmission, microcontroller keeps searching spikes within the window A. If there are more than 20 spikes being detected within the amount of neural data A, which means the quality of the neural signal is acceptable with the current position of microelectrode. Then after the process of reading A, recalculate the new threshold, then redo step 1.

ii) If there are no enough spikes being detected in the window to time A, the microcontroller will automatically generate the movement waveforms to move the microelectrode by a single step. After the microelectrode movement waveforms are sent, redo step 1.

b) System 2 is a wireless control system for multiple microelectrodes, which could manually control each one of the electrodes to move forward or backward individually with simple commands from external computer. With the capability of controlling more than 16 channels individually, it is designed for 3 microelectrodes only due to the PCB space limitation. According to Figure 1, first, when you want to move the microelectrodes, simply type in the corresponding command in the Arduino software, the computer will send the wireless triggering signal to microcontroller though the Bluetooth wireless module. Secondly, when the microcontroller receives the trigger command from computer, it will send a confirmation signal back to the computer, and it will generate the corresponding control signals to the MEMS chip. Thirdly, the MEMS chip will control the movements of each electrode.

i) Once the command of microelectrode's number and movement direction is input, the computer will wirelessly send it to the microcontroller through the Bluetooth module.

ii) The microcontroller generates the corresponding four waveforms for the selected microelectrode.

iii) The waveforms are transmitted via the 555 Timers and the demultiplexers, to the MEMS chip.

iv) The MEMS chip moves the microelectrode forward/backward.

Table 1: Trigger Signals for Generating Movement Control Waveforms for 3 Microelectrodes

Trigger Signal	Microelectrode Selection	Direction Selection
a	1	Forward
b	1	Backward
c	2	Forward
d	2	Backward
e	3	Forward
f	3	Backward

2.3. Pre-Amplifier PCB Design

The pre-amplifier PCB is based on the Intan RHA2000 series amplifier chip, which is a fully integrated 46 dB (200 V/V) amplifier. This amplifier chip has

a) low input-referred noise: 2 μVrms;

b) low power operation: less than 500 μW per channel;

c) low supply voltage requirements (2.9V-3.6V) permit operation from small batteries;

d) upper cutoff frequency of all amplifiers set by external resistors: adjustable from 10 Hz to 20 kHz;

e) lower cutoff frequency of all amplifiers set by external resistor: adjustable from 0.02 Hz to 1.0 kHz, true zero gain at DC rejects electrode offset voltages.

All the features are suitable for the low-power wireless headstage for neurophysiology experiments. At the core of each RHA2000-series chip is an array of low-noise amplifiers with a mid-band gain of 46 dB (200 V/V) and built-in filtering to isolate frequencies of interest.

The amplifier has a 3rd-order Butterworth low-pass filter, which could minimize aliasing and reject the noise and signals that are beyond the desired. All possible DC offset voltages at the input pins are completely rejected by the internal capacitors, thus, eliminates the problems with built-in potentials at electrode-tissue interfaces [10].

The upper bandwidth of the amplifiers may be programmed to any value between 10 Hz and 20 kHz, and the lower bandwidth from 0.02 Hz to 1.0 kHz, by means of three external resistors per chip. This flexibility allows the chips to be optimized for different types of signals (e.g., 0.1 – 100 Hz for EEG or EKG signals, 250 Hz – 7.5 kHz for neural action potentials). Frequency of 1-7.5 kHz is the range of neural spike potentials and local field potentials. Two off-chip resistors, RH1 and RH2, are tied to the pins setH1 and setH2 to set the upper bandwidth (fH) of the amplifiers. An external resistor RL tied to the pin setL to set the programmable lower bandwidth of RHA2000-series chip.

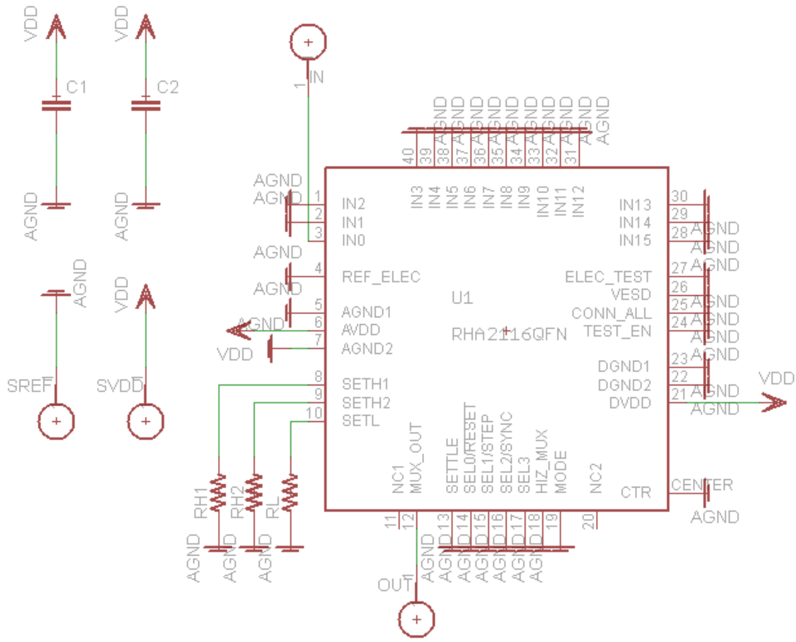

Figure 2: Schematic Design for Pre-Amplifier

As shown in Figure 2, the RH1 and RH2 are chosen as 15.8 kΩ and 26.7 kΩ to set the upper bandwidth (fH) as 7.5 kHz, and RL is chosen as 86.6 kΩ to set the lower bandwidth (fL) as 1Hz [17].

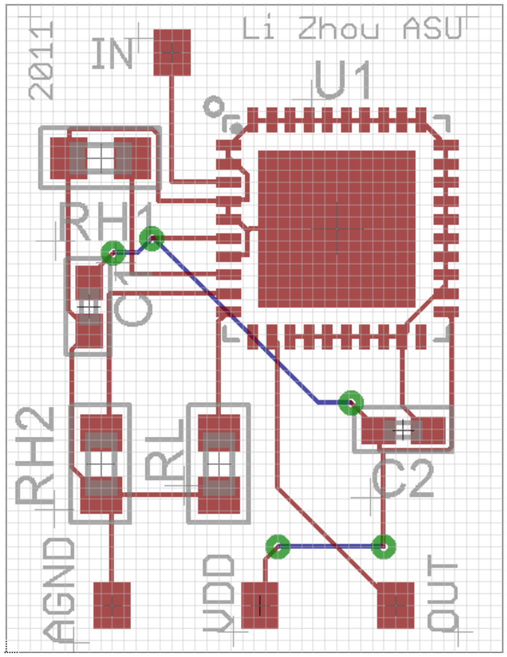

Figure 3: Printed Circuit Board Design for Pre-Amplifier

As shown in the Figure 3, the PCB has two layers of routing, red and blue. All the electrical components and input output pins are on the red side of the PCB.

2.4. Waveforms for Moving the Microelectrodes

The moveable microelectrode MEMS chip needs four square waveforms to drive one microelectrode, which are forward drive, disengage reverse drive, reverse drive and disengage forward drive. The voltage level of forward drive and reverse drive is 7 V, and the voltage level of disengage reverse drive and disengage forward drive is 6.5 V. The period of the waveforms for a single step movement is 6t (where t is time period). Each time period t is set to 100 ms for visualization of movement of actuators. The Figures below are the waveforms for driving the microelectrodes.

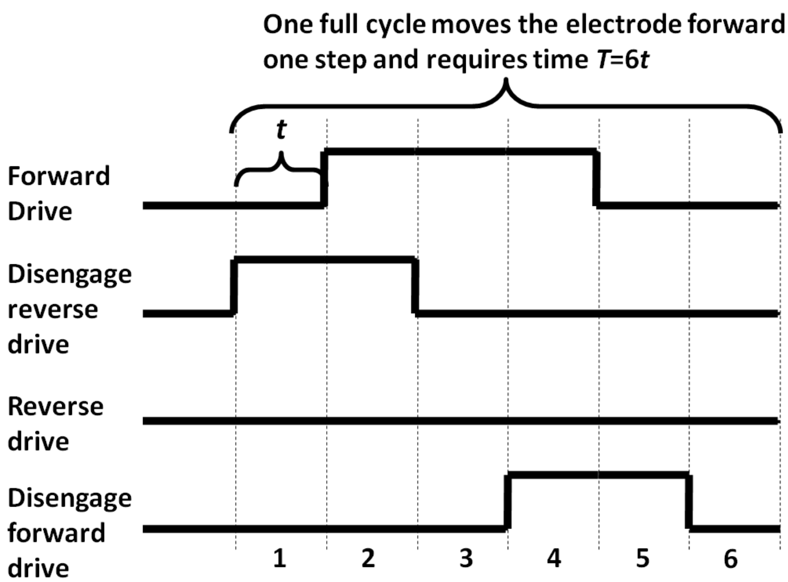

Figure 4: Waveforms for Forward Actuation [18]

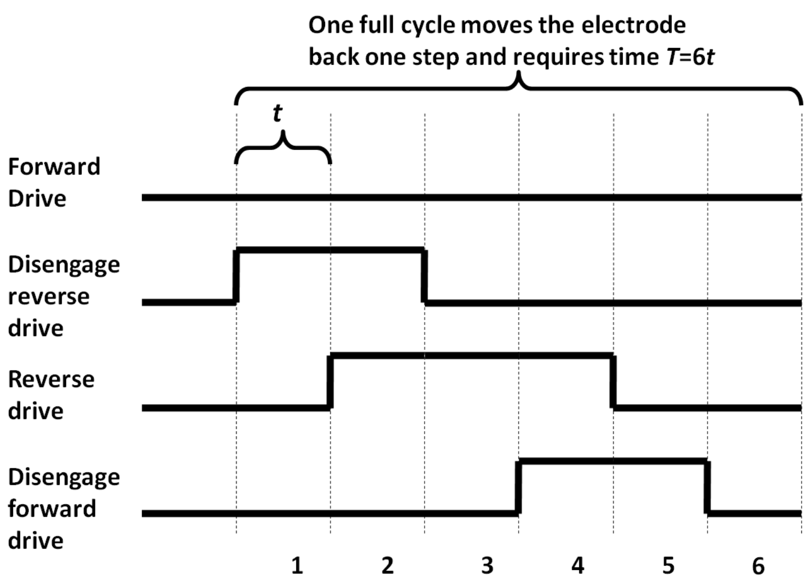

Figure 5: Waveforms for Backward Actuation [18]

As we can see, the forward and reverse driving signals share the same two waveforms: disengage reverse drive and disengage forward dive. However, they have one more separate waveform, which means the system requires four waveforms in total.

One full cycle or "click" of actuator requires time T=6×t.

2.5. 555 Timer and Demultiplexer

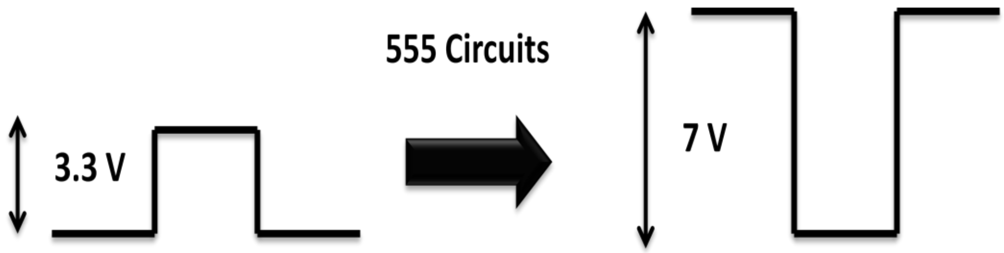

Figure 6: Schematic of Function of 555 Timers

As shown in Figure 6, the 555 timers act like an amplifier in the system, except they will reverse the up level and low level.

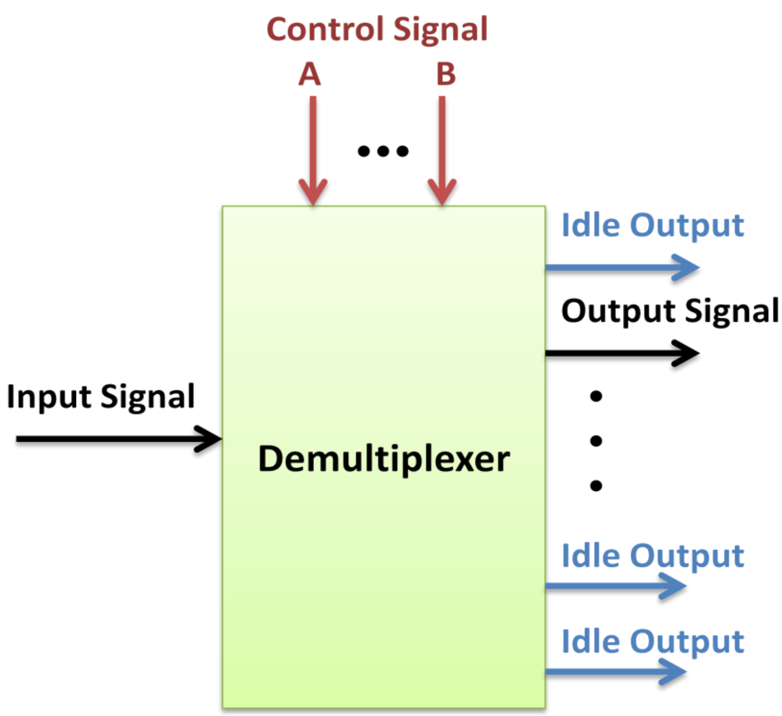

Figure 7: Schematic of Function of Demultiplexer

As shown in Figure 8, the control signal decides which output channel is selected, and the output channel will output exact the same signal as input. Other output channels are idle at the same time.

Table 2: 1 to 4 Truth Table of Demultiplexer

Control input pins (A••B)	Output	Microelectrode selection
0000	o_0	1
0001	o_1	2
0010	o_2	3
...	...	...

Table 2 shows the truth table of the inputs and outputs of the demultiplexer.

2.6. Breadboard Circuits Test of Generating Control Waveforms for Moving Microelectrodes

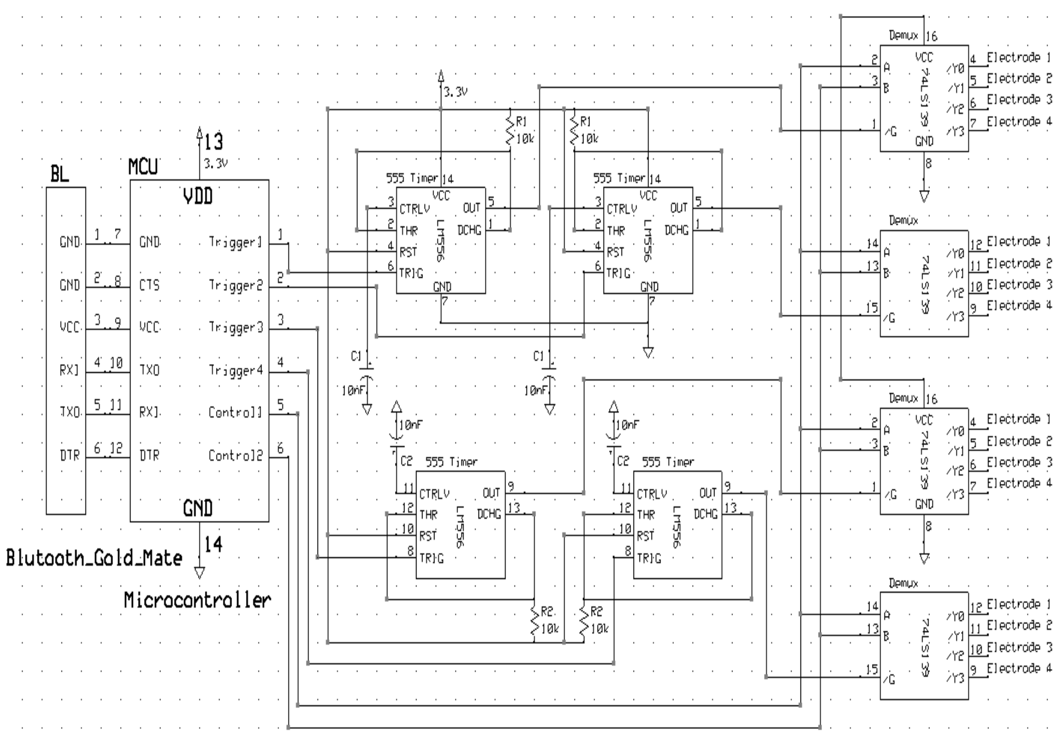

Figure 8: Schematic Design for Waveform Generation of Movable MEMS

The circuit is designed for wirelessly controlling 3 microelectrodes. As Figure 6 described, the Bluetooth module waits for the trigger signal sent from external computer. The microcontroller generates the four corresponding movement waveforms when it gets trigger signal from Bluetooth module. With two dual 555 timers and two dual 1-4 demultiplexer, the microactuators in MEMS can receive the waveforms with high enough voltages and currents to drive the microelectrodes. The waveforms generated by this circuit are tabulated in Chapter 3.

2.7. System Architecture

Due to size limitation, the complete wireless transmitting and control headstage package should be as small as possible. They will be mounted on the rodents' head or placed as a backpack. Hence, based on the successful breadboard testing, the 3D System-On-Package design was adapted. Instead of the normal 2D PCB design, multiple much smaller size PCBs are being stacked up together using wire bonding, which decreases the perimeter in a considerable way.

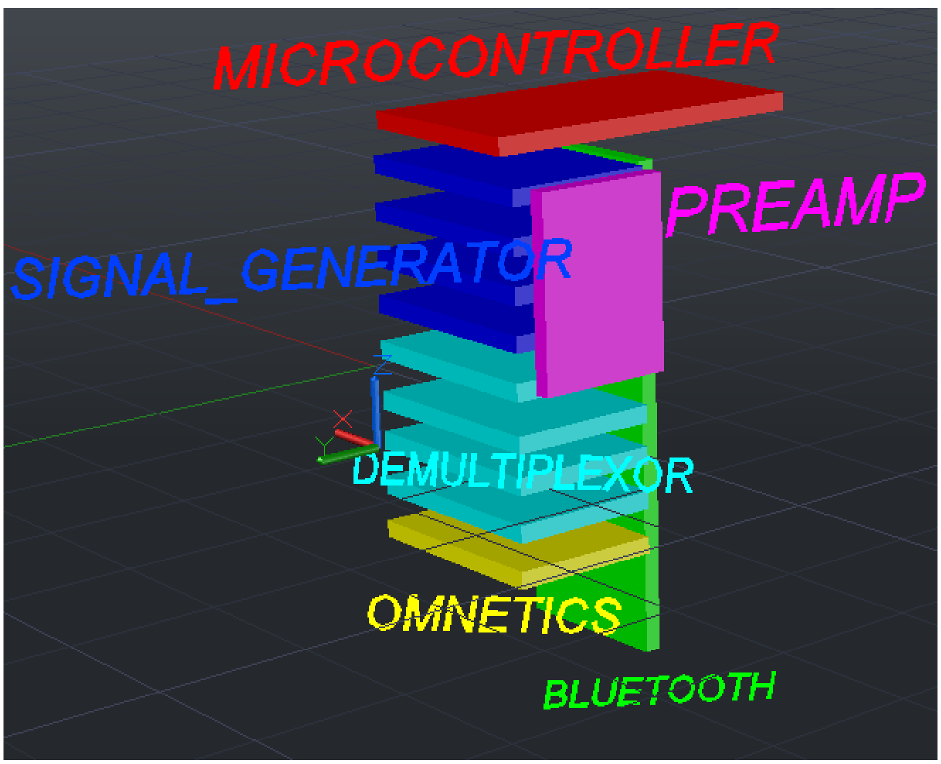

Figure 9: System-on-Package Demo **Error! Reference source not found.**

As in the Figure 7, the top red layer is the microcontroller; the four blue layers are 555 timers, resistors and capacitors; the four cyan layers are four 1-16 quad-demultiplexers; the bottom yellow layer is the Omnetics PCB; the side green vertical PCB is the Bluetooth module and the small purple one is the pre-amplifier PCB.

The anticipated size of the complete 3D package is around $1.3 \times 0.55 \times 1.7$".

2.8. Challenges and Approaches

2.8.1. The need for 555 Timers and Demultiplexers

There two kinds of control signal for the movement of a microelectrode. One is the microelectrode selection signal; the number of bits of this signal depends on the number of microelectrodes. The other one is the forward or backward movement control signal, which needs four bits to generate four waveforms. The number of digital output pins of the microcontroller is limited; hence, for controlling the movement of multiple microelectrodes, a one-to-many mapping strategy is needed. Demultiplexers are used for this purpose. For the demultiplexers, the control input is the selection signal from microcontroller, which decides which microelectrode is selected. A demux with one input and sixteen outputs and four control signals is used in the design, and it is capable of selecting for a maximum of sixteen microelectrodes.

The output pins of the microcontroller, Arduino Pro-mini, have an output voltage of 3.3 volts, which is not high enough to drive the microactuator to move the microelectrodes. The microactuator needs at least 7 volts for maximum displacement. So the 555 timers are used as an amplifier to provide higher voltages and current drives to the microactuator.

2.8.2. Data Transfer Rate Limitation

In the original design the ADC on amplifier board has a sampling rate of 25 KHz per channel and resolution of 16 bit/sample, then the data transfer rate needs to be at least 25KHz * 16 bit/sample = 400 Kbit/s per channel, which is beyond the maximum data transfer rate of Bluetooth (115.2 Kbit/s). Hence, it is reasonable to transfer the neural data of just one channel. Due to lower sampling rate and resolution, the on-chip ADC of the microcontroller has been chosen. With the sampling rate of 10 KHz and the ADC resolution of 10 bit/sample, it requires the data transfer speed at 10 KHz * 10 bit/sample = 100 Kbit/s, which is less than 115.2 Kbits/second.

CHAPTER 3

DATA ANALYSIS AND RESULTS

3.1. Breadboard Test

3.1.1. Microcontrollers' Movement Control Waveforms with Cable Connection

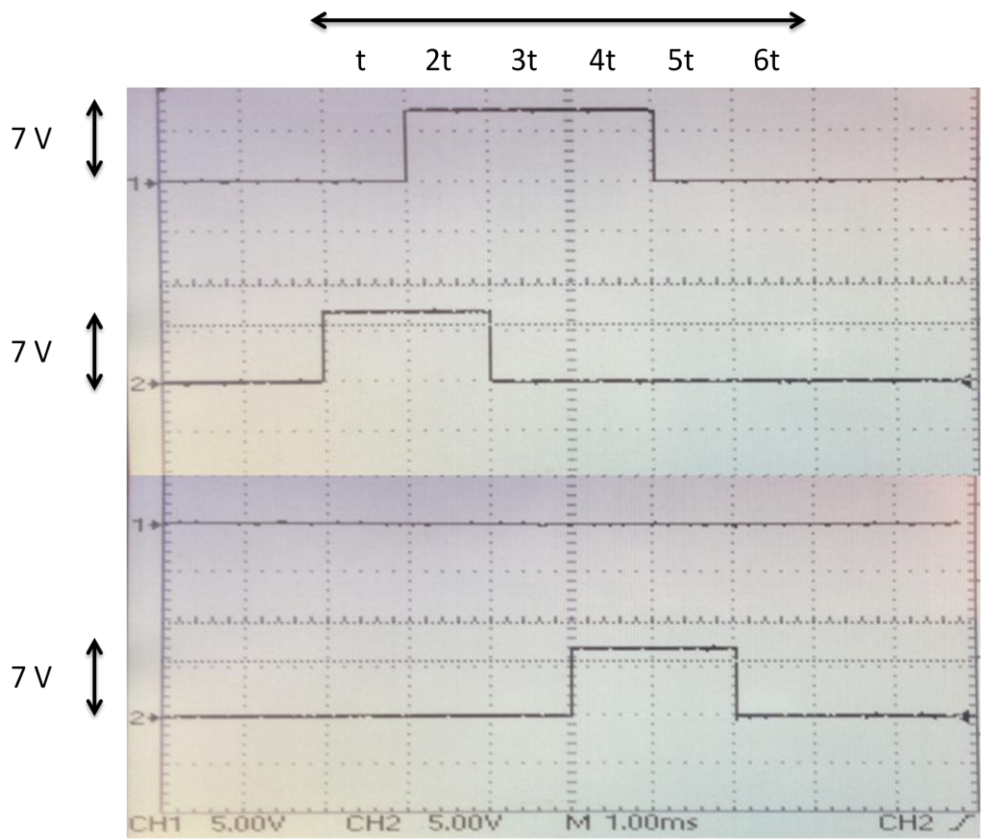

Figure 10: Screen Capture of Waveforms for Forward Actuation (Cable)

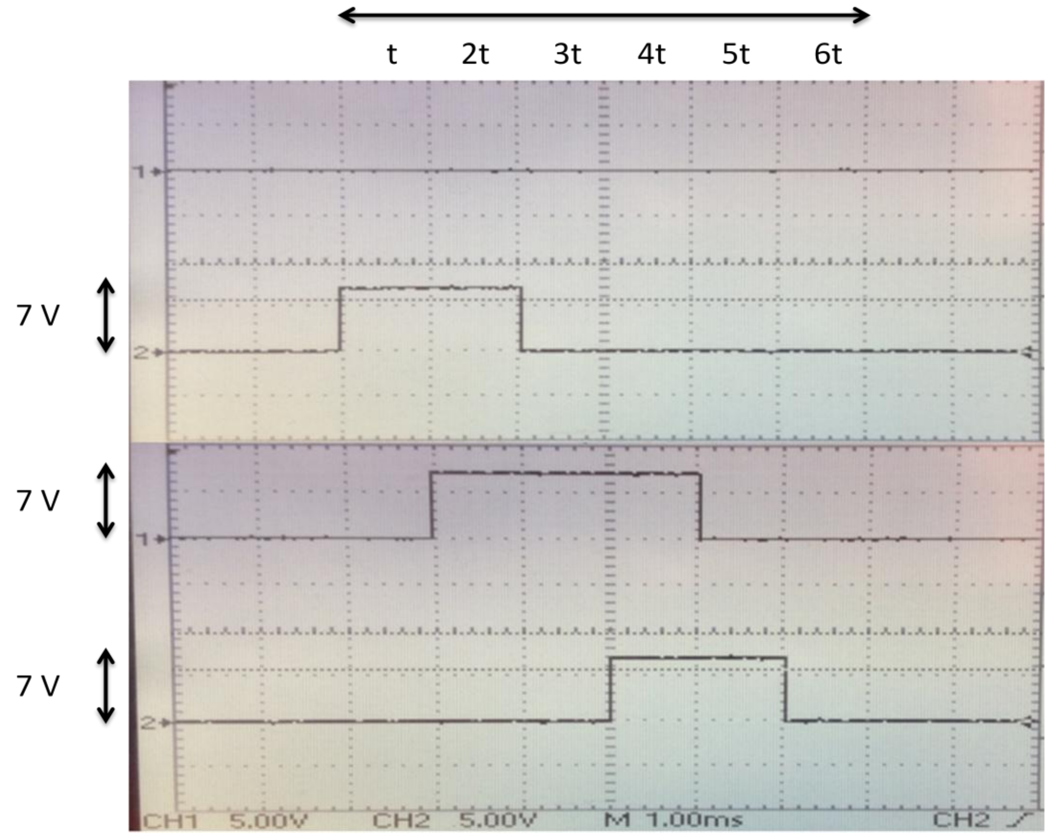

Figure 11: Screen Capture of Waveforms for Backward Actuation (Cable)

An oscilloscope was used to test the control signal. The waveforms shown in Figure 10 and Figure 11 are the waveforms needed for driving microelectrodes forward and backward. All output waveforms have the up level of 7 volts, which are linear to the voltage of power supply. Period t is set as 1 ms during the test.

3.1.2. Microcontrollers' Movement Control Signal with Bluetooth Connection

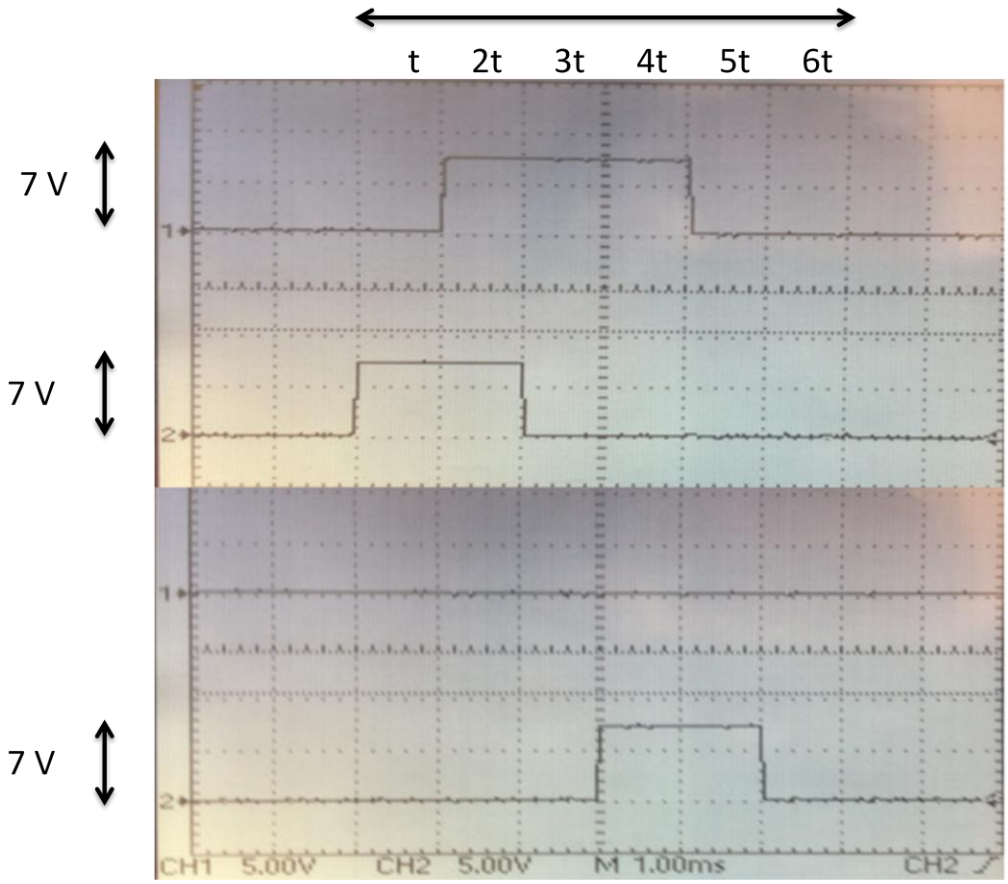

Figure 12: Screen Capture of Waveforms for Forward Actuation (Bluetooth)

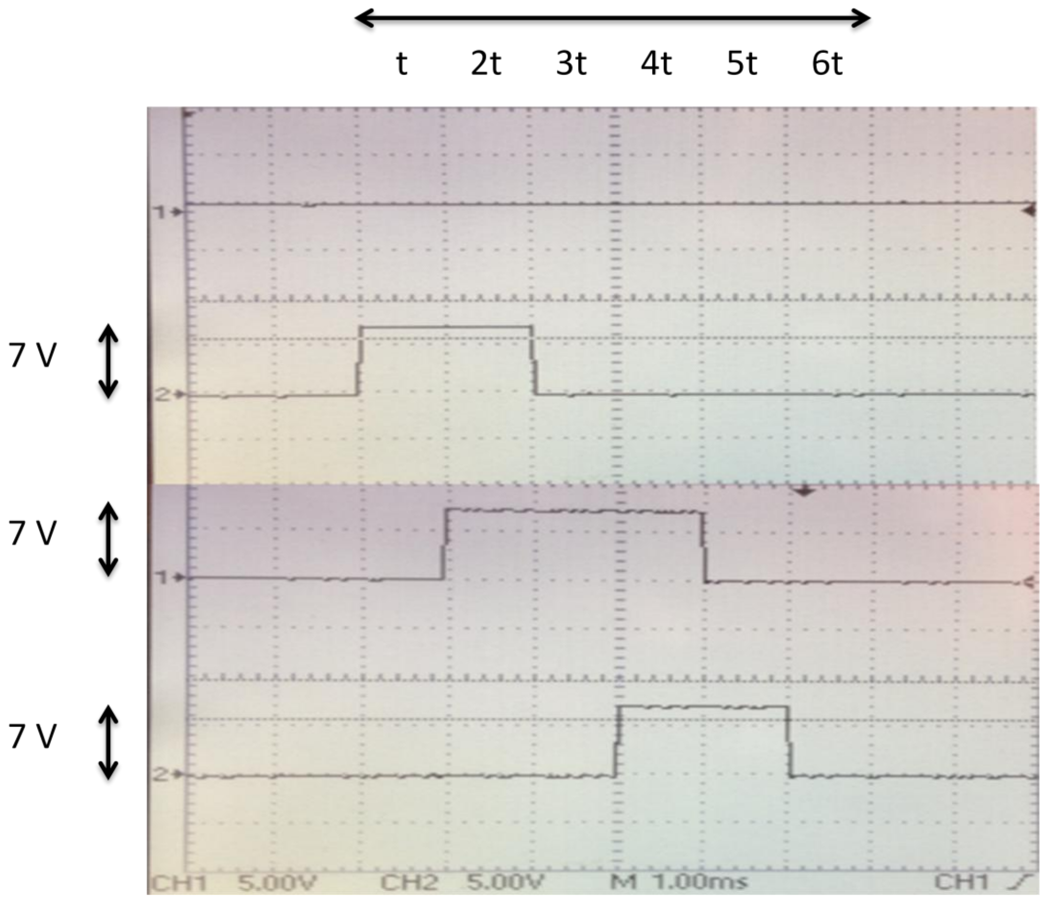

Figure 13: Capture of Waveforms for Backward Actuation (Bluetooth)

Figure 12 and Figure 13 shows that waveforms generated by the trigger signal sent through Bluetooth link are the same as cable connection. The Bluetooth connection works properly for transmitting wireless movement signal.

3.2. 3D System-On-Package Test

The complete 3D System consists of multiple layers of PCBs, which are sandwiched by electrical interconnects.

3.2.1. Packaging Technique and Equipment

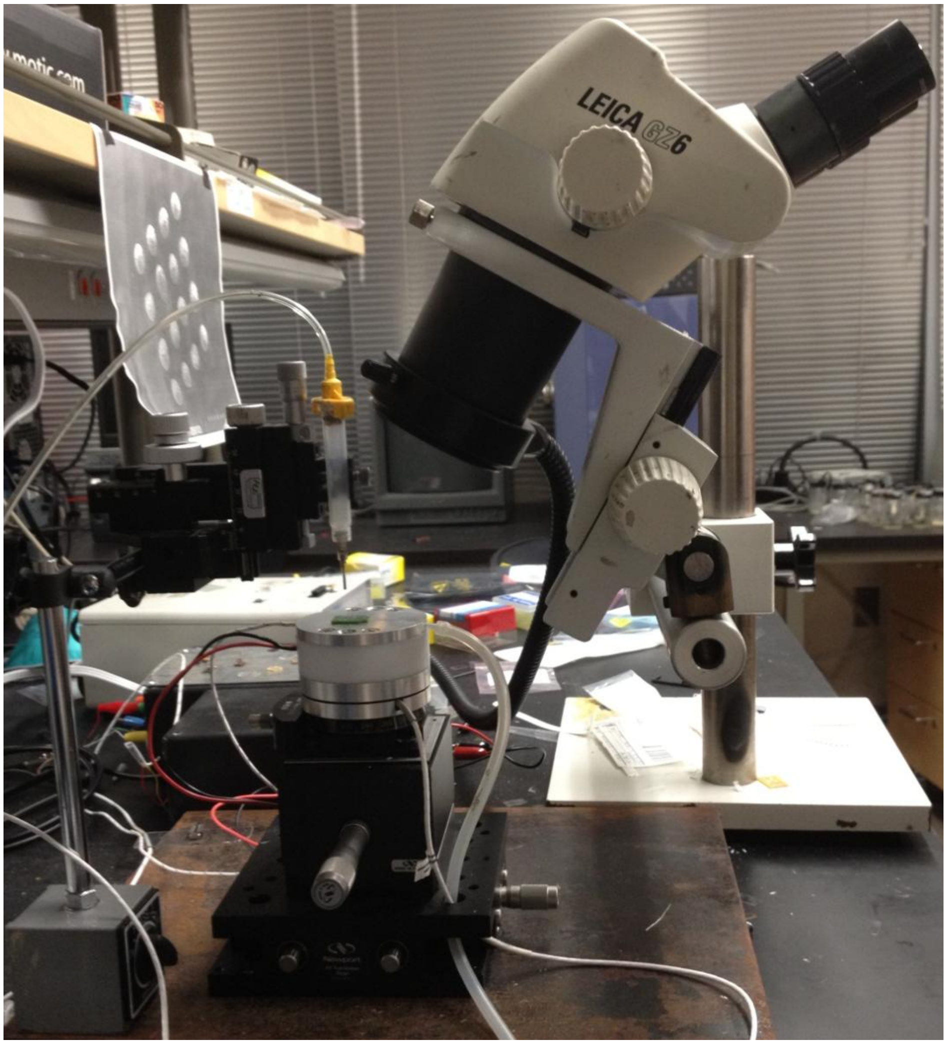

Figure 14: Photograph of Custom-Built Packaging Platform

Printed circuit boards and 3D stacked packaging were done on a custom-built platform for solder paste dispensing as shown in Figure 14.

3.2.2. Pre-Amplifier PCB

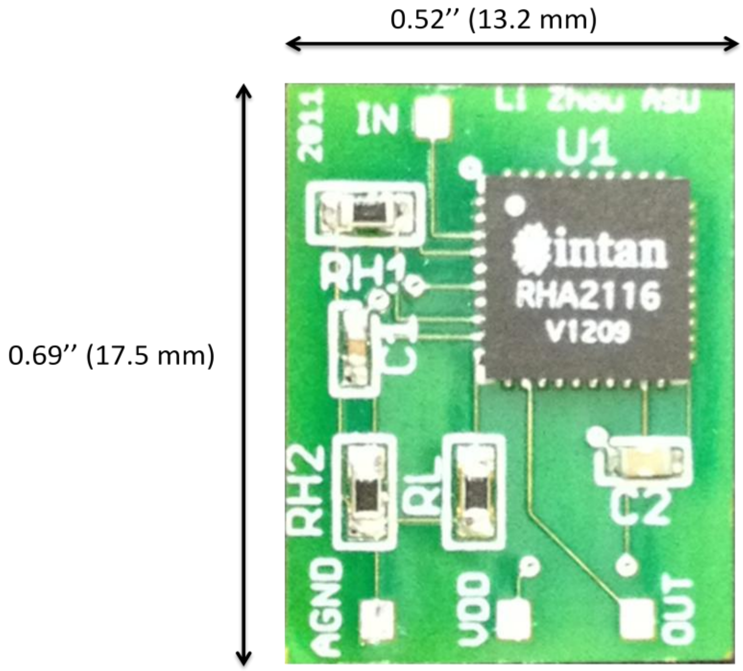

Figure 15: Fully packaged printed circuit board with Intan pre-amplifier. Input connects to the electrodes through Omnetics and wire connection. Output connects to the input of microcontroller.

The name of each component and the input/output pin is shown in Figure 12, the PCB of Pre-Amplifier. A soldering platform shown in Figure 14 was used to solder the tiny components onto the PCB, which required manual experience of surface-mount chip packaging.

3.2.3. 3D System-On-Package Demo

3.2.3.1. Signal Modulation System — 555 timers & Demultiplexers

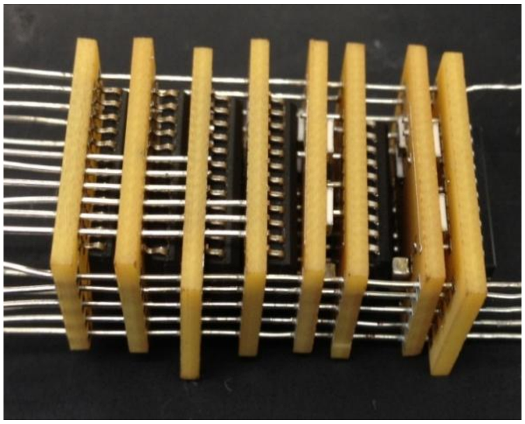

Figure 16: Realization of 3D System-On-Package for Modulating Microelectrodes' Movement Signal

Figure 13 shows the 3D System-On-Package of 555 Timer Circuits and Demultiplexers. The left four layers are four quad 1-16 demultiplexer; the right four layers are 555 timers, resistors and capacitors needed for modulating the waveform amplitudes.

3.2.3.2. Full Movement Signal Generating System

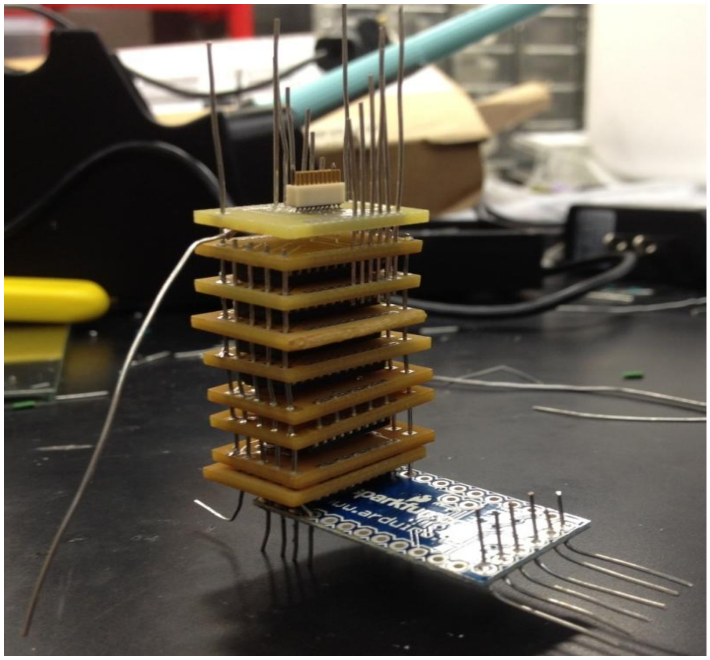

Figure 17: Realization of Full Movement Signal Generating System

As shown in the Figure 14, the top layer is the PCB with an Omnetics™ connector attached, whose male header will connect to the female header on the MEMS chip. The bottom layer is the microcontroller, Pro-Mini from Arduino, which would provide the initial control signal and movement waveforms.

3.2.4. Demultiplexer Selection Result

Table 3: Truth Table of 1 to 16 Demultiplexer

Control input pins $(\mathbf{A_0A_1A_2A_3})$	Output	Microelectrode selection
0000	o_0	1
0001	o_1	2
0010	o_2	3
...	...	...

Table 3 shows the truth table of the inputs and outputs of the demultiplexer in 3D System-On-Package Test.

0000 => microelectrode 1

0001 => microelectrode 2

0010 => microelectrode 3

3.2.5. Microcontrollers' Movement Control Signal with Cable Connection

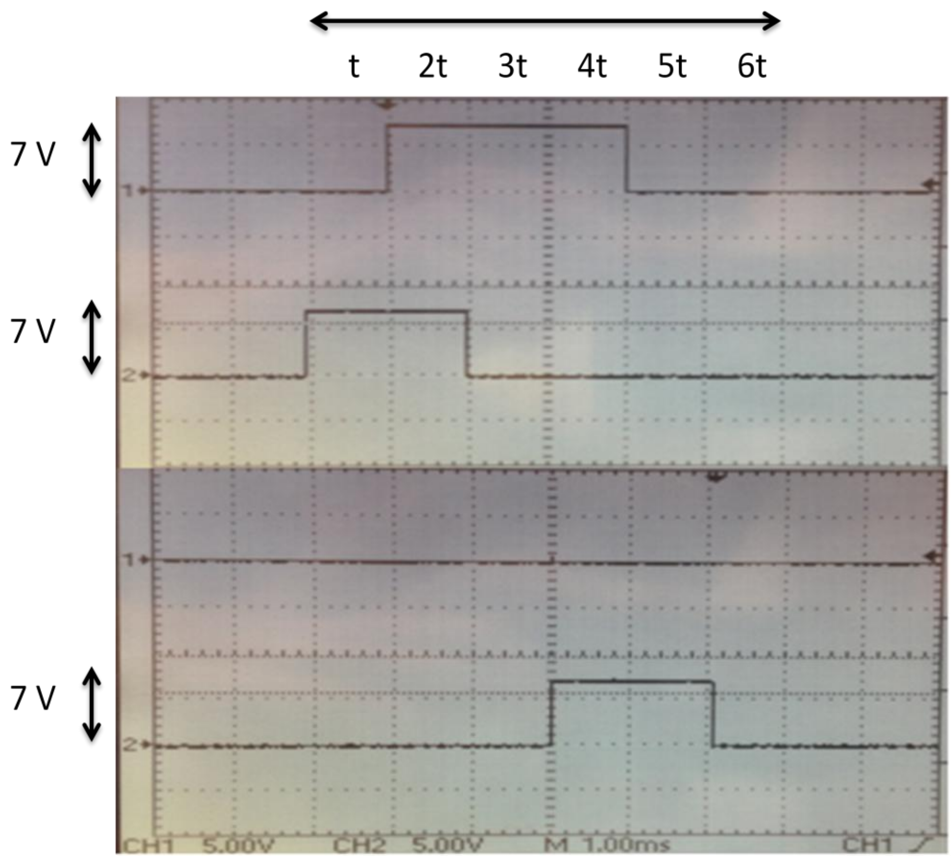

Figure 18: Screen Capture of Waveforms for Forward Actuation (Cable)

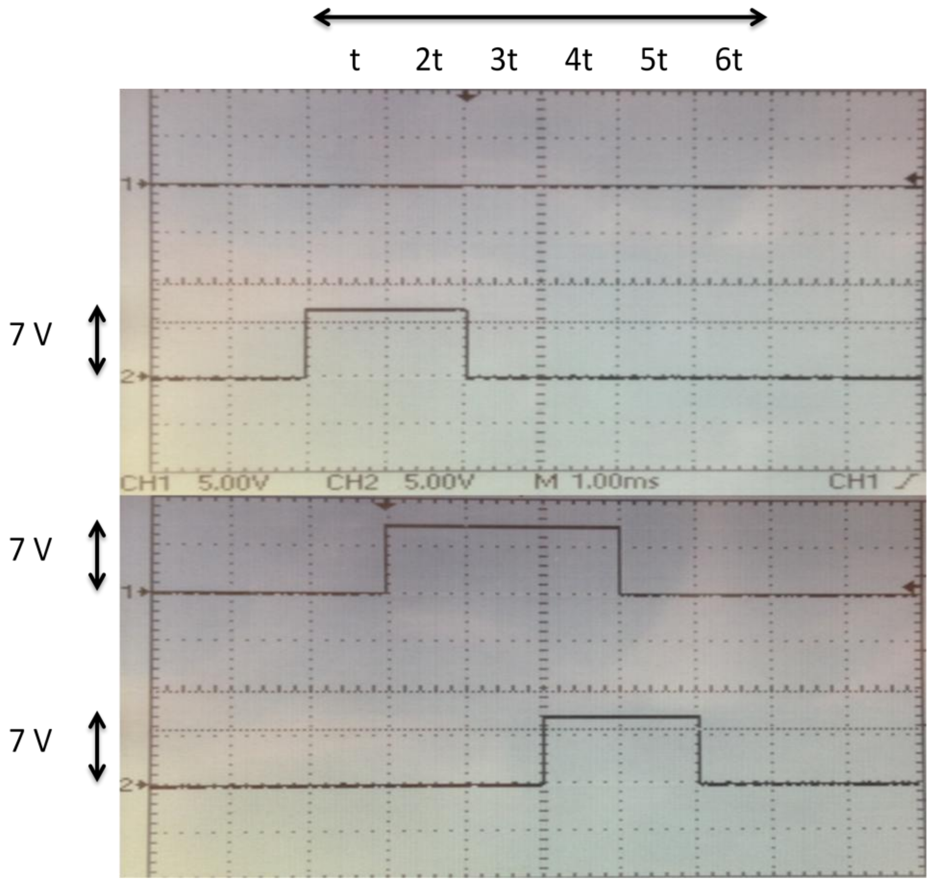

Figure 19: Screen Capture of Waveforms for Backward Actuation (Cable)

As shown in the Figure 18 and Figure 19, the 3D stacked package system for signal generating works successful.

3.2.6. Microcontrollers' Movement Control Signal with Bluetooth Connection

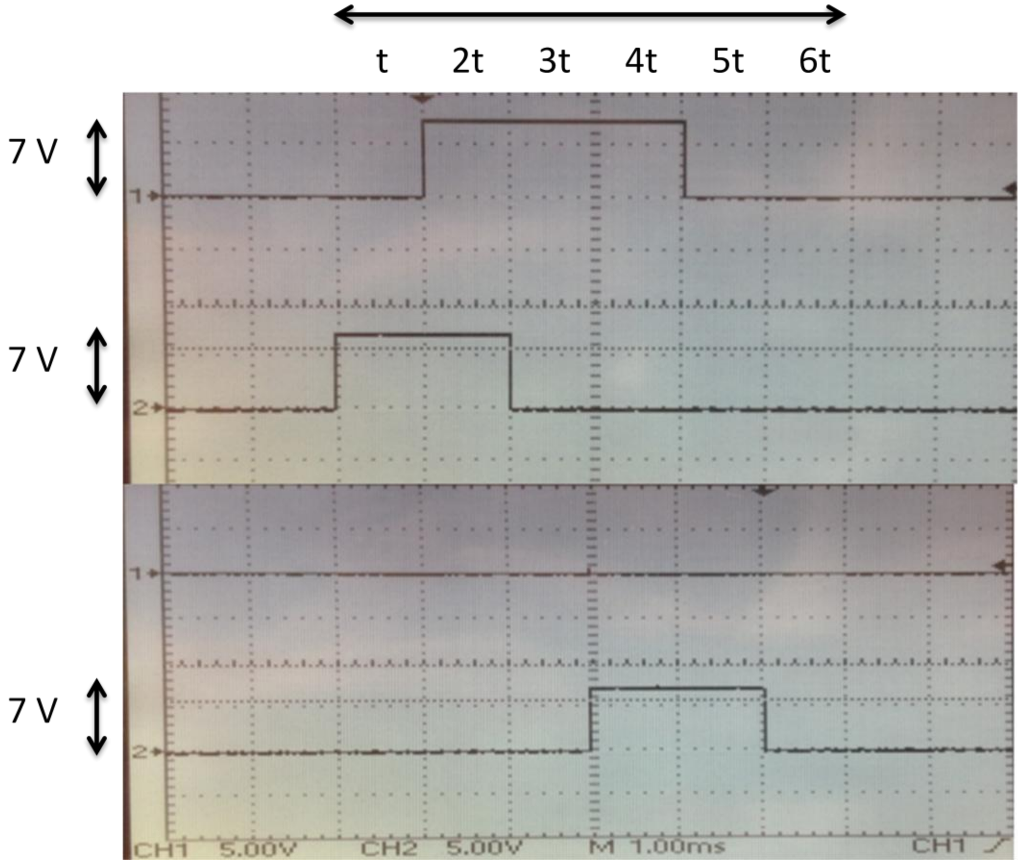

Figure 20: Screen Capture of Waveforms for Forward Actuation (Bluetooth)

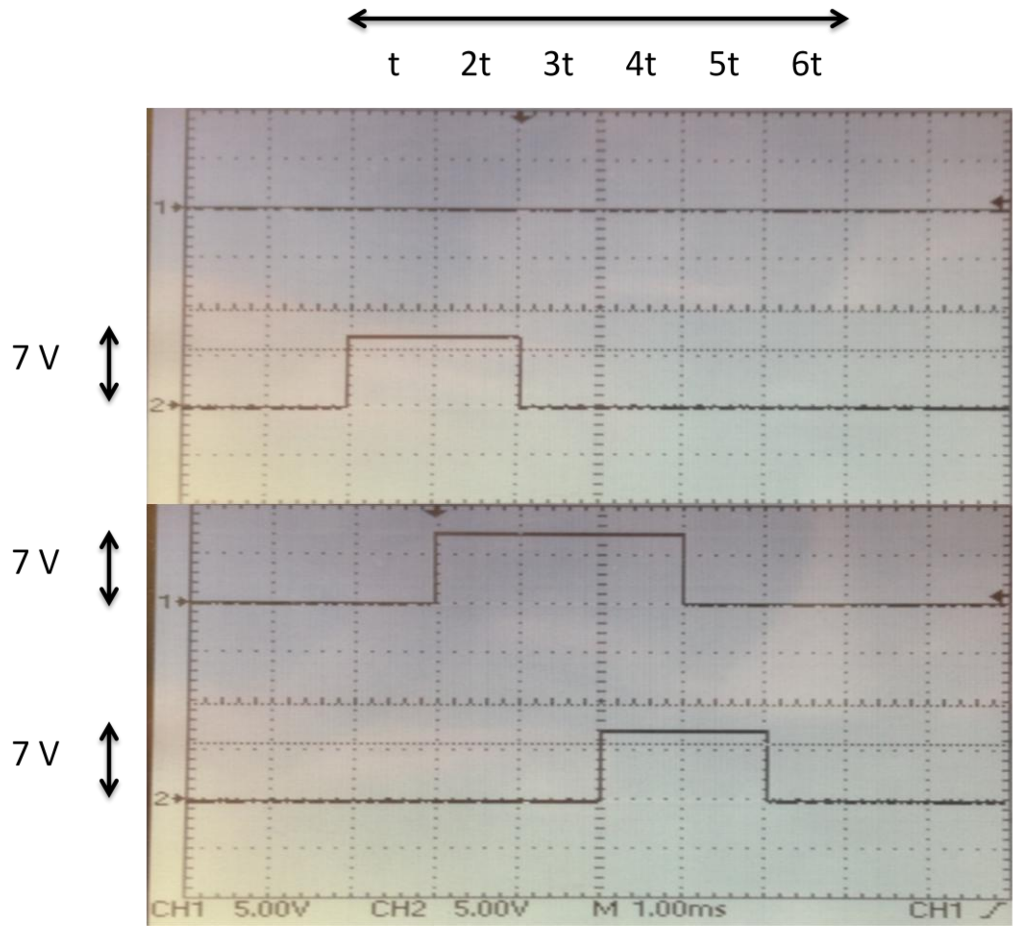

Figure 21: Screen Capture of Waveforms for Forward Actuation (Bluetooth)

As shown in Figure 20 and Figure 21, the Bluetooth link in 3D stacked package system also works properly for transmitting trigger signal from an external computer to the microcontroller.

3.2.7. Microelectrode Movement Result

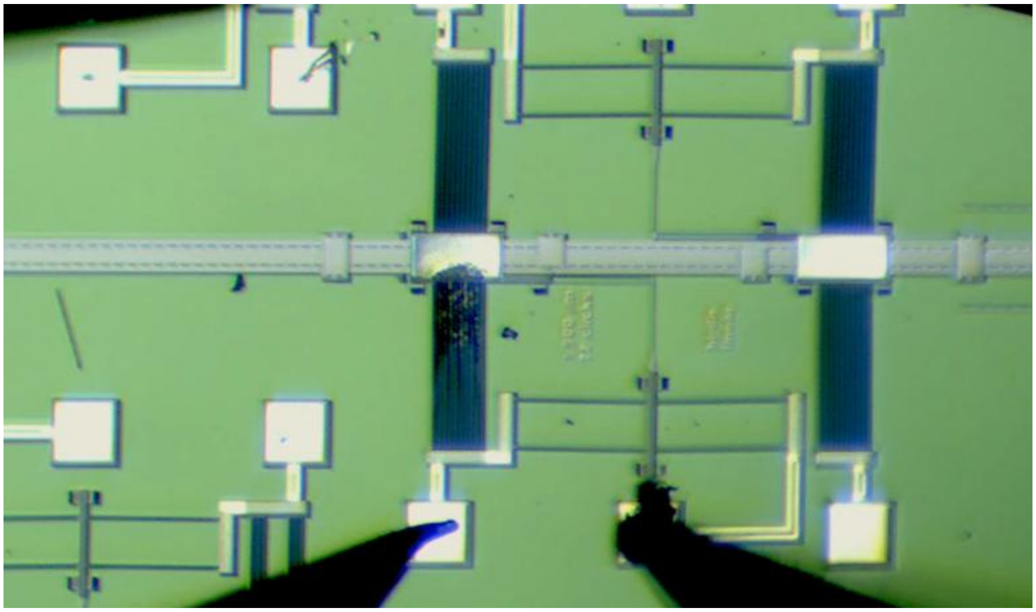

Figure 22: Screen Capture of Microelectrode in Steady Position

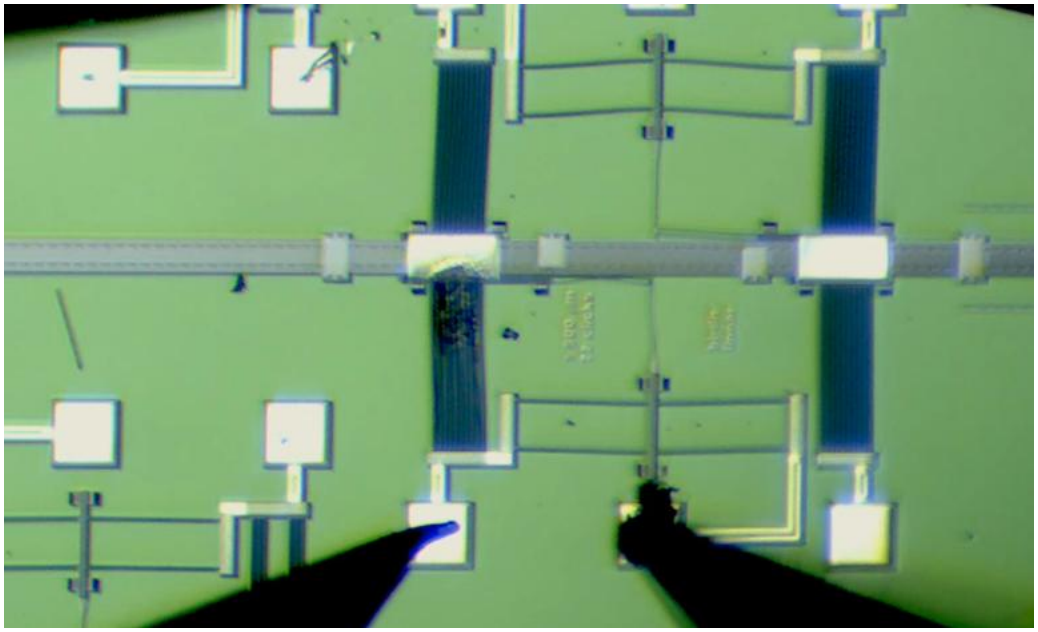

Figure 23: Screen Capture of Microelectrode in Moving Position

As shown in Figure 22 and Figure 23, with the waveforms from package, the microactuator drove the microelectrode perfectly.

3.2.8. Tables of Size and Power Consumption

Table 4: Component Parameter

	Dimension	Average power consumption	Weight
Amplifier	0.52×0.69×0.06" (13.2x17.5x1.5 mm)	Less than 500uW per channel	0.69 g
Bluetooth	1.75x0.65" (1.5x13.2x25.8 mm)	26.4mW (Sleep) 90mW (Active)	3.92 g
Microcontroller	0.7x1.3" (18x33mm)	0.18uW (Power-save) 0.36mW (Active)	2 g
555 Timer Circuit PCB	0.55x0.8"	15mW	1.51 g
Demultiplexer PCB	0.55x0.8"	20mW	1.51 g
Omnetics PCB	0.55x0.8"	N/A	1 g
3D Stacked Package	1.3 x 1.4 x1.75"	N/A	N/A

The complete package was designed to mount on the headstage on the rodent's head. However, the table 4 shows that the package is still too large and too heavy, and it need to be mount as a backpack.

During the neural signal transmission process, microcontroller and pre-amplifier only consume very little power. Most the power consumption is based on the Bluetooth module. Therefore, according to the table 4, a 3.3V 250 mAh coin cell battery (825 mW) with dimension of 20×3.2 mm can power the process of neural signal transmission only for approximate 9 hours (825 mW / 90 mW).

3.3. Electrical Characteristics of Stacked 3D Package

Table 5: Electrical Characteristics

Parameter	Min.	Typ.	Max.	Unit
Supply Voltage (DC)	6.5	7	9.5	V
Output Voltage	6.5	7	9.5	V
Output current (Sending Control waveforms)	70	72	100	mA

Output current (Standby/Idle)	0.27	0.3	0.45	mA

The output parameters are measured from the output pins of demultiplexers. Table 5 shows the current and voltage output parameter. 6.5 volts is the minimum voltage needed for movement waveforms. 9.5 volts is the maximum voltage that the stacked 3D package can generate. Both voltage and current meet the requirement to make microactuator drive the microelectrodes to move.

3.4. Wireless Neural Data Acquisition Testing

Since the moveable microelectrodes MEMS is not finished yet, sinusoid dummy signals are used to test the wireless neural data transmission. Furthermore, the microcontroller cannot read the negative voltage; an offset voltage was added to the input. Tested with dummy signal that is provided by a waveform generator.

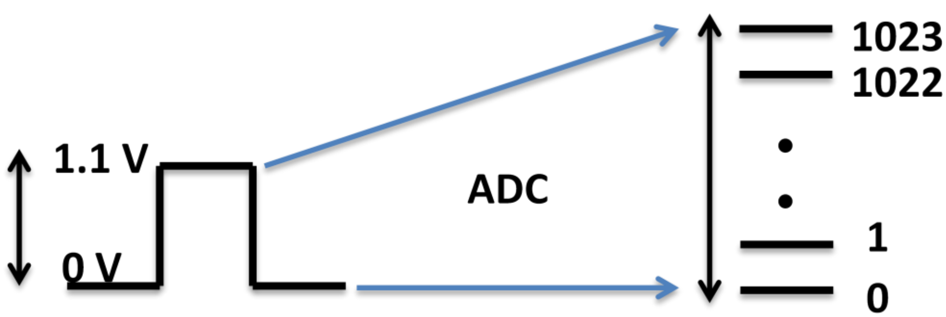

Figure 24 Operating Principle of ADC

The microcontroller (Arduino Pro-Mini) has an on-chip analog-to-digital converter that reads this changing voltage and converts it to a number between 0 and 1023 (10 bits resolution). When there are 0 volts going to the pin, and the input value is 0. When there are 1.1 volts going to the pin and the input value is 1023. In between, it returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin. This means that it will map input voltages between 0 and 1.1 volts into integer values between 0 and 1023. This yields a resolution between readings of 1.1 volts / 1024 units or 0.0011 volts (1.1 mV) per unit [20]. Hence, the actual values are needed to be converted by being divided by 1024 and then times 1.1 V.

3.4.1. Result Comparison of Different Amplitude

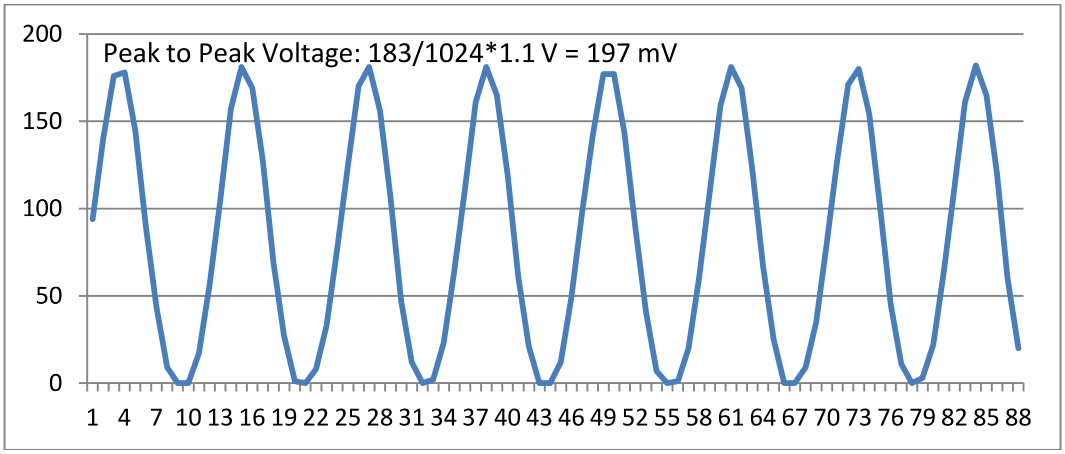

Figure 25: Result with SineWave_100Hz_100mV Input

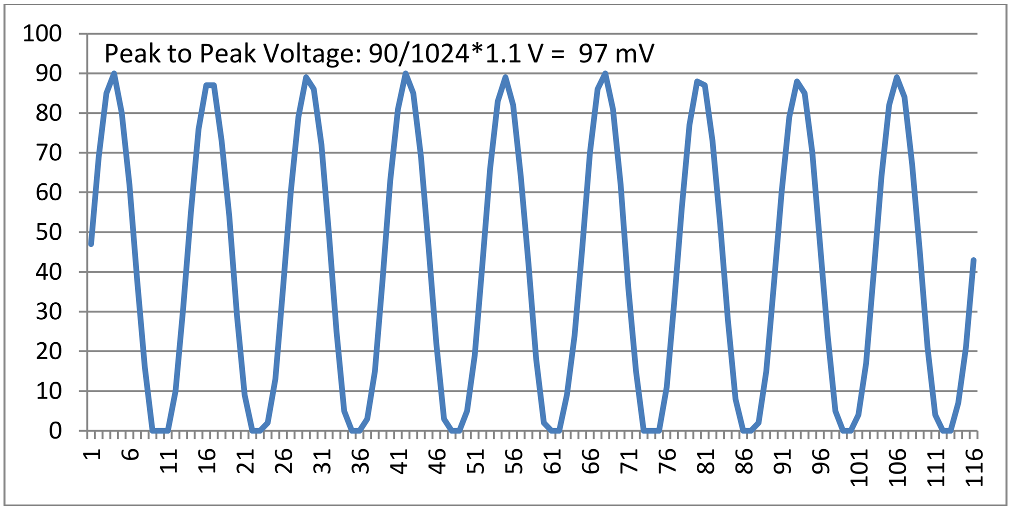

Figure 26: Result with SineWave_100Hz_50mV Input

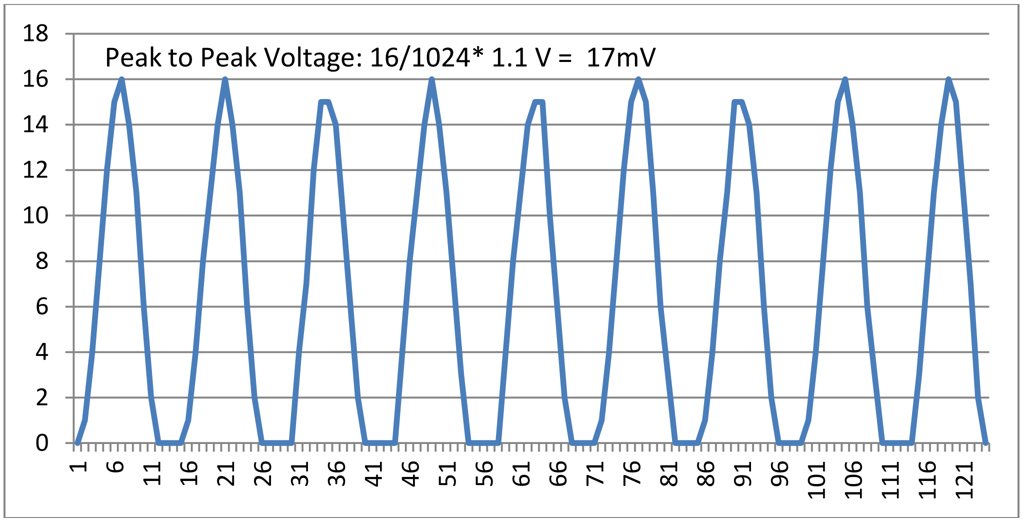

Figure 27: Result with SineWave_100Hz_10mV Input

The horizontal ordinate is the number of sampled neural data. Figure 25, Figure 26 and Figure 27 shows that at the same frequency of 100 Hz, the quality of received signals decreased with the input voltages decreasing. The Peak-to-Peak values of the results are less than inputs, which may be caused by noise interference and low sampling accuracy. However, their qualities are acceptable.

3.4.2. Result Comparison of Different Frequency

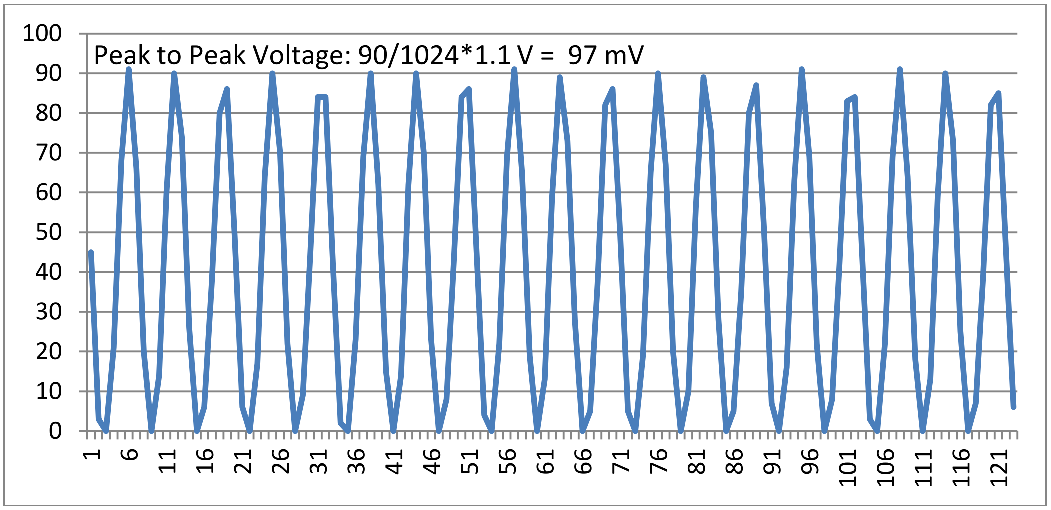

Figure 28: Result with SineWave_200Hz_50mV Input

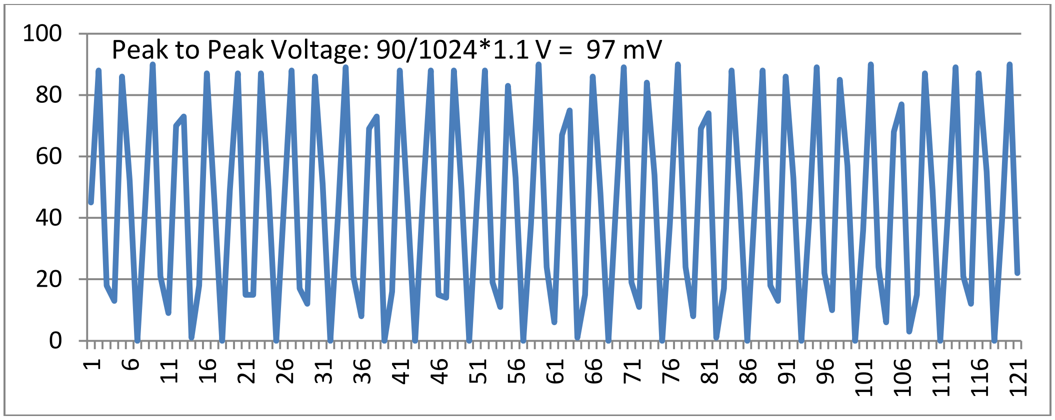

Figure 29: Result with SineWave_350Hz_50mV Input

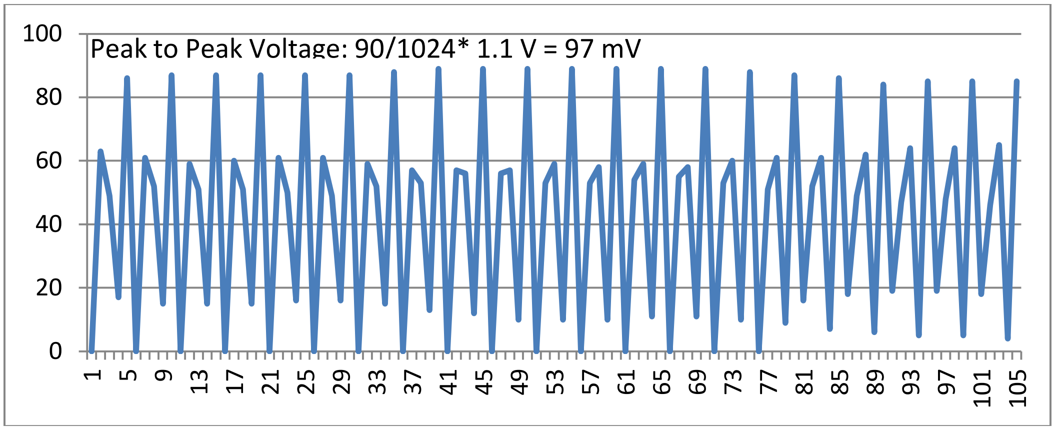

Figure 30: Result with SineWave_500Hz_50mV Input

The Nyquist rate (f_N) is the minimum sampling rate required to avoid aliasing, equal to twice the highest frequency (B) contained within the signal.

$$f_N = 2B \qquad \text{Formula 1}$$

To avoid aliasing, the sampling rate (f_S) must exceed the Nyquist rate (f_N)

$$f_S > f_N \qquad \text{Formula 2}$$

Normally the sampling rate should be at 3 to 4 times the Nyquist rate to get better output signals.

$$f_S > 4 \times f_N \qquad \text{Formula 3}$$

Total sampled data numbers of 11249, 74532 and 361958 were stored in the computer when the wireless transmissions were tested at durations of 10s, 60s and 300s.

$$f_{S1} = 11249/10 = 1125Hz$$

$$f_{S2} = 74532/60 = 1242Hz$$

$$f_{S3} = 361958/300 = 1206Hz$$

The three formulas show that the sampling frequency approaches 1.2 kHz.

According to formula 1 and formula 3, the signal with frequency below $B = 1200/4/2 = 150Hz$ would have a good quality.

As shown in the Figure 26, Figure 28, Figure 29 and Figure 30, the result has visible aliasing started at the frequency of 200 Hz, and serious aliasing occurred at frequency of 350 Hz and 500 kHz.

It takes about 100 microseconds (0.0001 s) for the microcontroller to read an analog input, so the maximum reading rate is about 10,000 times a second [20]. Therefore, the sampling rate of the microcontroller should be 10 kHz. However, the process of converting analog signal to digital signal may cost some time and delay the sampling rate, which leads to the fact that the actual sampling rate is around only 1.2 kHz.

3.5. Close Loop Control Testing

Since there is no neural signal to test, a dummy signal provided by a signal generator is used. The voltage of the dummy signal was increased/decreased manually to imitate the input neural signal. When the voltage is steady, all the sampled value are smaller than or equal to the threshold (sum of mean plus 3 times the standard deviation), which indicates the quality of the input signal is not acceptable. Then the system would automatically send the movement waveform to the MEMS to move the microelectrode. (For convenient observation, here I just changed the code to turn on a red LED instead).

Moreover, when the voltage was increased, the input samples became larger than the threshold. When there were more than 20 points of sampled data being larger than the threshold, which indicates that the quality of the signal is acceptable for further analysis, instead of sending any movement waveforms, the microcontroller transferred the neural data. The process is similar to the system 1, which was introduced in introduction.

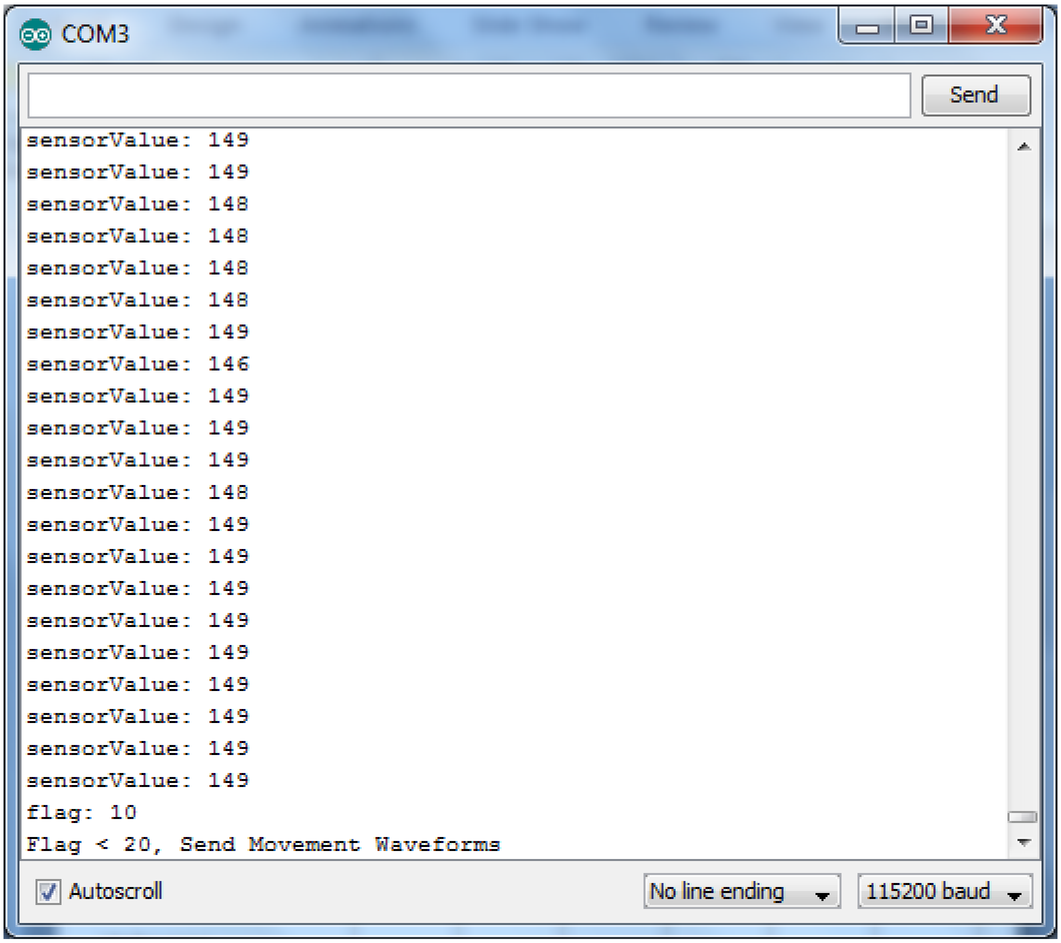

Figure 31: Screen Capture of Bad Signal Quality

Figure 31 shows the result of bad signal quality, and the microcontroller will generate the movement waveforms.

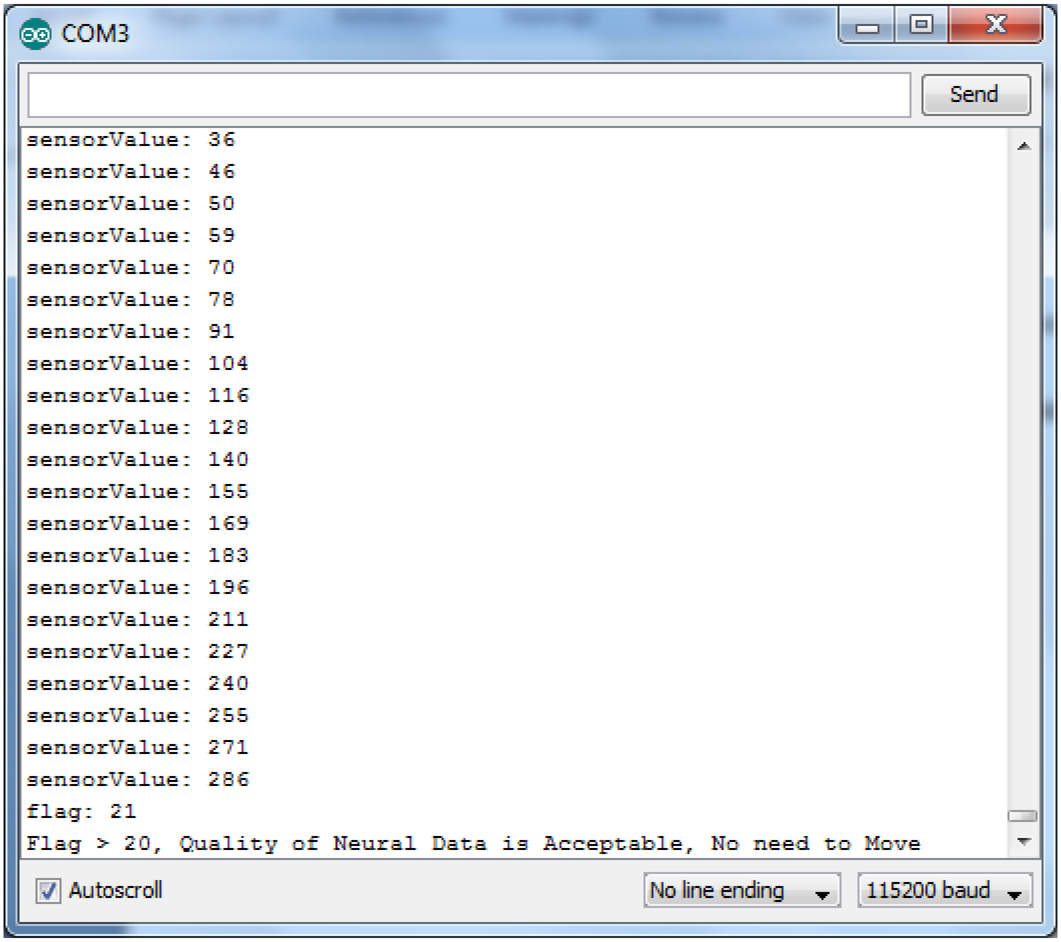

Figure 32: Screen Capture of Good Signal Quality

Figure 32 shows the quality of the input signal is acceptable, and the system will start to transfer data.

CHAPTER 4

SUMMARY AND DISCUSSION

4.1. Performance and Stability Analysis

4.1.1. Wireless Movable Microelectrode Trigger Signal's Transmitting and Corresponding Control Signal's Generating

For transmitting control signals for movable microelectrodes, the fully packaged system was tested with Bluetooth link for different ranges. It turns out the Bluetooth communication link worked properly for at least a 15 meters range, which is the longest distance from the workspace to the door. That distance is long enough for transmitting control signal wirelessly for such implantable applications.

From the testing results above, the Bluetooth communication module worked very successfully for transmitting control signals. However, there were noise issues during the testing process. The breadboard-based Bluetooth wireless transmission system is very stable for signal generating and transferring, while the 3D stacked package system is not stable enough, which may be due to the PCB design is not robust enough to shield the outside interference. There are two possible reasons for non-robust signal transmission: one is the exposed wire connections due to the multiple stacking of PCBs brings about a lot of noise and interference; the other one is that the antenna of the Bluetooth module may also draw noise into the system. Hence, a grounded metal box was added to shield the outside noise. Further testing showed the package had reduced outside noise interference during signal transmission.

4.1.2. Wireless Neural Signal Acquisition

Wireless neural signal transmission performance was tested with dummy signals for different amplitudes and different frequencies. As the results shown above, the

Bluetooth link works fine for transmitting the neural data with the speed below 115,200 bps, which avoids all the disadvantages that cable connection causes. However, even though the accuracy of the microcontroller is fine for the neural signal sampling, the sampling rate of on-chip ADC is not acceptable for transmitting neural signal.

A typical neuron generates 10-100 spikes per second when active. Resting or "spontaneous" activity of neurons ranges up to 1-10 spikes per second. While the data sampling rate of the on board ADC is 10 KHz. Furthermore, normally each spike is less than a millisecond in duration, so theoretically each spike could only have less than 10 samples with current microcontroller. In fact, the sampling rate is around 1.2 kHz, which is even smaller than 10 KHz. It is not acceptable for neural signal transmission and analysis.

4.2. Future Work and Improvement

The low voltage level output waveforms of microcontroller need to be amplified before sending to the MEMS chip. If there is a MCU with output voltage reaching 7 volts, there is no need of 555 timers circuits to modify the movement waveforms.

Furthermore, there are both negative and positive voltages of neural signal, while the microcontroller, Arduino Pro-Mini, cannot read negative voltages. It is necessary to add an offset voltage on the neural signal to realize the complete signal transmission, which will add more noise to the neural signal.

So in order to obtain the signals with better quality, a new MCU with the capability to read both positive and negative analog input is needed. In addition, to acquire a better-reconstructed neural signal after the transmission, the MCU should have an on-chip ADC with higher sampling rate (>20 kHz) and higher resolution (>16 bits).

The pre-amplifier PCB designed based on the Intan amplifier could only amplify the neural signals for 200 times, while as requirement the amplified signals are still too low for further processing and analyzing. Therefore, a second stage amplifier may be added after the pre-amplifier to further amplify the neural signal.

Data transfer rate is relatively low for neural data transmission. It is the most important thing to find out or design a wireless module with a proper size to satisfy the higher data transfer rate, especially for multiple channels' signal recording, which is the trend right now.

REFERENCES

[1] Nathan M. Neihart, and Reid R. Harrison*, "Micropower Circuits for Bidirectional Wireless Telemetry in Neural Recording Applications", IEEE Transactions on Biomedical Engineering, VOL. 52, NO. 11, NOVEMBER 2005.

[2] Reid R. Harrison, Ryan J. Kier, Cynthia A. Chestek, Vikash Gilja, Paul Nuyujukian, Stephen Ryu, Bradley Greger, Florian Solzbacher, and Krishna V. Shenoy, " Wireless Neural Recording With Single Low-Power Integrated Circuit ", IEEE Transactions on Neural Systems and Rehabilitation Engineering, VOL. 17, NO. 4, AUGUST 2009.

[3] Reid R. Harrison, and Cameron Charles, "A Low-Power Low-Noise CMOS Amplifier for Neural Recording Applications" IEEE Journal of Solid-State Circuits, Vol. 38, No. 6, June 2003.

[4] William M. Reichert, "Indewelling Neural Implants, Strategies for Contending with the In Vivo Environment," Frontiers in Neuroengineering, Duke University, North Carolina, Boca Raton (FL): CRC Press, 2008.

[5] Tayfun Akin, Khalil Najafi, and Robert M. Bradley, "A Wireless Implantable Multichannel Digital Neural Recording System for a Micromachined Sieve Electrode ", IEEE Journal of Solid-State Circuits, VOL. 33, NO. 1, JANUARY 1998.

[6] J Muthuswamy, M Okandan, T Jain*, and A Gilletti, "Electrostatic microactuators for precise positioning of Neural microelectrodes, "IEEE Trans Biomed Eng, 52:1748-1755, 2005.

[7] J Muthuswamy, S Anand*, A Sridharan*, “Adaptive movable neural interfaces for monitoring single neurons in the brain,” Frontiers in Neuroscience, vol. 5, article 94, 2011 (invited review) Abstract.

[8] N Jackson*, A Sridharan*, S Anand*, M Baker, M Okandan and J Muthuswamy, “Long-term neural recordings using MEMS based moveable microelectrodes in the brain,” Frontiers in Neuroengineering, 3:10. doi: 10.3389/fneng.2010.00010,2010. (article)

[9] J Muthuswamy, M. Okandan, and N. Jackson*, “Surface Micro-Machined Polysilicon Probes for Neurophysiology,” *J. Neurosci. Meth.*, 142(1):45-54, 2005.

[10] J Sutanto*, S Anand*, A Sridharan*, R Korb*, Li Zhou, M Baker, M Okandan and J Muthuswamy, “Packaging and Non-Hermetic Encapsulation Technology for Flip-chip on Implantable MEMS Devices,” (in press) for *IEEE/ASME J Microelectromech Syst*, 2011.

[11] J Sutanto*, S Anand*, C Patel* and J Muthuswamy, “Novel first-level interconnect techniques for flip-chip on MEMS,” (in press) for *IEEE/ASME J Microelectromech Syst*, 2011.34.

[12] Reid R. Harrison, "The Design of Integrated Circuits to Observe Brain Activity", Proceedings of the IEEE, Vol. 96, No. 7, July 2008.

[13] Michael Rizk, Cad A Bossetti, Thomas A Jochum, Stephen H Callender, Miguel A L Nicolelis, Dennis A Turner and Patrick D Wolf, "A fully implantable 96-channel neural data acquisition system" Journal of Neural Engineering, 2009.

[14] Harrison R. R., "The design of integrated circuits to observe brain activity" Proc. IEEE 96 1203-16, 2008.

[15] Reid R. Harrison, Ryan J. Kier, Sohee Kim Loren Rieth, David J. Warren, Noah M. Ledbetter Gregory A. Clark, Florian Solzbacher, Cynthia A. Chestek, Vikash Gilja, Paul Nuyujukian Stephen I. Ryu, Krishna V. Shenoy, "A Wireless Neural Interface for Chronic Recording".

[16] Reid R. Harrison, Paul T. Watkins, Ryan J. Kier, Robert O. Lovejoy, Daniel J. BlackBradley Greger, and Florian Solzbacher, "A Low-Power Integrated Circuit for a Wireless 100-Electrode Neural Recording System", IEEE Journal of Solid-State Circuits, Vol. 42, No. 1, January 2007.

[17] Intan_RHA2000_series_datasheet. Available: http://www.intantech.com/downloads.html

[18] Sindhu Anand[1], Jemmy Sutanto[1], Michael Baker[2], Murat Okandan[2], Jit Muthuswamy[1], " Linear ratcheting microactuators improve performance for MEMS brain implant applications ", unpublished.

[19] Zikai Tee, "3D System-On-Package (SoP) Signal Generator to Control MEMS Movable Microelectrode Arrays," unpublished.

[20] Arduino Website. Available: http://arduino.cc/en/Tutorial/AnalogReadSerial

APPENDIX A

ARDUINO CODE FOR WIRELESS CONTROLLING MOVABLE MICROELECTRODES

```
int pin1 = 13;    // Trigger Signal
int pin2 = 12;    // Trigger Signal
int pin3 = 11;    // Trigger Signal
int pin4 = 10;    // Trigger Signal
int pin5 = 6;      // Select Signal A0
int pin6 = 7;     // Select Signal A1
int pin7 = 8;     // Select Signal A2
int pin8 = 9;     // Select Signal A3

void setup() {
  Serial.begin(115200);
  pinMode(pin1, OUTPUT);
  pinMode(pin2, OUTPUT);
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  pinMode(pin5, OUTPUT);
  pinMode(pin6, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    char value = Serial.read();
    if (value == 'a') {                 //Select Microelectrode 1 and Drive it Forward

      //Select E1    Demux Output 0.    A3A2A1A0 = 0000.  The 555 Timer will reverse the
      //signal, so if we want A3A2A1A0 = 0000, we provide A3A2A1A0 = 1111;

      digitalWrite(pin5, HIGH);
      digitalWrite(pin6, HIGH);
      digitalWrite(pin7, HIGH);
      digitalWrite(pin8, HIGH);

      // Forward Waveform
      digitalWrite(pin1, LOW);
      digitalWrite(pin2, HIGH);
      digitalWrite(pin3, LOW);
      digitalWrite(pin4, LOW);
      delay(1);
      digitalWrite(pin1, HIGH);
      delay(1);
      digitalWrite(pin2, LOW);
      delay(1);
      digitalWrite(pin4, HIGH);
      delay(1);
      digitalWrite(pin1, LOW);
      delay(1);
      digitalWrite(pin4, LOW);
    }

    else if (value == 'b') {                //Select Microelectrode 1 and Drive it Backward

      // Select E1    Demux Output 1. A3A2A1A0 = 0000.  The 555 Timer will reverse the
      //signal, so if we want A3A2A1A0 = 0000, we provide A3A2A1A0 = 1111;

      digitalWrite(pin5, HIGH);
```

```
  digitalWrite(pin6, HIGH);
  digitalWrite(pin7, HIGH);
  digitalWrite(pin8, HIGH);

  //Reverse Waveform
  digitalWrite(pin1, LOW);
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, LOW);
  digitalWrite(pin4, LOW);
  delay(1);
  digitalWrite(pin3, HIGH);
  delay(1);
  digitalWrite(pin2, LOW);
  delay(1);
  digitalWrite(pin4, HIGH);
  delay(1);
  digitalWrite(pin3, LOW);
  delay(1);
  digitalWrite(pin4, LOW);
}


 else if (value == 'c') {            //Select Microelectrode 2 and Drive it Forward

  // Select E2    Demux Output 2. A3A2A1A0 = 0001.  The 555 Timer will reverse the
  //signal, so if we want A3A2A1A0 = 0001, we provide A3A2A1A0 = 1110;

  digitalWrite(pin5, LOW);
  digitalWrite(pin6, HIGH);
  digitalWrite(pin7, HIGH);
  digitalWrite(pin8, HIGH);

  // Forward Waveform
  digitalWrite(pin1, LOW);
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, LOW);
  digitalWrite(pin4, LOW);
  delay(1);
  digitalWrite(pin1, HIGH);
  delay(1);
  digitalWrite(pin2, LOW);
  delay(1);
  digitalWrite(pin4, HIGH);
  delay(1);
  digitalWrite(pin1, LOW);
  delay(1);
  digitalWrite(pin4, LOW);


}

else if (value == 'd') {            //Select Microelectrode 2 and Drive it Backward

  // Select E2    Demux Output 3. A3A2A1A0 = 0001.  The 555 Timer will reverse the
  //signal, so if we want A3A2A1A0 = 0001, we provide A3A2A1A0 = 1110;
```

```
  digitalWrite(pin5, LOW);
  digitalWrite(pin6, HIGH);
  digitalWrite(pin7, HIGH);
  digitalWrite(pin8, HIGH);

  //Reverse Waveform
  digitalWrite(pin1, LOW);
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, LOW);
  digitalWrite(pin4, LOW);
  delay(1);
  digitalWrite(pin3, HIGH);
  delay(1);
  digitalWrite(pin2, LOW);
  delay(1);
  digitalWrite(pin4, HIGH);
  delay(1);
  digitalWrite(pin3, LOW);
  delay(1);
  digitalWrite(pin4, LOW);
}

else if (value == 'e') {          // Select Microelectrode 3 and Drive it Forward

  // Select E3    Demux Output 4. A3A2A1A0 = 0010.  The 555 Timer will reverse the
  //signal, so if we want A3A2A1A0 = 0010, we provide A3A2A1A0 = 1101;

  digitalWrite(pin5, HIGH);
  digitalWrite(pin6, LOW);
  digitalWrite(pin7, HIGH);
  digitalWrite(pin8, HIGH);

  // Forward Waveform
  digitalWrite(pin1, LOW);
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, LOW);
  digitalWrite(pin4, LOW);
  delay(1);
  digitalWrite(pin1, HIGH);
  delay(1);
  digitalWrite(pin2, LOW);
  delay(1);
  digitalWrite(pin4, HIGH);
  delay(1);
  digitalWrite(pin1, LOW);
  delay(1);
  digitalWrite(pin4, LOW);


}

else if (value == 'f') {          // Select Microelectrode 3 and Drive it Backward

  // Select E3    Demux Output 5. A3A2A1A0 = 0010.  The 555 Timer will reverse the
  //signal, so if we want A3A2A1A0 = 0010, we provide A3A2A1A0 = 1101;
```

```
    digitalWrite(pin5, HIGH);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, HIGH);
    digitalWrite(pin8, HIGH);

    //Reverse Waveform
    digitalWrite(pin1, LOW);
    digitalWrite(pin2, HIGH);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, LOW);
    delay(1);
    digitalWrite(pin3, HIGH);
    delay(1);
    digitalWrite(pin2, LOW);
    delay(1);
    digitalWrite(pin4, HIGH);
    delay(1);
    digitalWrite(pin3, LOW);
    delay(1);
    digitalWrite(pin4, LOW);
   }

   else if (value == 'z') {        //Test Connection with on-board LED (pin 13)

    digitalWrite(13, HIGH);
    delay (1000);
    digitalWrite(13, LOW);
   }


   Serial.println(value);    //When microcontroller received the data, return it to computer
                              // and show it on screen to confirm the successful transmission.
  }

}
```

APPENDIX B

ARDUINO CODE FOR WIRELESS NEURAL SIGNAL TRANSMISSION WITH MOVABLE MICROELECTRODES CLOSED LOOP CONTROL

```
#define Analog_In A1
#define SAMPLES 100        // sample numbers for standard deviation

int s_val[SAMPLES];
int flag = 0;                         // Detected spike numbers during data amount of A
int data_amount = 100000; //data_amount is the amount of neural data (A) that determine
                                       //whether the quality of neural data is acceptable.
int senserValue = 0;

float sampleSum = 0;
float meanSample = 0;
float sqDevSum = 0;          // variance
float stDev = 0;                  // standard deviation
float threshold = 0;


int pin1 = 2;    // Trigger Signal
int pin2 = 3;    // Trigger Signal
int pin3 = 4;    // Trigger Signal
int pin4 = 5;    // Trigger Signal
int pin5 = 6;    // Select Signal
int pin6 = 7;    // Select Signal

void setup() {
  Serial.begin(115200);
  //for (int thisPin = 2; thisPin < 8; thisPin++) {
  //      pinMode(thisPin, OUTPUT);
  //}
  pinMode(pin1, OUTPUT);
  pinMode(pin2, OUTPUT);
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  pinMode(pin5, OUTPUT);
  pinMode(pin6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(13, OUTPUT);
  analogReference(INTERNAL);
}

void loop() {

/*-------------------------------------------------------------------------------------------------------*/
     // calculate mean and standard deviation

     sampleSum = 0;
     for(int a = 0; a < SAMPLES; a++) {
        s_val[a] = analogRead(Analog_In);
        sampleSum += s_val[a];
     }

     meanSample = sampleSum/float(SAMPLES);

     // HOW TO FIND STANDARD DEVIATION
     // STEP 1, FIND THE MEAN. (Just did.)
     // STEP 2, sum the squares of the differences from the mean
```

```
      sqDevSum = 0.0;

    for(int a = 0; a < SAMPLES; a++) {
      // pow(x, 2) is x squared.
      sqDevSum += pow((meanSample - float(s_val[a])), 2);  //Variance
    }

     // STEP 3, FIND THE MEAN OF THE SUM OF SQUARES
     // STEP 4, TAKE THE SQUARE ROOT OF THAT

    stDev = sqrt(sqDevSum/float(SAMPLES));    // TADA, STANDARD DEVIATION.
                                              //This is in units of sensor ticks (0-1023)

//     Serial.print("meanSample: ");
//    Serial.println(meanSample, DEC);
//     Serial.print("stDev: ");
//     Serial.println(stDev, DEC);
//     delay(3000);

/*----------------------------------------------------------------------------------------------------*/

                         // analysis the neural data

 //       Serial.print("meanSample: ");     //Check the value of mean and standard Deviation
//        Serial.println(meanSample, DEC);
//        Serial.print("stDev: ");
//        Serial.println(stDev, DEC);

          threshold = (meanSample+5*stDev+10) ;  // Calculate the threshold

//        Serial.print("threshold: ");                    //show the value of threshold
//        Serial.println(threshold, DEC);
//        delay(3000);

//        sensorValue = analogRead(Analog_In);  //show the input analog value

          Serial.print("sensorValue: ");
          Serial.println(analogRead(Analog_In), DEC);

//        delay(3000);

    for(int i = data_amount; i >0; i--) {
                // Reading all data_amount (100000) points.
                // If there is no enough points being larger than standard deviation,
                // which indicates the quality of neural signals not acceptable,
                // Then, we move the electrode forward by one-step.

       if (analogRead(Analog_In) > threshold) {

         // detetcted a spike.

           flag += 1;                    //The detected spike number increases
           Serial.print("flag: ");
           Serial.println(flag, DEC);
```

```
        // pin 13 is on, which indicates we get one spike
        digitalWrite(13, HIGH);   // set the LED on
        delay(3000);
        digitalWrite(13, LOW);    // set the LED off
        delay(1000);

        // after confirming the spike, start to transfer neural signal

//Assumeing the sampling rate is 10kHz, each spike last 1to 3 ms, then there are
 //approximately 30 samples at most for each spike. Transfer the following 30 signals

         for (int j = 30; j > 0; j--) {

             Serial.println(analogRead(Analog_In), DEC);

         }

      }

      if (i == 1) {

        if (flag > 20) {

          flag = 0;            //reset the counter
          i=data_amount;  //reset the counter

        }

      }

    }

/*---------------------------------------------------------------------------------------------------*/

  // move the microelectrode

  //Select Microelectrode
  digitalWrite(pin5, LOW);
  digitalWrite(pin6, LOW);

  // Forward Movement Waveform
  digitalWrite(pin1, HIGH);
  digitalWrite(pin2, LOW);
  digitalWrite(pin3, HIGH);
  digitalWrite(pin4, HIGH);
  delay(1);
  digitalWrite(pin1, LOW);
  delay(1);
  digitalWrite(pin2, HIGH);
  delay(1);
  digitalWrite(pin4, LOW);
  delay(1);
  digitalWrite(pin1, HIGH);
  delay(1);
  digitalWrite(pin4, HIGH);
```

```
  //delay(3000);

  // LED shows whether the movement signal is sent or not.

  digitalWrite(5, HIGH);   // set the LED on
  delay(2000);
  digitalWrite(5, LOW);    // set the LED off
  delay(1000);
 // }


}
```

APPENDIX C

PROCESSING CODE FOR WIRELESS NEURAL SIGNAL TRANSMISSION WITH MOVABLE MICROELECTRODES CLOSED LOOP CONTROL

```
// Graphing and Storing Sketch
//This program takes ASCII-encoded strings
// from the serial port at 115200 baud and graphs them. It expects values in the
// range 0 to 1023, followed by a newline, or newline and carriage return


import processing.serial.*;

Serial myPort;       // The serial port
int xPos = 1;        // horizontal position of the graph

   PrintWriter output;
   float actual_value;

void setup () {
// set the window size:
size(800, 600);

   output = createWriter( "analog_data.csv" );  //Store the neural data as file
                                                //analog_data.csv.

// List all the available serial ports
println(Serial.list());
// I know that the first port in the serial list on my laptop
// is always my  Arduino, so I open Serial.list()[0].
// Open whatever port is the one you're using.

myPort = new Serial(this, Serial.list()[0], 115200);
//myPort = new Serial(this, "COM4", 115200);

// don't generate a serialEvent() unless you get a newline character:
// myPort.bufferUntil('\n');
// set inital background:
background(0);
}
void draw () {
// everything happens in the serialEvent()
}

void serialEvent (Serial myPort) {
// get the ASCII string:
String inString = myPort.readStringUntil('\n');

if (inString != null) {
// trim off any whitespace:
inString = trim(inString);
// convert to an int and map to the screen height:
float inByte = float(inString);
//inByte = map(inByte, 0, 1023, 0, height);

   //output.println(2*inByte);  //print the line into excel file
   output.println(inByte);  //print the line into excel file

// draw the line:
stroke(127,34,255);
```

```
line(xPos, height, xPos, height - inByte);
// point(xPos, height-inByte);
// at the edge of the screen, go back to the beginning:
if (xPos >= width) {
xPos = 0;
background(0);
}
else {
// increment the horizontal position:
xPos++;
}
}
}
```