Santa Clara University
DEPARTMENT of ELECTRICAL ENGINEERING

Date: June 19, 2006

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Trang Phan, Joshua Yee

ENTITLED

<u>Music is Possible II</u>

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING

____
THESIS ADVISOR
Prof. Shoba Krishnan

____
ELECTRICAL ENGINEERING
DEPARTMENTCHAIR
Prof. Samiha Mourad

MUSIC IS POSSIBLE FOR THE MOTOR IMPAIRED II

by

Trang Phan and Joshua Yee

SENIOR DESIGN PROJECT REPORT

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Electrical Engineering
School of Engineering
Santa Clara University

Santa Clara, California

June 19, 2006

ABSTRACT

Music is Possible II is a continuation and expansion of *Music is Possible*, a representation of assistive technology, a musical instrument device, as a means of self-expression and therapy for those with motor impairments. To demonstrate this intent, our project incorporates an input interface, analog and digital data processing, and audio output. The device is currently comprised of an on/off switch, five different input buttons, a microcontroller for data processing, and a wireless transceiver. The device functions like a BASIC piano/keyboard with five pushbuttons and is capable of playing a possible of 9 notes through the microcontroller's coding. Simple songs are thus readily played with minimal musical experience. The research and design of assistive technology is valuable for improving life for those with motor impairments who may have difficulty doing what are usually considered routine activities. This project offers a new means of freedom and entertainment that may also create many positive secondary effects that can potentially benefit all members of society. By participating in the development of the field of assistive technology, we as engineers can express the values of "conscience, competence, and compassion," that Santa Clara has instilled in us.

Keywords: Assistive Technology, Motor Impairment, Audio Control, Musical Notes, Therapy, Microprocessor, Analog, Digital Processing.

ACKNOWLEDGMENTS

First we would like to thank Joshua and Jeremy F., two students from The Avalon Academy, for their great amount of insight into the lifestyle of those with motor impairment. In addition, we would like to thank the director of the Academy, Kinga Czegeni, as well as the staff, other students and their parents for their willingness and enthusiasm to support us in this project.

In addition we would like to thank the School of Engineering: The Dean's Fund Committee for providing the opportunity by funding the project; Engineering Technician, Yohannes Kahsai, for supplying us materials and instruments; the Electrical Engineer Administrative Assistant, Ben Greer, for introducing us to the MIDI program; Professor Christopher Kitts for his practical suggestions regarding components and testing; Professor Sally Wood for her lecture on testing & verification; and all the other faculty and staff for their support. Last but not least, we would like to thank our advisor, Professor Shoba Krishnan, for her constant reassurance and encouragement in all aspects of the Senior Design process. And for keeping us motivated.

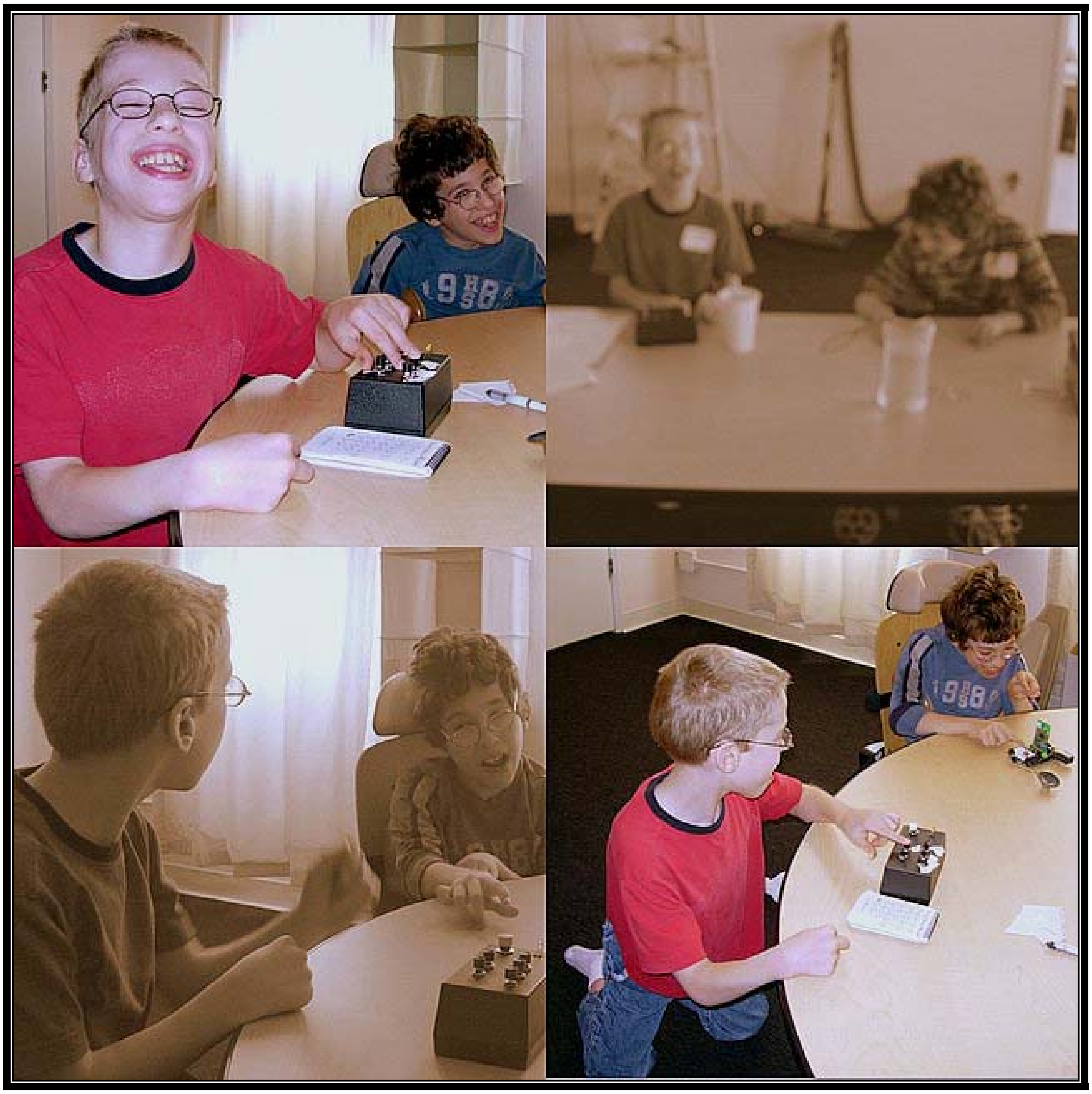
Lastly, we would like to thank our families, peers, and friends: Our families for their support; Kevin Ip and Luis Vicencio for their help in continuing their efforts from their project last year; Phelps Williams for helping us with a lot of coding and troubleshooting microcontroller/wireless issues; Jacob Latone for his suggestions and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

MUSIC IS POSSIBLE II
2005-2006



INTRODUCTION
CHAPTER 1

1.1 Background

Assistive technology has helped the many lives of the disabled, impaired, and the physically challenged. An assistive technology device is "any item, piece of equipment, or product system... that is used to increase, maintain, or improve functional capabilities of individuals with disabilities." These devices can range from pencil grips and splints (low-end technology) to voice synthesizers and Braille machines (advanced technology). This technology gives those with disabilities, opportunities to make tasks easier [Kelker, 1997]. Because there is no cure for some disabilities (e.g. cerebral palsy), it is important to create a device that would result in "the greatest possible independence" [Howard, 2000].

Cerebral palsy is a "term used to describe a group of chronic disorders impairing control of movement." The term cerebral palsy is derived from the word "cerebral" which relates to the brain's hemispheres and "palsy" which refers to impairment in the body's movements. Thus, cerebral palsy is caused by "faulty development" or brain damage in the motor areas [National Institute of Neurological Disorders and Stroke, 2005]. As a result, those with cerebral palsy have difficulties controlling the motor function (e.g., motor control and coordination) [Zisook, 2005]. Because of the motor impairment for those with cerebral palsy, participating in activities, such as playing musical instruments, does not come as easy as it would for the non-impaired. One who expresses an interest to participate in such activities should not be prevented from doing so because of one's disability. Jeremy and Joshua F., who attend The Avalon Academy [**Figure 1.1**], have cerebral palsy and were interested in playing a musical instrument. We believed that they too should be given an opportunity to play, thus we decided to continue with last year's project.

Figure 1.1: Joshua and Jeremy F. from The Avalon Academy

1.2 Motivation

1.2.1 *Overview*

Our competence in our technical scholarship allowed us to express compassion and conscientiousness by creating devices such as this project. By showing our awareness of the situations of our targeted group, we understood their needs better, thereby approaching them in humility and partnership. Though we acknowledged that we cannot completely eradicate all negative medical conditions with one single design project, this opportunity initiated our professional careers with a mindset compassion, humanness, focus, and most importantly, hope.

1.2.2. *Personal Interests*

On a less technical level, the intention of this project was also to address our own interest in music and psychology/neurology. Discussed below are our own individual concerns and pursuits that we jotted down back in October 2005:

> "One of the most universal forms of entertainment is music. A child's introduction to music and performance has been shown to improve their academic and social lives as well. Moreover, an outlet of creative expression is constantly encouraged over sometimes less productive forms of entertainments (TV, video games, etc.). I myself regret having not spent at least a minimal amount of time playing the violin or piano as a child. In high school, I taught myself to play guitar and took a few piano lessons, but my childhood years prior to that now indeed seem "wasted" because of the perspective I now have regarding music. My personal goal for this project is to further allow musical access to the children who were first involved last year at the Avalon Academy, Joshua and Jeremy and for more children in general. I believe this device could also encourage instrument practice for musicians in general if it is developed accurately enough. My personal experience with guitar may also allow for a design with more capability than the average toy electronic guitar, through current limitations in the respect are unknown at this time." –*Joshua Yee* (interest—music)

> "Pursuing a career in the field of Neuropsychology, I find myself 'squandering' time browsing through materials in the field of neurology and psychology. When I received more information about this project, I took this project as an opportunity to put those squandered time into use. Not only would I be applying my technical skill that I have learned, but also I can get a hands-on experience in my vocational areas along with reading materials on psychology/neurology. My personal objective for this project is to offer whatever we have, whether it is our technical skills or simply our presence, to Joshua and Jeremy F. Not only do I

hope our project provides entertainment for them, but also as psychotherapy, music therapy, and/or vibroacoustic therapy. The National Institute of Health (NIH) has shown that music can have a positive effect for those with brain injuries. Studies have even shown that music therapy for those with cerebral palsy can help stimulate the brain thus increasing neuron connections. In addition, I have found that vibroacoustic therapy (i.e. patients interact with sounds and vibration) can alleviate the symptoms of those with cerebral palsy. And in case we happen to fail in these areas, my main hope is that we would touch the lives of Joshua and Jeremy in some ways or forms. After all, in order to have a 'strong positive impact' on a community, one has to start out small." *--Trang Phan* (interest—neurology and psychology)

1.2.3. *Project Goals and Motivation*

Everybody, including the motor impaired, should have the same opportunity to access the technology that makes life more enjoyable. One way to make life more enjoyable for the motor impaired is to have a device that would encourage independence and to provide opportunity for entertainment. Assistive technology makes this possible.

We wanted to work in this area because we believed in this philosophy and wanted to improve the lives of those who are less fortunate than ourselves. We want to continue this project because the children, specifically Joshua and Jeremy F. whom Kevin Ip and Luis Vicencio worked with last year [**Figure 1.1**], are excited about this device.

Not only will our project serve as entertainment for the children but also as a form of music and therapy. The National Institute of Health (NIH) has shown that music can have a positive effect for those with brain injuries [Howard, 2000]. Studies have shown that music therapy can help stimulate the brain for those with cerebral palsy, which increases the neuron connections [Krakouer, 2001]. In addition, vibroacoustic therapy—where patients are stimulated with sounds and vibration—alleviates the symptoms of those with cerebral palsy [Howard, 2000].

Creating a device that provides some form of therapy is important, but we also needed to create a device that the children with cerebral palsy would enjoy using. A combination of entertainment and therapy would be useless if clients did not want to use it. Therefore we sought to design a device that has a fun aspect and encourages greater independence by addressing the targeted user's concerns that were posed by the glove interface design. Shown in **Figure 1.2.3** is a glove interface designed by Ip and Vicencio using five flex sensors.

Figure 1.2.3: Music is Possible I Device
(Flex Sensors Shown disconnected)

1.3 Literature Reviews

Note: The following literature reviews were directed towards a device more akin to the *Music Is Possible I* device than our final device. The review made is still important and pertinent because it helped us further understand what the weaknesses were in the previous device, as well as where we could still apply fundamental engineering concepts.

1.3.1 ***Force Sensing Data Glove***:

Data gloves have been used in many different applications such as biomechanics, robotics, virtual reality, telecheric application, and so forth. Tarchanidis and Lygouras's designed a data glove that uses force sensors in addition to tracking finger position [Tarchanidis, 2003]. The force sensors are attached to the glove and as the fingers are bent, the resistance is changed, much like the design used for *Music Is Possible*. The force sensor is made of a steel blade with strain gauges on both sides. They use the PIC 16C74A microcontroller as an analog-to-digital converter (ADC). With the current output, a digital-to-analog converter (DAC) produced. Kevin Ip and Luis Vicencio's work was based on this concept and included an array of flex sensors attached to a glove which would be used as the interface for our musical device. Tarchanidis and Lygouras's work gave us an idea of what type of transducer and/or microcontroller we can possibly use and where we thought whether or not it was effective in a more general application than ours [Tarchanidis, 2003].

1.3.2 *Gesture Recognition for Virtual Reality:*

Virtual reality, another application using data gloves, was proposed by Weissmann and Salomon [Weissmann, 1999]. This data glove is used as an input device that measures 18 angles for the joints: "two for each finger, one each for the angles between neighbouring fingers, as well as one each for thumb rotation, palm arch, wrist pitch, and wrist yaw." Using this, the data glove could detect several common hand gestures, which suggested the possibility for a sign language translator. Using Weissmann and Salomon's idea of measuring 18 angles for the joints gave us insight into a device with more options than just playing different notes. For example, the angles of the wrists could be used to determine the octave of the notes to be played. Palm arching was another possibility for controlling different musical instruments to be played [Weissmann, 1999].

1.3.3 *Parkinson's Disease:*

For the device proposed by Su [Su, 2003], three-dimensional electromagnetic sensors were used to record the motion of the hand and fingers. This device uses 11 sensors and is targeted to those with Parkinson's disease. The glove could measure the tremor, rigidity of the wrist, dexterity of finger pinching, and hand gripping of both those with Parkinson's symptoms and those without the symptoms through the use of generator and sensor coils [**Figure 1.3.3**]. The device can offer more accurate diagnoses due to its recording capabilities that captured hand movements. The device could also "[record] progression of disease" and "[monitor] the effects of therapeutic interventions." In other words, it could watch the patient for declining health or motor function. Like this device, our project was also targeted towards a particular group that is motor-impaired. We had hoped to use the device in a similar fashion for those with cerebral palsy. For example, we had initially intended to design and produce capable of recording and capturing the motion of a user much like a video to be played modified at a later time. We still believe this could be applied towards future work on this or a similar project [Su, 2003].

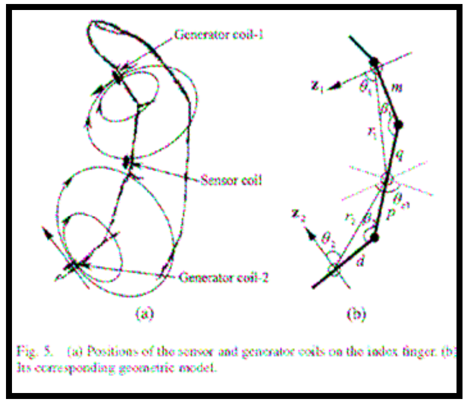


Figure 1.3.3: Parkinson's disease data-glove [Su, 2003]

1.3.4 ***Music Controller:***

This design is roughly the same as our project. Their scope is not targeted as ours is for the motor impaired, however the features and usability are more complex both physically and electronically. In addition to the glove-type interface, flex sensors, and microcontroller that we had initially planned to implement, their project also features a footswitch (a level of control we had not considered) and a computer/MIDI program connection—there is no stand-alone module with self-sufficient power supply, wireless/cordless, etc.—but these aspects allow them to produce a greater range of instruments, essentially limited only by the current MIDI software. This was the most pertinent aspect of their design for our project. Eventually we had intended to produce a MIDI-capable device that could communicate with a computer for further audio signal processing. It also appears they do not address ergonomics with regards to the flex sensors, an issue that directly defined our entire approach to the design process. They merely contented themselves with presenting the feasibility of the device [Chong, 2002].

1.3.5 Data Glove with Reducing Sensors:

This was a very interesting and novel design: A magnetic field generator is used to produce fields and responds to changes in those fields due to deflection of sensor coils attached to fingers [Fahn, 2005]. The physical proof of bending angles, field deflection, and ideal calibration are very clear and concrete. The experimental prototype accurately displayed the functionality of the data glove sensor system. They did not as clearly propose the use of the data though a possible use includes hand motion mapping, to produce a possible sign language "signal" that could be converted to teletype or develop an "image," much like an electronic painting, more flexible than a simple XY-axis mouse. The complexity of the design is much higher than ours, which prevented us from realistically producing a similar device. The one most important aspect, however, is the unobtrusiveness of the sensors: By locating them between the finger joints, there is a reduction in the stress on the fingers due to the sensors (compared to flex sensors), as well as a greater sensitivity to multiple degrees of freedom of the hand [Fahn, 2005].

1.4 Statement of Project Objective and Contribution

Our objective for this project was to make life more enjoyable for the motor impaired by designing a device that would encourage independence, dexterity, and strength. The secondary benefits are the entertainment factor and exercise. Because the previous year's project work created some expectation for adaptation and expansion, we needed to improve instrument capability, amplification and external output, the input interface, and wires/circuitry.

SYSTEM DEFINITION
CHAPTER 2

2.1 Customer Needs

Since our customers/users are the motor-impaired, Jeremy and Joshua F. and the Avalon Academy had volunteered last year to be testing resources and provide input on the project. We contacted them again this year to further understand what was needed in such a therapeutic device. Joshua and Jeremy gave us suggestions, discussed what a typical child with cerebral palsy or any other motor impairment would need in a musical device, and explained further possibilities [**Figure 2.1**]. They emphasized the necessity of a device that must allow ambidexterity; sensitivity to those with limited motor function, yet insensitivity for spastic motion; and fun and enjoyable to use. As younger people, they were not initially concerned with issues such as power consumption or effective wireless range; however, these issues were generally subsumed by their suggestions and our implementations of them.

Figure 2.1: Jeremy (on the left) and Joshua (on the right) providing feedback

2.2 System Requirements/Specification

Based on the users' concerns of the previous device (i.e. not comfortable, dangling wires, not portable, not easy to use.), we came up with a few requirements for our system. First, the system must promote independence. Independence is important because it would encourage the children to use the device more. In addition, studies have said that independence is one way to alleviate symptoms with motor impairment [Howard, 2006]. Furthermore, parents and children of the Avalon Academy have said that being independent would make the children proud of themselves which they felt is important. Therefore, *The Music is Possible*'s glove interface would not meet this requirement because they would have to depend on somebody to put the glove on them. Thus the children need a system that they can use from start to finish (including setting it up and putting it away). Because the children complained about how the glove gave them "sweaty hands," we need to be sure that the device would be comfortable. We want to be sure that the device is comfortable enough that it does not discourage the children from using a musical/physical therapeutic device.

Secondly, the device needs to be portable so that the children can carry the device around the room because some children with motor impairments cannot sit by a table. On that note, the system has to be battery-powered just in case the outlet is not nearby. And because the device needs to be portable and to promote independence, it has to be handheld or compact so that the children can easily carry it around and even place it on their laps. Also, the device should be wireless, so that there are no concerns such as loose wires.

In addition to the promoting independence and having a portable, wireless compact system, we need to address the audio capabilities. Users should be able to control the volume according to his/her comfort hearing range. Lastly, because the system is created for children with motor impairment, it is important that the system can senses input in a reasonable manner. For instance, children with motor impairments may have a limited range of motion; some may have difficulty providing enough force while others cannot control their spastic motion. Thus, the system must allow only a certain range of sensitivity, yet reject accidental motion.

Table 2.2 gives a summary that list of needs and their respective requirements, as well as the justifications for each.

Need	Requirements	Justifications
Promote Independence	Easy to set-up and put away	Independence helps alleviate symptoms of motor impairments.
Portable	Battery-powered	User may not sit near an outlet plug.
Handheld	Enclosed system with plastic casing	System should be durable, weatherproof, and compact.
Freedom from extraneous wiring	Wireless data processing and transmission	Wiring, from previous design, occasionally loosens or becomes entangled.
Improved Audio Capabilities	Tone/amplification control; improved audio data processing	Increased audibility and realism of musical device.
Senses input "intelligently"	Sensitive enough to accept intended input, yet rejects spastic, accidental motion	Users have limited range of motion (either too much or too little)

Table 2.1: System Needs, Requirements, and Justifications Analysis Table

The remainder of the sections [Section 2.2 System Requirement/Specification] is a brief summary of the system requirements, specifications, and constraints.

2.2.1 *Inputs*:

Finger pressure is a necessary input of the system, though we cannot define a specific region of motion because of user-dependent sensitivity. However, qualitatively, below a certain threshold, the input will be ignored (due to "accidental" motion). Above a certain threshold, the input will be sensed and processed by the device. The second input required is the volume/tone control. The user would define a volume setting according to a comfortable range of hearing and desired tone.

2.2.2 *Outputs*:

The output of this device must be the variable audio (tone and volume).

2.2.3 *Function/Behavior*:

The system must have input, wireless signal, and audio processing capabilities.

2.2.4 *Performance Requirements*:

The wireless system must have a viable range (dependent on the distance the test clients were able to maintain from the testing surface—i.e. receiver placed on table with client x distance away). It needs to account for simple line-of-sight. The battery life/power supply should depend on the time the users spend on the device; It should account for both low-power “standby” and “on” modes. In addition, the system volume must be adjustable.

2.2.5 ***Secondary Priority Requirements:***

Below is a list of ideas of what the system could include but is not necessary:

- Switchable synthesized instruments (guitar/drums/piano, etc.)
- Upgradeable/updateable sound quality/processing
- Rechargeable capacity, either by removable battery or charging dock
- Headphones/speaker defeat feature

2.2.6 Constraints:

- Budget: $620–Our project must not exceed the amount of funding given.
- Time–We had only nine months to design the system.
- Man-power—Given the amount of time we had and with only two of us, there was only so much we can do and a maximum rate at which we could work to accomplish our goals.
- Limitations in experience—We were not familiar with wireless systems or MIDI software.
- Equipment—Because of the budget, we had to use some of the existing equipment that did not work exactly how we expected it to.
- BASIC Stamp II—This microprocessor did not allow interrupts and there is an audible time delay in the data processing.
- The device must comply with FCC and other Federal regulations.

2.3 System Overview

Based on the system requirements mentioned in Section 2.2, we decided to create a device that is a baton/remote control-like shape that function like a keyboard/piano. From the system level requirements, that was obtained from the sytem level [**Figure 2.3**].

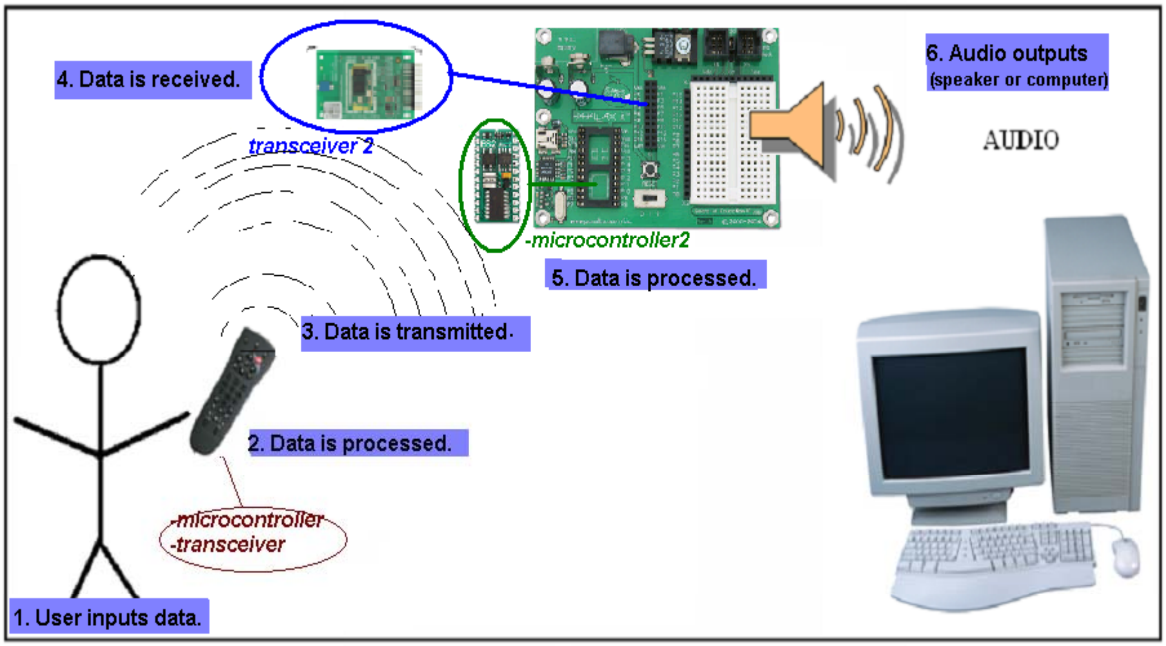


Figure 2.3: System Overview Diagram User Scenario

First, the customer (e.g. child with cerebral palsy) powers on the device to initiate wireless communication. Then, the desired instrument and volume are selected by this remote. The user can then play music with corresponding notes mapped to specific pushbuttons incorporated into this baton-like controller. After the notes have been played, a microprocessor in the "instrument" converts the data to a wireless signal that can be sent by a transceiver to another transceiver. (Since this system uses two transceivers, more features can be added in the future if two-way communication is necessary.) Once the other transceiver has received the data, it is sent into the second microprocessor, which converts the data back to an analog signal. This microprocessor either connects to a speaker or to a computer, which then produces an audio signal.

2.4 Functional Analysis

2.4.1 *Functional Decomposition*:

Using the ideas from the previous year's project group, we extended our system by adding the capacities of both internal and external hardware. **Figure 2.4.1** shows a high-level breakdown of the system functions. The first row of blocks shows the three major components: Sensory data, data processing, and audio processing. The second row shows the previous year's work. The last row shows our possible choices we made.

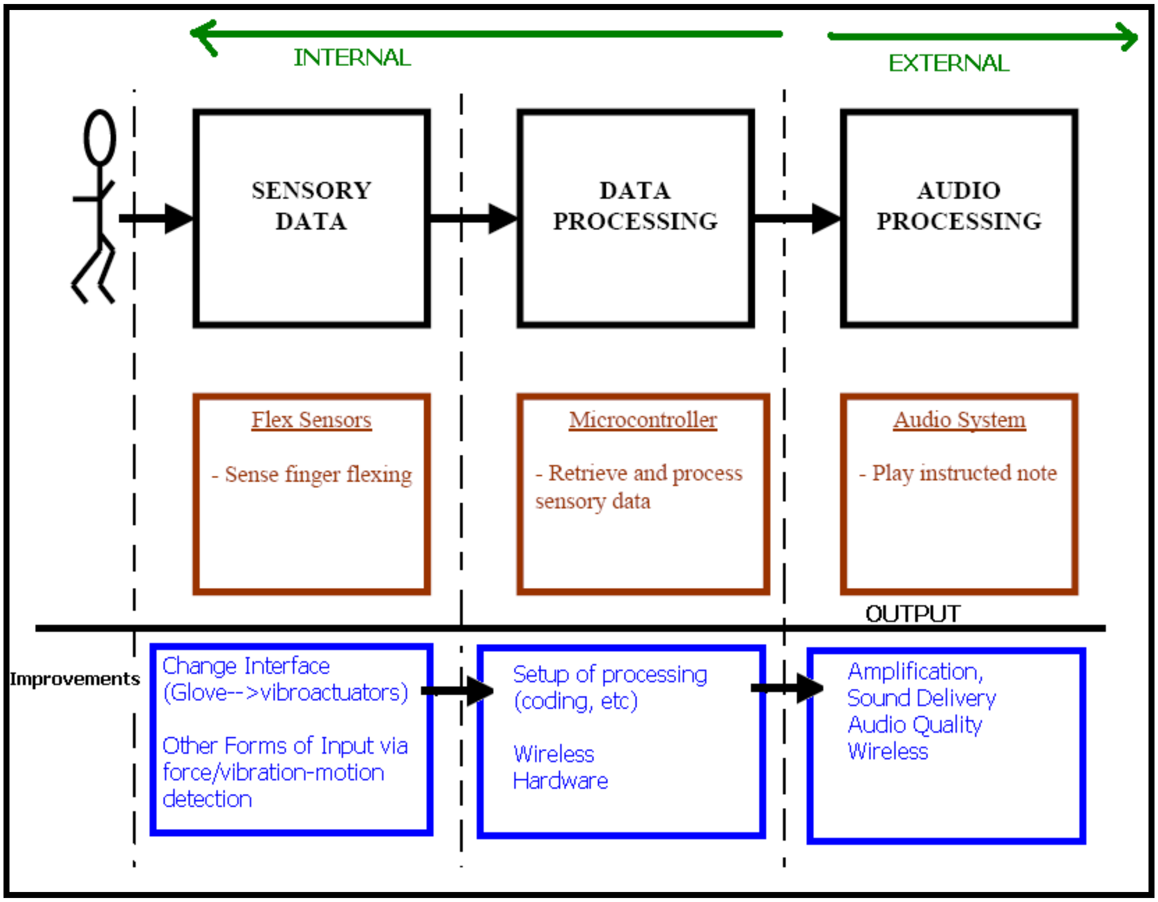


Figure 2.4.1: Functional Decomposition

Figure 2.4.2 shows a similar diagram with the specific components of the subsystems. Details of different components used are also given in the figure. For the interface, there are two major components: the pushbuttons and the microcontroller. We used two types of pushbuttons: SPST Switches for the note buttons and SPDT for the power on/off button. And for the microcontroller we used the BASIC Stamp II OEM Module. This microcontroller transmits data to the second microcontroller via the EmbeddedBlue EB500 Transceiver. This data is then processed using the BASIC Stamp

II Board of Education. There are two ways to connect the data processing with the audio processing. The first one is to directly connect the desktop speaker to the Board of Education. The second is to have a wireless link to the computer via Bluetooth USB Adapter, and MIDI software can be used.

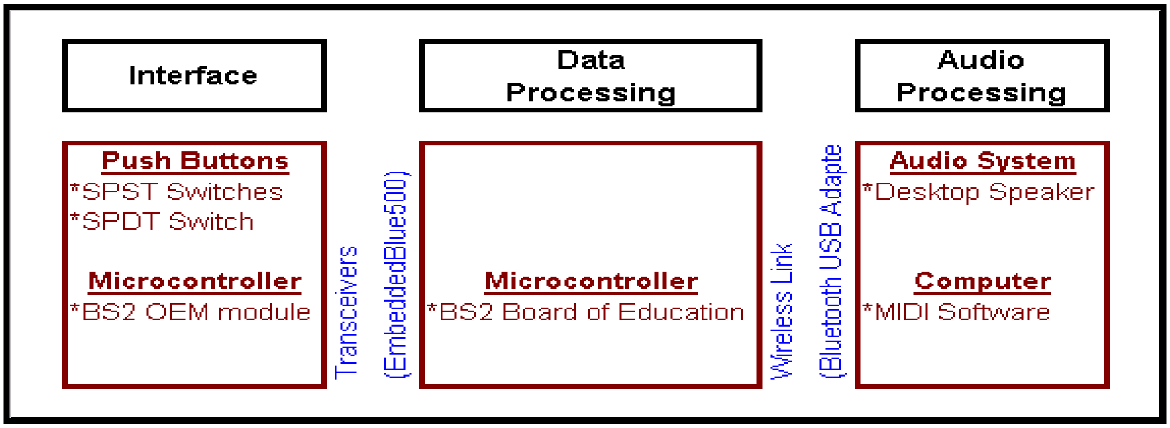


Figure 2.4.2: Subsystems

Figure 2.4.3 gives an image of each of the major components. Circled on the top, left corner are the five SPST switches where the users input musical notes. The user can turn on/off the device by pressing the white SPDT switch. The LED lights up when the OEM and the transceiver inside the device is powered. The Basic Stamp II OEM microcontroller and the Embedded Blue 500 Transceiver inside of the device are shown in the bottom, right corner. This makes up the interface subsystem.

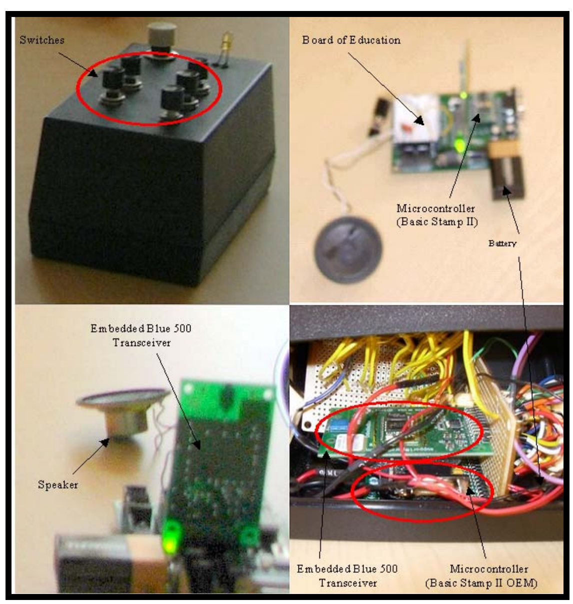


Figure 2.4.3: Component Parts

The wireless transceiver inside of the interface device, transmit inputted data to transceiver on the Board of Education (BOE) as shown on the lower, left quadrant of **Figure 2.4.3**. The data is processed using the Basic Stamp II microcontroller located also on the BOE (shown in the top, right image). This makes up the data processing subsystem. The speaker is directly connected to the BOE for audio processing test verification (bottom, left image).

2.4.2 *Function Summary:*

In summary, the inputs for the system are controlled by manual keying, volume control, tone/instrument control, and computer programming. The output of this system is musical notes. The main constraints are having a system that is compact, battery-operated and AC powered, and FCC et al. regulation compliance.

INPUTS:

1. Manual keying
2. Volume control
3. Tone (instrument) control
4. Computer programming

OUTPUTS:

1. Musical Notes

CONSTRAINTS:

1. Compact
2. Battery-operated and AC powered
3. FCC et al. regulations compliance

2.5 Benchmarking

Knowing and exploring other designs people have published or produce for their projects is important for our own understanding of our project and its potential capabilities. Through benchmarking, we got a much better sense of what the different attributes, capacity, and parameters are that are involved in other, similar projects. We can incorporate some sort of range into our project.

2.5.1 *Flex Sensor Used by the Hypersense Complex*

The Hypersense Complex Project consists of members from the Australian National University pictured in Figure 2.5.

Figure 2.5.1: Hypersense Complex Project

The team has implemented eight bendable flex sensors per person, with four sensors on each hand. Bending these flex sensors allows the team to play musical sounds. These sound data are then loaded into the computer. The sensors are programmed according to the type of music they would like to make. Table 2.2 is a table that lists the attributes and parameters of the flex sensor that this group used.

Attribute/Capability	Parameters/Description
Nominal resistance (at 0 degrees)	10, 000 ohms (10K)
Maximum Resistance	30-40 K ohms
Maximum Force Requirement	5 gram
Maximum Deflection Registered	90 degrees
Sensor Measurement width	¼ inch
Sensor Measurement length	4 ½ inch
Sensor Measurement thickness	0.020 inch
MSRP (Flexible Bend Sensor)	$10

Table 2.2: Flex Sensor Specifications

2.5.2 *ATMEL—Smart RF Wireless Data Microtransmitter*

Atmel has a variety of RF application that can be used in integrated circuit. The product includes a receiver, transmitters, and transceivers that can be integrated with the MARC4 4-bit or AVR microcontroller. Table 2.3 is a table of the attributes for the microtransmitter.

Attribute/Capability	Parameters/Description
RF Frequency Range	264-456 MHz
Power	2-volt operation
RF Output Power Adjustable	Over 36dB w/ 1db Resolution
Modulation	OOK
Data Bandwidth	10 Kbps Manchester
Flash Programmable Memory	2KB
Memory	EEPROM, SRAM

Table 2.3: Atmel Wireless Microtransmitter

2.5.3 *Essential Reality P5 Glove*

The reality P5 glove is one of the first to be used as a virtual three-dimensional controller, shown in Figure 2.7. Like the Hypersense Complex Project, this device also uses some form of bending sensor. In additional to the bend sensors, the P5 is also based on the technologies of optical tracking for roll, pitch, and yaw measurement. This glove is also portable and can be used anywhere. Table 2.5.3 lists other capabilities and descriptions of this device.

Figure 2.5.3: Virtual Reality P5 Glove

Attribute/Capability	Parameters/Description
Weight	4.5 ounces
Degrees of Tracking	6 (X, Y, Z, Yaw, Pitch, and Roll)
PC Interface	Plug-and-Play USB
Wired	Yes
Receiver	Yes (Infrared)

Table 2.4: P5 Virtual Reality Glove Benchmarking

2.6 System Options/Tradeoffs

2.6.1 Trade-Off Analysis I:

One of the most important decisions in our project was to choose which microcontroller to use. There are three different possibilities we can use as our microcontroller as shown in Figure 2.6.1.

Microcontroller Options

1) BASIC Stamp 2 series
2) AVRmini
3) PC

Table 2.5 shows a summary of our tradeoff analysis for the microcontroller.

Microcontroller	Cost	Complexity	Memory	Size	Availability
BASIC Stamp 2	Low (+)	Low (+)	Low (-)	Low (+)	Yes (+)
AVRmini	Medium (~)	High (--)	Medium (+)	Low (+)	No (-)
PC	High (--)	Low (+)	High (++)	High (--)	Yes (+)

Table 2.5: Microcontroller Trade-off

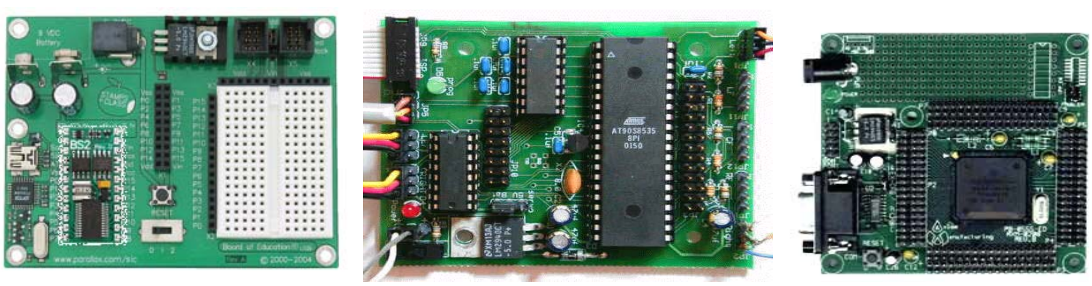


Figure 2.6.1: Microcontrollers--BASIC Stamp, AVRMini, PC microcontroller (L to R)

We chose the BASIC Stamp-type microcontroller because it is portable, durable, and relatively user-friendly processing device. One of the main reasons we chose to use BASIC Stamp was because it was widely available: It was used in the previous project, and there are a few classes in the School of Engineering that are currently using the BASIC Stamp board. Thus it was easier to ask for help. Although the AVRmini microcontroller provides more features than the BASIC Stamp microcontroller such as interrupts, speed, and more memory, it has a steeper learning curve. Because of man-power and time constraints, we would not have enough time to learn the fullest extent of AVRmini's features. In addition, the extra features that the AVRmini provides were not crucial for this particular project; thus we decided to continue using the BASIC Stamp board. Continuing to use the BASIC Stamp board saved us money because we already had at least one available to us. This microcontroller choice can then be supplemented by a computer platform such as a PC in order to update programming, debug, or monitor occasionally. Although the PC is the most powerful of all three options, it is not as portable as the other microcontrollers, nor should it be technically compared in the category of microcontroller. It is unreasonable to expect the children to carry the computer around as an instrument, or even just part of one. For our final combination we decided had the most optimal performance by providing portability while maintaining computability.

2.6.2 Trade-Off Analysis 2:

Another important analysis is what type of wireless system we should use.

Wireless System

1) Transceiver-Transceiver
2) Receiver-Transceiver
3) Transmitter-Receiver
4) Transmitter-Transceiver

Table 2.6 gives a summary of the tradeoff of the four wireless pairs.

Wireless Pair	Cost	Flexibility	Directionality	Power Consumption
Transceiver-Transceiver	High (~)	High (+)	Bi (++)	High (--)
Receiver-Transceiver	Medium (~)	Medium (-)	Uni (-)	Medium (~)
Transmitter-Receiver	Low (+)	Low (--)	Uni (-)	Low (-)
Transmitter-Transceiver	Medium (~)	Medium (-)	Uni (-)	Medium (~)

Table 2.6: Wireless Pairing Trade-off

For the wireless system we chose a transceiver-transceiver, more specifically the Embedded Blue 500 Transceivers as shown in **Figure 2.6.2**, pair to allow the greatest flexibility through bidirectional wireless communication. The cost-benefit analysis led us to conclude that the minimal price difference is outweighed by the usefulness as seen in **Table 2.6**. Our emphasis on bi-directionality is due to our conclusion that, in the future, it might be necessary for the computer and microprocessor to communicate with each other, rather than only from the microprocessor to the computer.

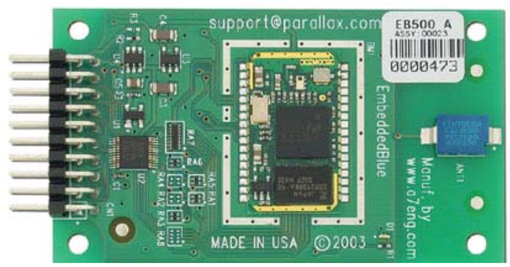
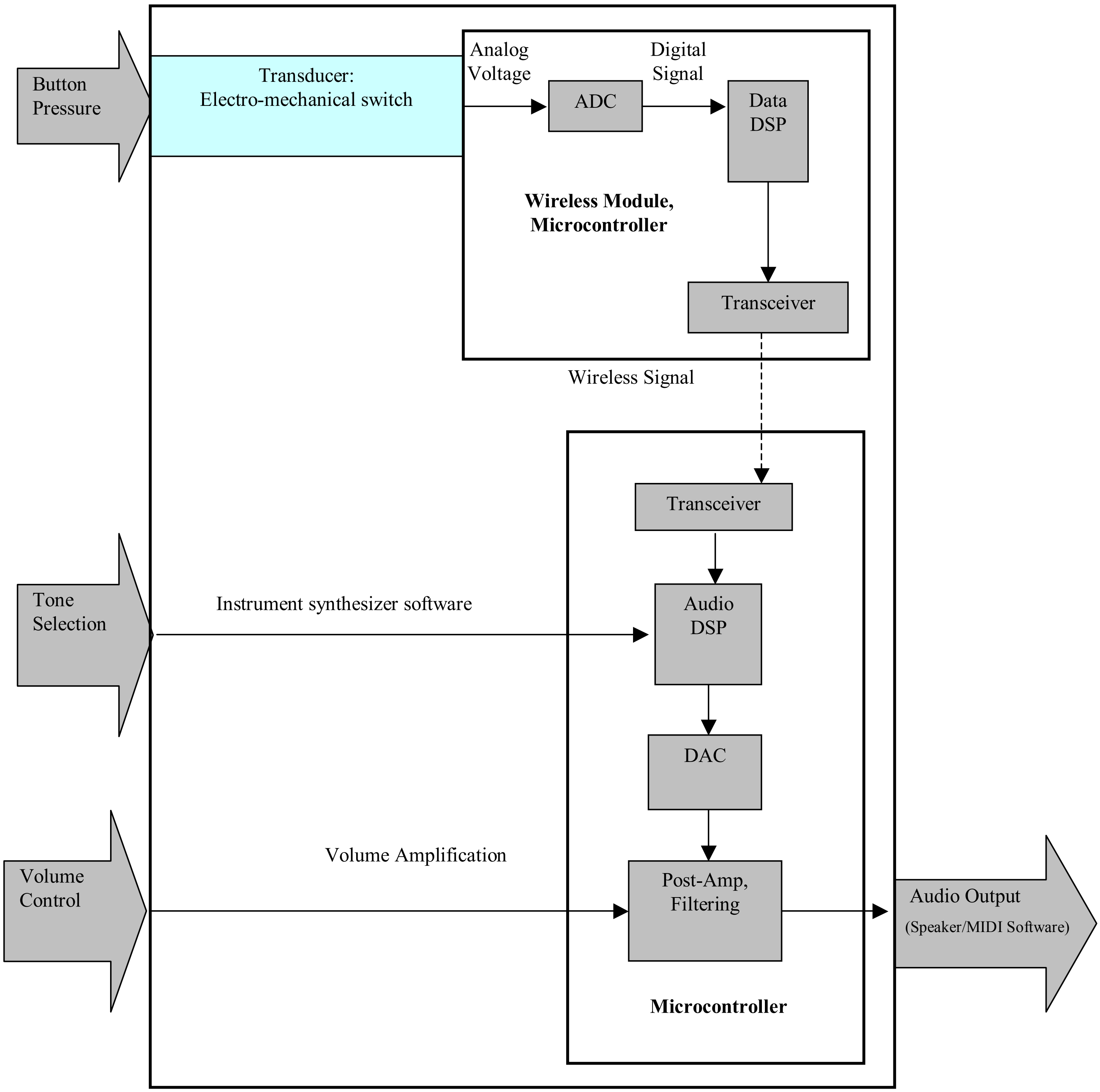


Figure 2.6.2: Embedded Blue 500 Transceiver

2.7 System Layout

Figure 2.7 is a system block diagram that shows how all the subsystems are connected to each other. There are three main inputs: Button pressure, tone selection, and volume control. The button pressure creates an electro-mechanical connection that leads to the microcontroller and wireless module. In this unit, there is the ADC, Data DSP, and the transceiver. This transceiver sends a wireless signal to the second microcontroller unit. This unit is connected to the transceiver to the audio DSP, then to the DAC, and post-amp filtering. Another input to the Audio DSP is the tone selection where the software synthesizes the instrumental device. Another input for the post-amp filtering is the volume control where one can amplify the volume according to ones need. The entire system outputs an analog sound to either a speaker or the computer using MIDI software.

Figure 2.7: Component Block Diagram



2.8 Team Management

2.8.1 Project Challenges/Team Management:

Our team consisted of two members, both in the electrical engineering department. We divided up the minor tasks such as documentation, research, circuitry, etc., and when we needed help we would let the other know what the other person could do. For the major components of the project, we worked together since our group was small enough. Because there were only two of us, we communicated through e-mails and phones for minor issues and met when we were actually doing the project.

We met the actual customers/users (e.g. Joshua and Jeremy F.) to understand their needs and desires. We received positive reactions from them when we presented them with both our prototype and the final implementation. The difficulty with that process was being able to actually visit The Avalon Academy and the boys regularly enough to make progress with the design.

2.8.2 Budget:

The Dean's Fund from the School of Engineering gave us a budget of $620. The major components that we purchased were the BASIC Stamp II OEM module, which cost $59 each. The other components were the wireless components, the Embedded Blue 500, which cost $99 and the USB D-Link Bluetooth Wireless Transceiver which cost $19. For more details on the budget, refer to **Appendix M**.

2.8.3 Timeline:

Due to the set amount of time we could work on this project (9 months) and because we have a small group, we had to plan out our time wisely. We projected our progress with a Gantt chart which can be seen in **Appendix L**.

2.8.4 Design Process:

Testing and measurement was done by both of us on the project team: We checked tolerances of components (microcontroller, circuitry, etc.); wireless system (max radius, power consumption, etc.); and the power system (reliability, power consumption, etc.). The second phase was a test run involving the clients of The Avalon Academy to verify their ranges of motion, sensitivity, preferred distance, etc. Inspection/analysis of circuitry was done before the visits, through real-time, program-based adjustment (note duration and frequency). Comparison of the majority of system requirements was qualitative, as mentioned above, depending upon the personal mobility of the individual clients. From our collected responses from the clients, we went on to make the necessary adjustments in circuitry, coding, or aesthetic appearance.

2.8.5 Risks and Mitigations:

We could not perceive any serious risks in the design of our project. The most dangerous aspect of our device was a potential choking hazard if the button caps were removed. Because the main input is enclosed in a project enclosure box made of ABS plastic, there were little external electrical components exposed.

INPUT INTERFACE
CHAPTER 3

3.1 Introduction

The two major components of the input interface we used were the pushbuttons [**Figure 3.1**] and the microcontroller [**Figure 2.4**]. We used an array of SPST pushbuttons as our musical input. The depression of the pushbuttons was light enough to accommodate the targeted users with limited motor capability, while simultaneously providing a tactile resistance for rejecting spastic or accidental motion. For the interface input, we used the BASIC Stamp II OEM module. The volume can be controlled by turning the dial of the speaker.

Figure 3.1: Color-coded SPST Pushbuttons

3.2 Design Choice

Following our visit with Jeremy & Joshua at the Avalon Academy, we concluded that it was necessary to consider finger deflection as *inadequate* means of input sensing or measurement. Possible options, as mentioned, included the pushbutton switches, a slider-bar resistive switch, or a keyboard or keypad-type array with force-sensing dynamics (volume of output in correlation to intensity of depression). The order of complexity increased with each step taken towards a simulation of a keyboard, while the ambiguity of the instrument decreased. Our intention was not to make a keyboard/synthesizer, but rather to provide a device that simulates a multitude of instruments while not resembling any one in particular.

3.3 Design Description

Flex sensors introduced a level of difficulty in motion higher than that of a simpler array of pushbuttons, therefore pushbuttons were chosen. We chose momentary-on SPST buttons for the musical switches and a push-push SPDT button for the power on/off switch. The explanation can be seen in **Table 3.1**. However, once that decision was reached, we were introduced to a new problem of producing an analog signal through an inherently digital input device. Another issue on the structural level that we faced was the type of enclosure and circuit mounting that would be most reasonably sized and comfortable for our users.

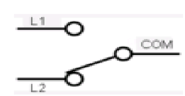
Switches	Descriptions	Symbols
SPST single pole single throw	A simple on-off switch: The two terminals are either connected together or not connected to anything. An example is a light switch.	
SPDT single pole double throw	A simple changeover switch: C (Common) is connected to L1 and L2.	

Table 3.1: Switch type comparison

3.4 Design Solution

Figure 3.4.1 shows a schematic of how we connected our SPST switches. This switch is connected so that it is active-high: when the button is depressed it is "on". For more explanation of how the circuitry works, go to **Appendix K**.

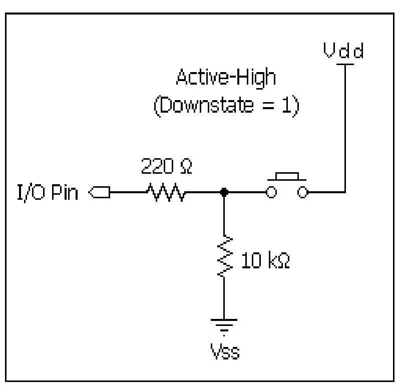


Figure 3.4.1: Schematic of Active-High Switch

We used a project enclosure that was drilled through for switch mounting, while much of the circuitry is point-to-point soldered, although the BASIC module and the transceiver are also hard-mounted inside the case [**Figure 3.4.2**].

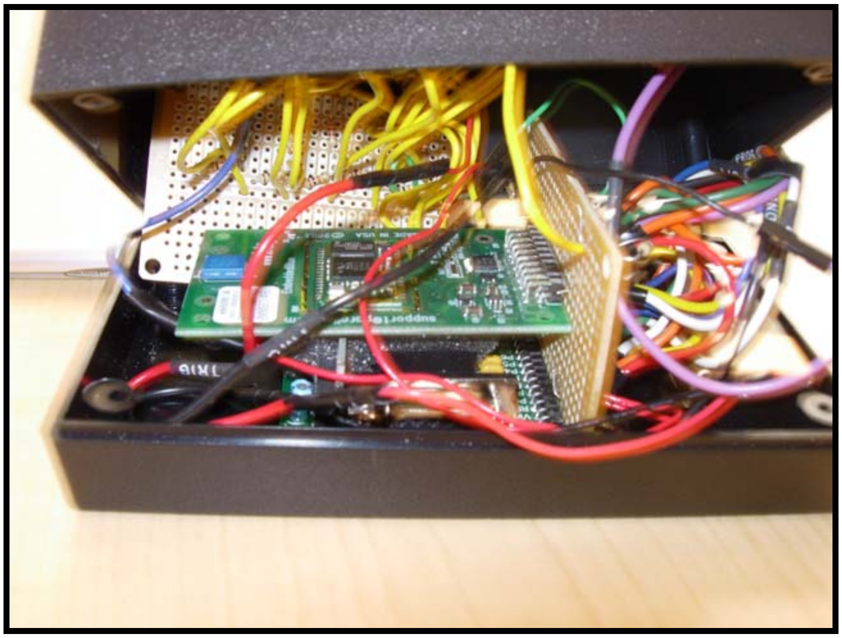


Figure 3.4.2: Project Enclosure with transceiver and microprocessor

3.5 Testing and Verification

We decided that the mechanical aspect of the switches was the most important to be tested. Much later after the input subsystem had been constructed, we took the entire device to The Avalon Academy to test qualitatively whether Joshua and Jeremy were comfortable with the mechanical resistance and rigidity provided by the pushbuttons we had selected. The answer was a solid yes; if the switches had been any softer, they would not be sure they had physical applied enough pressure to register an input, while a stiffer switch would have been too difficult to press at all.

3.6 Conclusion

We concluded from the qualitative test that the mechanical aspect of the switch was sufficient for our purposes. In the future, were an actual consumer product is constructed, proprietary pushbutton switches might be used for the input device.

DATA PROCESSING
CHAPTER 4

4.1 Introduction

The data processing system consists of two major components: the receiving transceiver and the microcontroller. The two transceivers were used to connect the input interface to the data processing subsystem. The receiving transceiver was connected to the BASIC Stamp II Board of Education as depicted in **Figure 4.1**. The microcontroller worked in conjunction with a computer to process the mixed-signal nature of the inputs. Once initially programmed, the device was able to function independently of a computer, acting as its own unit.

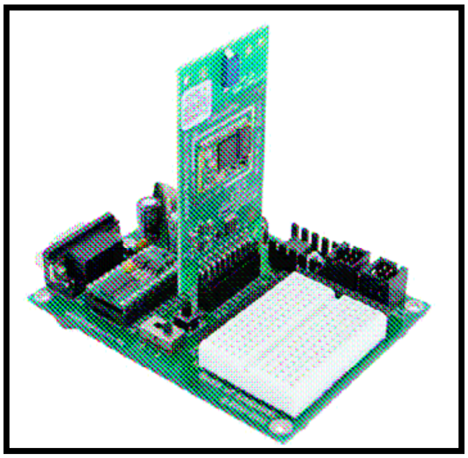


Figure 4.1: Board of Education with EB500

4.2 Design Choice

We chose the BASIC Stamp II Board of Education as our microprocessor in the data processing subsystem. Refer to section 2.6.1 for the tradeoff analysis of the microprocessor. Using a battery-powered unit increased the devices independence and portability rating. It was also necessary to do so due to the nature of the transceivers and a particular aspect of their power handling. While the unit transmitted the musical signal wirelessly to a receiving BASIC Stamp BOE board, it will be possible in the future to include headphone capability or and RCA stereo jack to connect to an external stereo speaker system.

4.3 Design Description

We are currently using the BASIC Stamp II Board of Education (BOE) board for data processing, complete with on-board serial computer interface and a modified 20-pin layout for I/O interface. This module is in turn connected to both the switch array and the wireless transceiver, an EmbeddedBlue 500 Transceiver. Problems facing us included a properly regulated power supply input for the Stamp module and the transceiver for the enclosed portion of the device, similar to the BOE.

4.4 Design Solutions

When enclosing the OEM Stamp module and the transceiver, we needed to provide a structure for support as well as wiring capability. The natural choice for that structure was a PC board, arranged and soldered in the proper fashion to connect the leads from both the OEM module and the transceiver to their appropriate pins.

4.5 Test/Verification Data

The testing of the device initially began with a Wheat Thins box for the sake of general structure. This was used in the second visit to The Avalon Academy to get a better idea of the general shape and size of the final device. Once measurements and suggestions were made, the final enclosure was decided upon to hold the data processing portion. This physical structure defined the coding that controlled the duration and frequency of the notes played because of the physical rigidity. The testing of the data processing hardware in the context of a project enclosure yielded positive results and indicated a functional device addressing our objectives.

Figure 4.5 is a chart of the final duration measured by an oscilloscope. The comparison was made from the coded frequency vs. the actual frequency output. After analyzing several output durations, we settled on the final duration, which we then coded into our receiving microcontroller to play the note (FREQUOUT function in BASIC). For further data, see **Appendix R**.

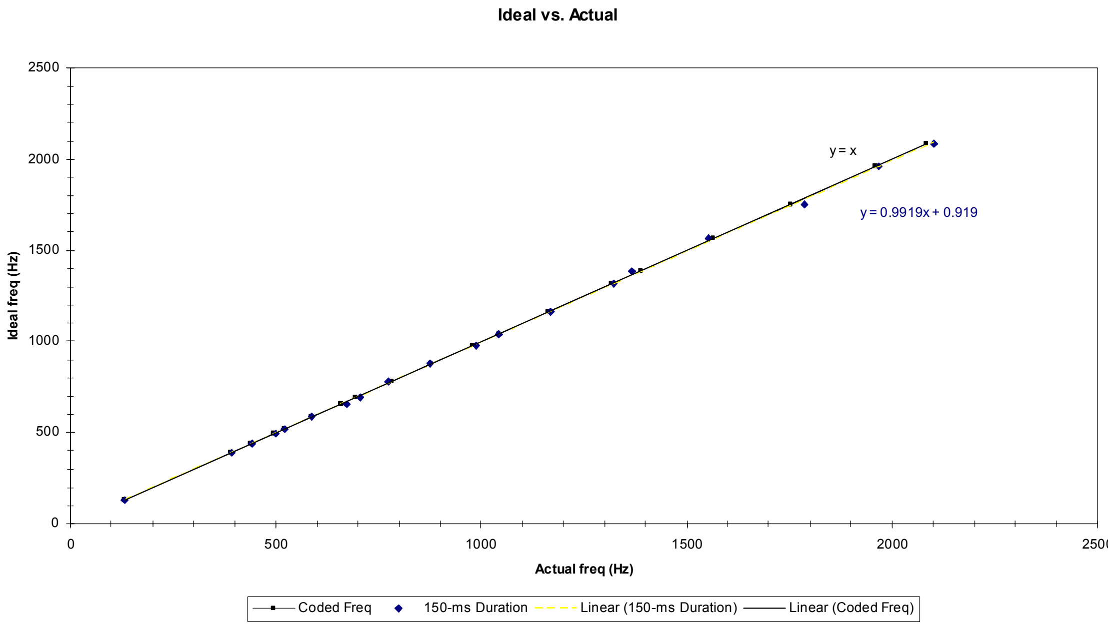


Figure 4.5: Note duration data measured via oscilloscope

4.6 Conclusions

We had a functional device following the initial Wheat Thins-box prototype. The data processing was difficult initially because of a lack of input capabilities (i.e. an insufficient number of pushbutton switches and structure to support them).

AUDIO PROCESSING

CHAPTER 5

5.1 Introduction

The audio portion of the device produced clear sound and has adjustable volume. An audio amplifier and reasonably-sized speaker are necessities to address these requirements. We have thus far completed the device to the degree that we can physical demonstrate its capabilities to address our project objectives. Future work can be done to show that this device will produce a signal for MIDI audio processing raising the level of sonic capabilities.

5.2 Design Choice

We chose to have two different options for audio processing. One is to use a desktop speaker. The speaker would simply connect to the Board of Education using the circuitry shown in **Figure 5.2.1**. This option allows the user to input his/her own volume preference.

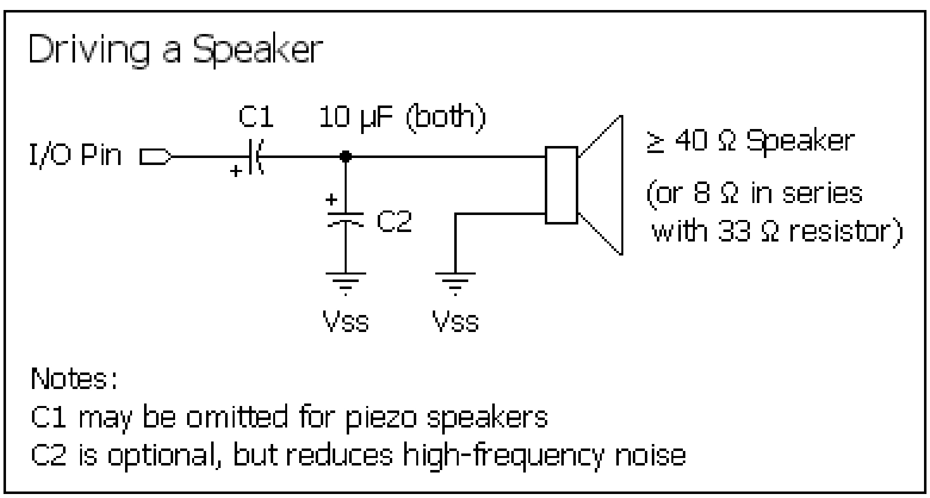


Figure 5.5.1: Driving Speaker Schematic

In addition to the desktop speaker, another option we considered was MIDI processing. MIDI is short for Musical Instrument Digital Interface. It is a way to electronically synthesize each musical note. Some of the reasons why we felt that MIDI processing would be good to use is because it is a standardized digital music that is used in many different application. It also takes up less size. For instance, much digital music would require 10 megabyte per minute of song; MIDI, on the other hand, requires only 10 kilobyte. Also unlike much other digital music, MIDI can give individual instructions for individual notes of each individual instrument. So, potentially, a user can change one note of the entire song with a different instrument. **Figure 5.5.2** is a snapshot of the MIDI software we used called Live Lite.

It shows that there are many instrument options. We can preset a key to play a certain sound. In this case, letters, a to f, were preset to play their respective rows. Letters, g to k, were preset to play their respective columns. For instance, pressing c on a keyboard would play the chorus located on the third row (bass, guitar chord, rhodes2, shaker, drum B). The user can also preset which instrument he/she likes to hear when a certain key was pressed.

Figure 5.5.2: Screenshot of Ableton's Live Lite 4

5.3 Design Description

Currently, the audio levels are sufficient for simple music. The device does what it was initially meant to do, which was playing music. Unfortunately, the nature of the data processing actually restricts the natural quality or fluidity of the music produced. We did not manage to produce a device capable of switchable tones (guitar/piano, etc.), but that would be encompassed in a future solution through MIDI processing.

5.4 Design Solution

To improve the sound projection from last year, we decided to simply connect the circuitry to the desktop speaker. To provide multiple instrument tones, we had planned to use a MIDI program.

5.5 Test/Verification Data

Figure 5.5.1 displays the circuitry for our speaker. We connected one end of the speaker to the ground, and the other end is connected to the I/O pin. The capacitor, C1, was used when we used the desktop speaker because the piezo speaker was not sufficient for our project. The capacitor, C2, was used to reduce the high-frequency noise. We found that 10 microfarad works well for our project according to the suggestions given in Parallax schematics.

5.6 Conclusion

For users who are not comfortable with a computer or do not have access to a computer, they can use the desktop speaker option. And those who want to play more sophisticated sounds can use MIDI processing. This addresses the flexibility issues originally encountered by the small piezoelectric speaker used by the previous year’s design for *Music is Possible I*.

PROFESSIONAL ISSUES
CHAPTER 6

6.1 Compassion

One issue that is relevant to our project is compassion because in a sense, it shows our solidarity with those who are suffering. Being aware of those who are less fortunate than ourselves gives us a stronger motivation to do this project. We did not only want to do this project for the sake of getting our engineering degree but also to do something that is more meaningful for developing our characters and improving the world. One way to do that is to learn how to be compassionate by associating with those who are suffering. In our case, we decided to unite with children with cerebral palsy for cause they were enthusiastic about. We made a product that would provide musical entertainment as well as therapy for these children. By getting to know their needs and desires, we were able to build a product that can possibly make their lives better thus alleviating any suffering. Such healings can strengthen our community and society.

6.2 Ethical

This project addresses ethical issues because there is a concern about the disabled. Everybody deserves the same rights and opportunities as the others. By looking at the justice or fairness approach, we can decide what we can do to so that everybody is treated equally. By creating some product for the disabled, they would get a similar opportunity to enjoy the technology as others, thus making the inequality gap smaller.

6.3 Usability

The usability of this device directly influences its design. Our clients' limited ranges of movement change the common conception of the "average user." Because of this, we must actively design the input system to be as accessible as possible—i.e. maintaining a balance between simplicity and functionality.

6.4 Manufacturability

Due to the continuing nature of our project, our intended areas of improvement will now have an impact on the manufacturability of future iterations of this device. Our efforts are strongly dictated by the usability of the device, as well, and thus govern the manufacturability. We hope that we laid a foundation that would actually get these types of device manufactures (particular for the Avalon Academy)

6.5 Lifelong learning

This project proves to be a vehicle of lifelong learning for both those who are designing the device and those who are intended users. Obviously the Senior Design Project is intended to sum up the corps of engineering knowledge we currently possess, but the opportunity to additionally provide a similar experience for the clients through the freedom the device imparts is the strongest and best motivation for those who are designing it.

6.6 Sustainability

> "Our prototype does address a few sustainability issues, at least in the narrow sense. The sustainability issue our prototype address is whether or not it can be easily adjusted to accommodate different users. The PBASIC source code that we used can be adapted to accommodate different users' limitations, which is done by adjusting the detection range values in the PBASIC code."
>
> --Kevin Ip and Luis Vicencio

This statement made by the previous year's project team is still accurate for our new circumstances. We have had success in our project due to the true sustainability of the code. Adaptations were indeed made in the code through concurrent design and this held true when we attempted to test non-impaired users.

6.7 Environmental Impact

There were no significant environmental issues that came up in our project.

6.8 Social

"This project impacted society by bringing more awareness of the needs of people with disabilities and what their lives are like. The hope is that in the next few years, more and more engineering seniors will choose projects in the area of assistive technology. A small step in that direction was taken this year, bringing more awareness to the school population of the needs of people with disabilities.
This project also had direct impact on the community outside the school. At The Avalon Academy, we were able to explore with students with motor impairments their teachers how to make this device fit them. We were very fortunate to be able have this connection with the Avalon Academy, and we hope that in future years, more connections like this will be made between engineering teams and the outside world."

--Kevin Ip & Luis Vicencio

Our experience this year was very much the same as the previous year's team.

6.9 Economics

"The primary motivation of our project is to make a product that will be adapted for people with motor impairments to create music. However, making a profit was not a primary motivation. The project depended solely on educational grants for funding and will likely in future be dependent on grant funding as well."

--Kevin Ip & Luis Vicencio

6.10 Political

There is nothing explicitly political in our project that we could discern.

6.11 Health & Safety

The only health and safety issue that arouse in the development of our project was electrical safety. Our prototype had to be plugged into a wall socket, but there was careful supervision when the students were testing out our device. There was no need in the scope of our project to make engineering decisions based on electrical safety.

SUMMARY
CHAPTER 7

We conducted a survey towards the end of our project year through The Avalon Academy which asked the opinions of parents of children there. The results were overall positive. The survey can be found in **Appendix N**, results in **Appendix O**.

The current design doubtless needs improvements in all the areas mentioned: Input, data processing, and output. When we visited with Joshua & Jeremy at the Avalon Academy, Jeremy made many suggestions regarding the interface. His concerns were that a glove apparatus would not be ambidextrous, as comfortable as initially thought, nor truly his desired form of device for self-expression. We discussed with him the importance of a wireless system, complete with a wireless data and audio transmission either to computer or home stereo system. It is certainly possible that a person intending to play this instrument will wish to be wandering around a room, but it seems more likely that they will still be nearby the receiving audio system. Thus portability is not as high a priority as we initially thought. Therefore, if we seek to design a device that is at least unexposed, while containing all the necessary components for proper functioning, has battery-operation capability, and addresses the current limit of audio output, and then we will have solved the majority of the question for this device. The device was first intended to be attached or connected to the hand and thus size was of greater concern, but if this new system is bulky, but still demonstrates integration of all the aspects we have characterized as necessary subsystems, then the entire system will be feasible.

We have met these objectives in our device in the following ways:

- Promotes independence through of ease of use. The device requires minimal set up and maintenance versus the original design (such as caregivers putting on the gloves for patients with cerebral palsy)
- Is comfortable by addressing symptoms such as sweaty hands or dangling wires.
- Is portable and battery-powered which allowing the children to move around the room freely rather than having to sit next to an electrical outlet or be

tethered to a computer cable. Children can sit on the ground or lie down in a fashion befitting their physical condition.

- Is handheld and compact allowing the children to carry the device anywhere they want with very little external help (unlike the piano/keyboard which would be too heavy and very difficult for the kids to move around)
- Is wireless to allow the children to move around freely without worrying about being outside of the range of the processing microcontroller.

Our device is a starting point for a therapeutic, musical device for the children with motor impairments, at least at the scale of an organization like The Avalon Academy. With the interest of Joshua and Jeremy F., parents, and the staff of The Avalon Academy, we feel that we have strongly developed a foundation and supporting ideas for any upcoming groups to work on a product could be used in the future. We would like to maintain this relationship between Santa Clara and The Avalon Academy through this project.

Bibliography

"About the P5." P5 Glove Virtual Control—P5 Virtual Reality Glove. <http://www.alliancedistributors.com/Alliance_Brand/Products.php> Oct. 2005.

"Cerebral Palsy: Hope Through Research." National Institute of Neurological Disorders and Stroke. <http://www.ninds.nih.gov/disorders/cerebral_palsy/detail_cerebral_palsy.htm#12053104> Oct. 2005. 13 Oct. 2005.

"Flexible Bend Sensor." Images SI Inc. <http://www.imagesco.com/catalog/flex/FlexSensors.html>. Oct. 2005.

"Smart RF Wireless Control System Application Note." Atmel. Atmel Corporation 2003.

Chong, Uipil and Changwon Lee. "Music Controller for Real Time Performance." IEEE Instrumentation and Measurement Technology Conference. May 2002. pp. 991-995.

Fahn, Chin-Shyurng and Herman Sun. "Development of a Data Glove with Reducing Sensors Based on Magnetic Induction." IEEE Transactions on Industrial Electronics, vol. 52, no. 2, April 2005. pp. 585-594.

Howard, Pierce J. "Cerebral Palsy." *The Owner's Manual for the Brain*. 2nd Edition. Bard Press: Atlanta, 2000. 303.

Howard, Pierce J. "Music as Psychotherapy." *The Owner's Manual for the Brain*. 2nd Edition. Bard Press: Atlanta, 2000. 188.

Kelker, Katharin A. "Family Guide to Assistive Technology." <http://www.pluk.org/AT1.html#2> Oct. 2005. 1997.

Kirn, Peter. "Hypersense Complex Gestural Glove for Music." Create Digital Music. <http://createdigitalmusic.com/index.php?option=com_content&task=view&id=770&Itemid=44> Oct. 2005.

Krakouer L., Houghton. S., Douglas, G., & West J. "The efficacy of music therapy in effecting behaviour change in persons with Cerebral Palsy." International Journal of Psychosocial Rehabilitation. 2001. 6, 29-37.

Riddell, Alistair, Somaya Langley, and Simon Burton. Hypersense Complex. <http://arrowtheory.com/hypersense/index.html> Oct. 2005.

Su, Yu, et al. “3-D Motion System (“Data-Gloves”): Application for Parkinson’s Disease.” IEEE Transactions on Instrumentation and Measurement. Vol. 52, No 3, pp 662-674, June 2003.

Tarchanidis, Kostas N. and John N. Lygouras. “Data Glove with a Force Sensor.” IEEE Transactions on Instrumentation and Measurement. Vol. 52, No. 3, pp 984-989, June 2003.

Weissmann, John and Ralf Salomon. “Gesture Recognition for Virtual Reality Applications Using Data Gloves and Neural Network.” Neural Networks. International Joint Conference. pp. 2043-2046, 1999.

Zisook, Sheldon O. “Cerebral Palsy Definition – Cerebral Palsy A-Z.” A Resource for Parents and Families <http://www.about-cerebral-palsy.org/definition/index.html> Oct. 2005.

TRANSMITTING DATA CODE
APPENDIX A

```
'
***********************************************************************
***
' MonkeySee.bs2
' A7 Engineering, 2003-2005
'
' This program is used with a Parallax SumoBot robot and an
EmbeddedBlue
' eb500 radio module.
'
' Description
' ===========
' This program demonstrates wireless communications between two eb500
radio
' modules. The program uses the infrared sensors to follow an object
and
' then transmits the movement information to a BOE-Bot equipped with an
eb500
' module.
'
' Instructions
' ============
' 1. Verify that the eb500 module is connected properly to the AppMod
'    header of the SumoBot robot.
' 2. Modify the connection logic to use the correct Bluetooth Address.
' 3. Compile and run this program.
'
'
***********************************************************************
***
'{$STAMP BS2}
'{$PBASIC 2.5}
'I/O Line 5 provides the connection status
INPUT 5

'-------[I/O]----------------------------------------------------
Switch1 PIN 11
Switch2 PIN 12
Switch3 PIN 13
Switch4 PIN 14
Switch5 PIN 15
WhichSwitch VAR Byte

'-----[Variables]------------------------------------------------
SwitchBits    VAR  Nib
bBuffer    VAR  Byte(4)
bErrorCode  VAR  Byte

'------[Initialization]------------------------------------------
'Wait for the eb500 radio to be ready
PAUSE 1000
Connect:
```

```
  'Connect to Monkey-Do
  SEROUT 1,84,["con 00:0C:84:00:0C:90",CR]
  SERIN 0,84,[WAIT("ACK",CR)]
  'Either an Err #<CR> or a ">" will be received
  SERIN 0,84,[STR bBuffer\6\">"]
  IF bBuffer(0) = "E" THEN ErrorCode

WaitForConnection:
  IF IN5 = 0 THEN WaitForConnection

'-----[Main Code]-------------------------------------------
Main:
  'Verify the connection is still up before each loop

  IF IN5 = 0 THEN Connect

  GOSUB PlaySwitch
  BRANCH SwitchBits,[Hold, PlaySwitch5_1, PlaySwitch5_2, PlaySwitch5_3,
PlaySwitch5_4, PlaySwitch1, PlaySwitch2, PlaySwitch3, PlaySwitch4,
PlaySwitch5]

PlaySwitch5_1:
  SEROUT 1,84,["1"]
  SwitchBits = 0
  GOTO Main

PlaySwitch5_2:
  SEROUT 1,84,["2"]
  SwitchBits = 0
  GOTO Main

PlaySwitch5_3:
  SEROUT 1,84,["3"]
  SwitchBits = 0
  GOTO Main

PlaySwitch5_4:
  SEROUT 1,84, ["4"]
  SwitchBits = 0
  GOTO Main

PlaySwitch1:
  SEROUT 1,84,["5"]
  SwitchBits = 0
  GOTO Main

PlaySwitch2:
  SEROUT 1,84,["6"]
  SwitchBits = 0
  GOTO Main

PlaySwitch3:
  SEROUT 1,84,["7"]
  SwitchBits = 0
  GOTO Main

PlaySwitch4:
```

```
  SEROUT 1,84,["8"]
  SwitchBits = 0
  GOTO Main

PlaySwitch5:
  SEROUT 1,84,["9"]
  SwitchBits = 0
  GOTO Main

Hold:
 GOTO Main

'-----[Subroutines]---------------------------------------------
PlaySwitch:


  IF (Switch5=1 AND Switch1=1) THEN
    SwitchBits = 1
  ELSEIF (Switch5=1 AND Switch2=1) THEN
    SwitchBits = 2
  ELSEIF (Switch5=1 AND Switch3=1) THEN
    SwitchBits = 3
  ELSEIF (Switch5=1 AND Switch4=1) THEN
    SwitchBits = 4
  ELSEIF (Switch1 = 1) THEN
    SwitchBits = 5
  ELSEIF (Switch2 = 1) THEN
    SwitchBits = 6
  ELSEIF (Switch3 = 1) THEN
    SwitchBits = 7
  ELSEIF (Switch4 = 1) THEN
    SwitchBits = 8
  ELSEIF (Switch5 = 1) THEN
    SwitchBits = 9
  ELSE
    DEBUG "button not pressed"
  ENDIF
RETURN


BadCommand:
  'DEBUG "A bad command was received."
  'END

ErrorCode:
  bErrorCode = bBuffer(4)
  'DEBUG "An error was received: ",STR bErrorCode,CR
  END
```

RECEIVING DATA CODE
APPENDIX B

```
'
**********************************************************************
***
' MonkeyDo.bs2
' A7 Engineering, 2003-2005
'
' This program is used with a Parallax Boe-Bot robot and an
EmbeddedBlue
' eb500 radio module.
'
' Description
' ===========
' This program demonstrates wireless communications between two eb500
radio
' modules. The program waits for a connection from the remote SumoBot
robot
' and then reads the movement information from the eb500 and performs
the
' specified action.
'
' Instructions
' ============
' 1. Verify that the eb500 module is connected properly to the AppMod
'    header of the Boe-Bot robot.
' 2. Compile and run this program.
'
'
**********************************************************************
***
'{$STAMP BS2}
'{$PBASIC 2.5}

'-----[ Select STAMP for Adjustments]----------------------------------
----------------
'TimeAdj (constant to adjust time)
'FreqAdj (constant to adjust frequency)
#SELECT $STAMP
#CASE BS2, BS2E
TimeAdj CON $100 ' x 1.0
FreqAdj CON $100 ' x 1.0
#CASE BS2SX
TimeAdj CON $280 ' x 2.5
FreqAdj CON $066 ' x 0.4
#CASE BS2P
TimeAdj CON $3C5 ' x 3.77
FreqAdj CON $044 ' x 0.265
#CASE BS2PE
TimeAdj CON $100 ' x 1.0
FreqAdj CON $0AA ' x 0.665
#ENDSELECT

Time CON 150
'-----[I/O Definitions]----------------------------------------
```

```
Speaker PIN 15

'-----[Variables]----------------------------------------------
CmdData    VAR  Byte

'-----[ Note Frequencies ]---------------------------------------------
-----
'Declare constant (CON) frequency for each note.
C  CON  65    'Ridiculously low notes
Db CON  69
D  CON  73
Eb CON  77
E  CON  82
F  CON  87
Gb CON  92
G  CON  97
Ab CON  103
A  CON  110
Bb CON  117
BE CON  124

C1  CON  131    'Very low notes
Db1 CON  139
D1  CON  147
Eb1 CON  154
E1  CON  165
F1  CON  175
Gb1 CON  185
G1  CON  195
Ab1 CON  207
A1  CON  220
Bb1 CON  234
BE1 CON  248

C2  CON  262    'Low notes
Db2 CON  278
D2  CON  294
Eb2 CON  307
E2  CON  329
F2  CON  350
Gb2 CON  370
G2  CON  391
Ab2 CON  414
A2  CON  439
Bb2 CON  467
BE2 CON  495

C3  CON  521    'Middle 'C'
Db3 CON  554
D3  CON  588
Eb3 CON  623
E3  CON  658
F3  CON  694
Gb3 CON  737
G3  CON  781
Ab3 CON  829
A3  CON  877
```

```
Bb3 CON  928
BE3 CON  980

C4  CON  1042    'High notes
Db4 CON  1102
D4  CON  1163
Eb4 CON  1240
E4  CON  1316
F4  CON  1389
Gb4 CON  1476
G4  CON  1563
Ab4 CON  1658
A4  CON  1754
Bb4 CON  1856
BE4 CON  1960

C5  CON  2084    'Very high notes
Db5 CON  2204
D5  CON  2326
Eb5 CON  2480
E5  CON  2632
F5  CON  2778
Gb5 CON  2952
G5  CON  3126
Ab5 CON  3316
A5  CON  3508
Bb5 CON  3712
BE5 CON  3920

C6  CON  4168    'Ridiculously high notes
Db6 CON  4408
D6  CON  4652
Eb6 CON  4960
E6  CON  5264
F6  CON  5556
Gb6 CON  5904
G6  CON  6252
Ab6 CON  6632
A6  CON  7016
Bb6 CON  7424
BE6 CON  7840

'-----[Initialization]-----------------------------------------
Initialize:
  'Wait for the eb500 radio to be ready
  PAUSE 1000
  'Set the initial state to hold
  CmdData = 0

'-----[Main Code]-----------------------------------------------
Main:
  'Wait for a command
  SERIN 0,84,[DEC1 CmdData]

  'Process the command
```

```
   BRANCH CmdData,[Hold, PlaySwitch5_1, PlaySwitch5_2, PlaySwitch5_3,
PlaySwitch5_4, PlaySwitch1, PlaySwitch2, PlaySwitch3, PlaySwitch4,
PlaySwitch5]

  'If the command was invalid, just loop again
  GOTO Main

'PlayBoth:
'  FREQOUT Speaker, Time */ TimeAdj, Db4 */ FreqAdj
'  GOTO Main

'PlayBoth1:
'  FREQOUT Speaker, Time */ TimeAdj, Eb4 */ FreqAdj
'  GOTO Main

PlaySwitch5_1:
  FREQOUT Speaker, Time */ TimeAdj, A4 */ FreqAdj
  GOTO Main

PlaySwitch5_2:
  FREQOUT Speaker, Time */ TimeAdj, BE4 */ FreqAdj
  GOTO Main

PlaySwitch5_3:
  FREQOUT Speaker, Time */TimeAdj, C5 */ FreqAdj
  GOTO Main

PlaySwitch5_4:
  FREQOUT Speaker, Time */ TimeAdj, D5 */ FreqAdj
  GOTO Main

PlaySwitch1:
  FREQOUT Speaker, Time */ TimeAdj, C4 */ FreqAdj
  GOTO Main

PlaySwitch2:
  FREQOUT Speaker, Time */ TimeAdj, D4 */ FreqAdj
  GOTO Main

PlaySwitch3:
  FREQOUT Speaker, Time */TimeAdj, E4 */ FreqAdj
  GOTO Main

PlaySwitch4:
  FREQOUT Speaker, Time */ TimeAdj, F4 */ FreqAdj
  GOTO Main

PlaySwitch5:
  FREQOUT Speaker, Time */ TimeAdj, G4 */ FreqAdj
  GOTO Main

HOLD:
                                GOTO Main
```

PREVIOUS PROJECT SOURCE CODE
APPENDIX C

This is the source code that the previous students had worked on which we will be revising for our senior design.

```
' Music is Possible Final Code
' Kevin Ip and Luis Vicencio
' 5/5/05
' {$STAMP BS2}
' {$PBASIC 2.5}
#SELECT $STAMP
#CASE BS2, BS2E
TmAdj CON $100 ' x 1.0 (time adjust)
FrAdj CON $100 ' x 1.0 (freq adjust)
#CASE BS2SX
TmAdj CON $280 ' x 2.5
FrAdj CON $066 ' x 0.4
#CASE BS2P
TmAdj CON $3C5 ' x 3.77
FrAdj CON $044 ' x 0.265
#CASE BS2PE
TmAdj CON $100 ' x 1.0
FrAdj CON $0AA ' x 0.665
#ENDSELECT
'-----[ Note Frequencies ]--------------------------------------------------
C4 CON 261
D4 CON 293
E4 CON 329
F4 CON 349
G4 CON 392
A4 CON 440
C5 CON 523
D5 CON 587
E5 CON 659
F5 CON 698
G5 CON 783
A5 CON 880
C6 CON 1046
D6 CON 1174
E6 CON 1318
F6 CON 1396
G6 CON 1568
A6 CON 1760
time CON 150 ' time (in ms) the note will be played for
'-----[ Declarations ]------------------------------------------------------
force1 VAR Word ' Stores raw output of Flex sensors
force2 VAR Word
force3 VAR Word
force4 VAR Word
force5 VAR Word
prev1 VAR Word ' Stores previous output of Flex sensors
prev2 VAR Word
prev3 VAR Word
```

```
prev4 VAR Word
prev5 VAR Word
sensor1 PIN 0 ' Flex sensor circuit (sensors 1 through 6)
sensor2 PIN 2
sensor3 PIN 4
sensor4 PIN 6
sensor5 PIN 8
Spkr PIN 15 ' Speaker I/O Pin
diff CON 10
x CON 500
val CON 0
rcmax CON 140
'-----[ Main Routine ]---------------------------------------------------
Init:
prev1 = val
prev2 = val
prev3 = val
prev4 = val
prev5 = val
DEBUG "Initialize!", CR
Display:
HIGH sensor1 'Discharge capacitors
HIGH sensor2
HIGH sensor3
HIGH sensor4
HIGH sensor5
PAUSE 20 'pause for 2 ms
RCTIME sensor1,1,force1 'Measure R/C charge time
RCTIME sensor2,1,force2
RCTIME sensor3,1,force3
RCTIME sensor4,1,force4
RCTIME sensor5,1,force5
DEBUG CLS, "prev1 = ", DEC prev1, CR,
"prev2 = ", DEC prev2, CR,
"prev3 = ", DEC prev3, CR,
"prev4 = ", DEC prev4, CR,
"prev5 = ", DEC prev5, CR,
"S1 = ", DEC force1, CR,
"S2 = ", DEC force2, CR,
"S3 = ", DEC force3, CR,
"S4 = ", DEC force4, CR,
"S5 = ", DEC force5, CR
IF (force1 > prev1) AND (force1-prev1 >= diff) OR (force1 > rcmax) THEN
FREQOUT Spkr, time */ TmAdj, C5 */ FrAdj
DEBUG "S1: ", "*C*", CR
prev1 = force1
ELSE
DEBUG "S1: no", CR
prev1 = force1
ENDIF
IF (force2 > prev2) AND (force2-prev2 >= diff) OR (force2 > rcmax)THEN
FREQOUT Spkr, time */ TmAdj, D5 */ FrAdj
DEBUG "S2: ", "*D*", CR
prev2 = force2
ELSE
DEBUG "S2: no", CR
prev2 = force2
```

```
ENDIF
IF (force3 > prev3) AND (force3-prev3 >= diff) OR (force3 > rcmax)THEN
FREQOUT Spkr, time */ TmAdj, E5 */ FrAdj
DEBUG "S3: ", "*E*", CR
prev3 = force3
ELSE
DEBUG "S3: no", CR
prev3 = force3
ENDIF
IF (force4 > prev4) AND (force4-prev4 >= diff+10) OR (force4 > rcmax)THEN
FREQOUT Spkr, time */ TmAdj, F5 */ FrAdj
DEBUG "S4: ", "*F*", CR
prev4 = force4
ELSE
DEBUG "S4: no", CR
prev4 = force4
ENDIF
IF (force5 > prev5) AND (force5-prev5 >= diff) OR (force5 > rcmax)THEN
FREQOUT Spkr, time */ TmAdj, G5 */ FrAdj
DEBUG "S5: ", "*G*", CR
prev5 = force5
ELSE
DEBUG "S5: no", CR
prev5 = force5
ENDIF
PAUSE x ' pause for x ms before playing the next note
GOTO Display
```

PROJECTED MIDI OUT SOURCE CODE

APPENDIX D

```
'{ $STAMP BS2 }
baudmode CON 32780 '12 + $8000 for BS2
midinoteon CON 144 'MIDI note on status byte
midinoteoff CON 128 'MIDI note off status byte
controlChange CON 176 'MIDI Control Change status byte
pitchBend CON 224 'MIDI Pitch Bend status byte
pressure CON 208 'MIDI pressure status byte
midiPin CON 0 'Pin used for reading flex sensor data
flexmin CON 72 'Value read when flex sensor at rest
flexmax CON 150 'Value read when flex sensor fully bent
dip VAR Word 'For reading dip switch values
value VAR Word 'Store the flex sensor reading
'Main loop
MAIN:
GOSUB READSWITCH
GOSUB READIP
IF dip = 0 THEN MOD
IF dip = 1 THEN PITCH
IF dip = 2 THEN VOLUME
IF dip = 3 THEN EXPRESSION
IF dip = 4 THEN FOOT
IF dip = 5 THEN BREATH
IF dip = 6 THEN DATAENTRY
IF dip = 7 THEN PAN
GOTO MAIN
'Get a reading from the 3 dip switches
READIP:
dip = IN1 + (IN2 * 2) + (IN3 * 4) 'dip value (0 to 7)
RETURN
'Check the reset switch
READSWITCH:
IF IN15 = 1 THEN RESET
RETURN
'Get a reading from the flex sensor
READPIN:
HIGH 0 'discharge cap
PAUSE 1 'for 1 ms
RCTIME 0, 1, value 'read the light sensor
IF value < flexmin THEN FLXMIN 'Check flex sensor data for min
IF value > flexmax THEN FLXMAX 'Check flex sensor data for max
value = (value - flexmin) / 6 'Format data in range 0 to 127
BACKPIN:
RETURN
'Flex sensor data value is below range so set to min (0)
FLXMIN:

value = 0
IF FLEXMIN > 0 THEN
DEBUG "flex on", CR
ELSE
DEBUG "no flex", CR
GOTO BACKPIN
'Flex sensor data value is beyond range so set to max (127)
```

```
FLXMAX:
value = 127
GOTO BACKPIN
'All notes off and reset pitch bend
RESET:
SEROUT 7, baudmode, 0, [controlChange, 121, 0]
SEROUT 7, baudmode, 0, [pitchBend, 127, 63]
PAUSE 200
RETURN
'Output Pitch bend data
PITCH:
GOSUB READPIN
SEROUT midiPin, baudmode, 0, [pitchBend, 127, value]
GOTO MAIN
'Output Modulation data
MOD:
GOSUB READPIN
SEROUT midiPin, baudmode, 0, [controlChange, 1, value]
GOTO MAIN
'Output Volume data
VOLUME:
GOSUB READPIN
SEROUT midiPin, baudmode, 0, [controlChange, 7, value]
GOTO MAIN
'Output Expression data
EXPRESSION:
GOSUB READPIN
SEROUT midiPin, baudmode, 0, [controlChange, 11, value]
GOTO MAIN
'Output Foot Pedal data
FOOT:
GOSUB READPIN
SEROUT midiPin, baudmode, 0, [controlChange, 4, value]
GOTO MAIN
'Output Breath control data
BREATH:
GOSUB READPIN
SEROUT midiPin, baudmode, 0, [controlChange, 2, value]
GOTO MAIN

'Output Data Entry data
DATAENTRY:
GOSUB READPIN
SEROUT midiPin, baudmode, 0, [controlChange, 6, value]
GOTO MAIN
'Output Pan data
PAN:
GOSUB READPIN
SEROUT midiPin, baudmode, 0, [controlChange, 10, value]
GOTO MAIN
```

PROJECT TEST CODE
APPENDIX E

```
'Music is Possible
' Trang A Phan and Joshua Yee
' 1/19/06

' {$STAMP BS2}
' {$PBASIC 2.5}

duration VAR Word
frequency VAR Word

'pushbuttons'
DO

DEBUG ? IN0
IF IN0=1   THEN       DEBUG "Do...", CR: FREQOUT 5, 500 ,1047 ' C6

DEBUG ? IN1
IF IN1=1   THEN       DEBUG "Re...", CR: FREQOUT 5, 500 ,1175 ' D6

DEBUG ? IN2
IF IN2=1   THEN       DEBUG "Mi...", CR: FREQOUT 5, 500 ,1319 ' E6

DEBUG ? IN3
IF IN3=1   THEN       DEBUG "Fa...", CR: FREQOUT 5, 500 ,1396 ' F6

DEBUG ? IN4
IF IN4=1  THEN        DEBUG "Sol..", CR: FREQOUT 5, 500,1568 ' G6

DEBUG ? IN5
IF IN5=1  THEN        DEBUG "La...", CR: FREQOUT 5,500,1760 ' A6

DEBUG ? IN6
IF IN6=1  THEN        DEBUG "C#/Db...", CR: FREQOUT 5, 500, 1109

DEBUG ? IN7
IF IN7=1  THEN        DEBUG "D#/Eb...", CR: FREQOUT 5, 500, 1245

DEBUG ? IN8
IF IN8=1  THEN        DEBUG "F#/Gb...", CR: FREQOUT 5, 500, 1480

DEBUG ? IN9
IF IN9=1  THEN        DEBUG "G#/Ab...", CR: FREQOUT 5, 500, 1661

DEBUG ? IN10
IF IN10=1 THEN        DEBUG "A#/Bb...", CR: FREQOUT 5, 500, 1865

DEBUG ? IN11
IF IN11=1 THEN        DEBUG "Ti...", CR: FREQOUT 5, 500, 1976

LOOP
```

BASIC STAMP SPECIFICATIONS
APPENDIX F

Stamp Specifications (revised 04/05)

Released Products	Rev.Dx / BS1-IC	BS2-IC	BS2e-IC	BS2sx-IC
Package	PCB w/Proto / 14-pin SIP	24-pin DIP	24-pin DIP	24-pin DIP
Package Size (L x W x H)	2.5" x 1.5" x .5" / 1.4" x .6" x .1"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"
Environment *	0° - 70° C (32° - 158° F) **	0° - 70° C (32° - 158° F) **	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)
Microcontroller	Microchip PIC16C56a	Microchip PIC16C57c	Ubicom SX28AC	Ubicom SX28AC
Processor Speed	4 MHz	20 MHz	20 MHz	50 MHz
Program Execution Speed	~2,000 instructions/sec.	~4,000 instructions/sec.	~4,000 instructions/sec.	~10,000 instructions/sec.
RAM Size	16 Bytes (2 I/O, 14 Variable)	32 Bytes (6 I/O, 26 Variable)	32 Bytes (6 I/O, 26 Variable)	32 Bytes (6 I/O, 26 Variable)
Scratch Pad RAM	N/A	N/A	64 Bytes	64 Bytes
EEPROM (Program) Size	256 Bytes, ~80 instructions	2K Bytes, ~500 instructions	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.
Number of I/O pins	8	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial
Voltage Requirements	5 - 15 vdc	5 - 15 vdc	5 - 12 vdc	5 - 12 vdc
Current Draw @ 5V	1 mA Run / 25 µA Sleep	3 mA Run / 50 µA Sleep	25 mA Run / 200 µA Sleep	60 mA Run / 500 µA Sleep
Source / Sink Current per I/O	20 mA / 25 mA	20 mA / 25 mA	30 mA / 30 mA	30 mA / 30 mA
Source / Sink Current per unit	40 mA / 50 mA	40 mA / 50 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins
PBASIC Commands***	32	42	45	45
PC Programming Interface	Serial (w/BS1 Serial Adapter)	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)
Windows Text Editor	Stampw.exe (v2.1 and up)	Stampw.exe (v1.04 and up)	Stampw.exe (v1.096 and up)	Stampw.exe (v1.091 and up)

Released Products	BS2p24-IC	BS2p40-IC	BS2pe-IC	BS2px-IC	Javelin Stamp
Package	24-pin DIP	40-pin DIP	24-pin DIP	24-pin DIP	24-pin DIP
Package Size (L x W x H)	1.2" x 0.6" x 0.4"	2.1" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.2" x 0.6" x 0.4"	1.24" x 0.60" x 0.45"
Environment *	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)	0° - 70° C (32° - 158° F)
Microcontroller	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC	Ubicom SX48AC
Processor Speed	20 MHz Turbo	20 MHz Turbo	8 MHz Turbo	32 MHz Turbo	25 MHz Turbo
Program Execution Speed	~12,000 instructions/sec.	~12,000 instructions/sec.	~6000/sec.	~19,000 instructions/sec.	~8,500 instructions/sec.
RAM Size	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	38 Bytes (12 I/O, 26 Variable)	32768 Bytes
Scratch Pad RAM	128 Bytes	128 Bytes	128 Bytes	128 Bytes	N/A
EEPROM (Program) Size	8 x 2K Bytes, ~4,000 inst.	8 x 2K Bytes, ~4,000 inst.	16 x 2K Bytes (16 K for source)	8 x 2K Bytes, ~4,000 inst.	32768 Bytes
Number of I/O pins	16 + 2 Dedicated Serial	32 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16 + 2 Dedicated Serial	16
Voltage Requirements	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 12 vdc	5 - 24 vdc
Current Draw @ 5V	40 mA Run / 350 µA Sleep	40 mA Run / 350 µA Sleep	15 mA Run / 36 µA Sleep	55 mA Run / 450 µA Sleep	80 mA Run / No Sleep
Source / Sink Current per I/O	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA	30 mA / 30 mA
Source / Sink Current per unit	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins	60 mA / 60 mA per 8 I/O pins
PBASIC Commands***	61	61	61	63	0 (Java)
PC Programming Interface	Serial (9600 baud)	Serial (9600 baud)	Serial (9600 baud)	Serial (19200 baud)	Serial (28800 baud)
Windows Text Editor	Stampw.exe (v1.1 and up)	Stampw.exe (v1.1 and up)	Stampw.exe (v1.33 and up)	Stampw.exe (v2.2 and up)	Javelin Stamp IDE

* 70% Non-Condensing Humidity

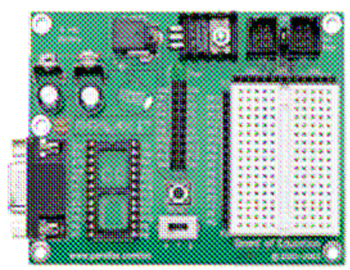
** Industrial Models Available, -40° - 85° C (-40° - 185° F). Contact Parallax Sales for information.

*** Using PBASIC 2.5 for BS2-type models.

BOARD OF EDUCATION DEVELOPMENT BOARD
APPENDIX G

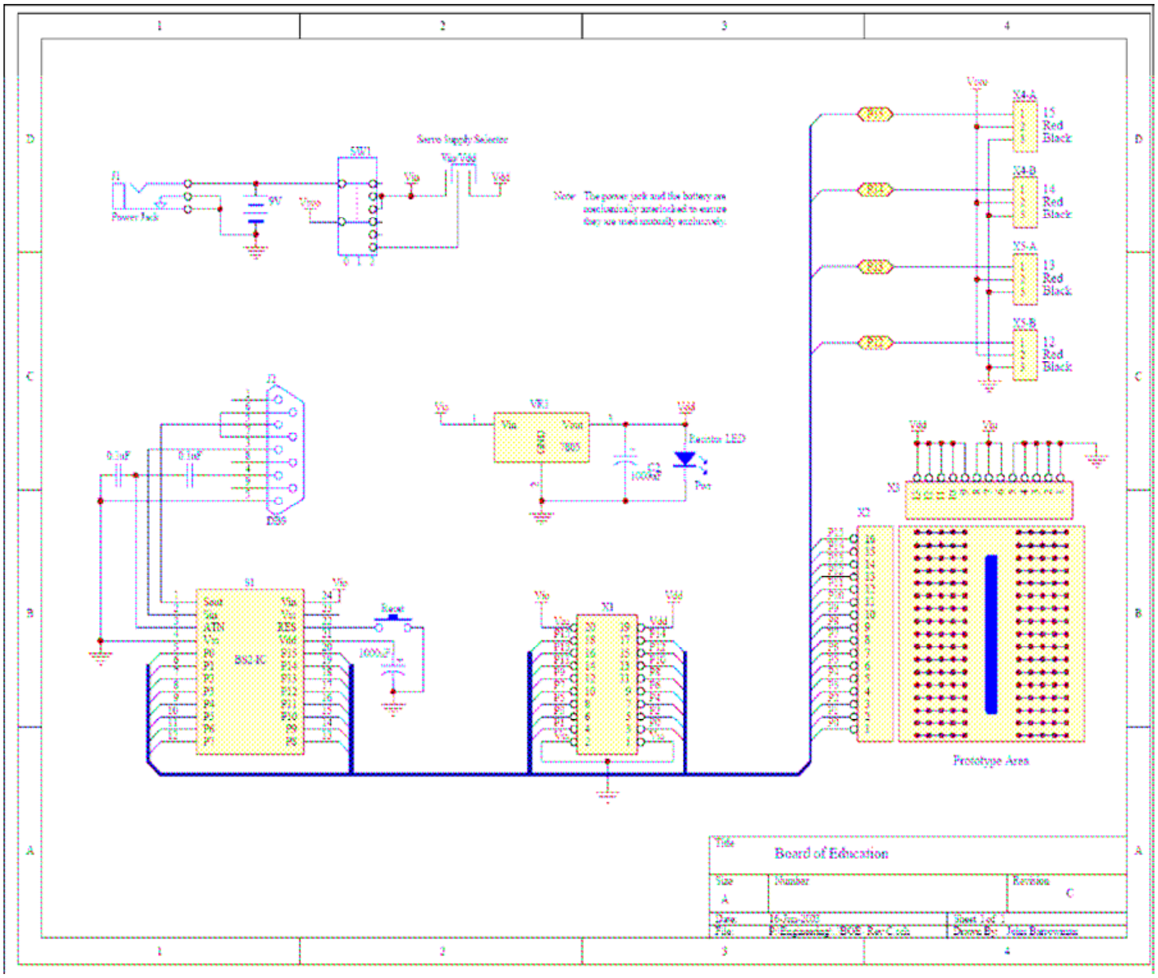
Main Features:

Mechanically interlocked power supply to prevent dual connection of wall pack and 9 volt battery
DB9 connector for BS2-IC programming and serial communications during run time
P0 - P15 I/O pins, Vdd and Vss connections brought adjacent to 5.1 x 3.5 cm (2" x 1 3/8) breadboard area
Includes set of ten (10) color coded 22 gauge wires
Female 10-pin dual row connector for optional AppMods (more breadboard space)
Traces on top of the board show connections between BS2-IC and Breadboard connections



BASIC STAMP II SCHEMATIC

APPENDIX H



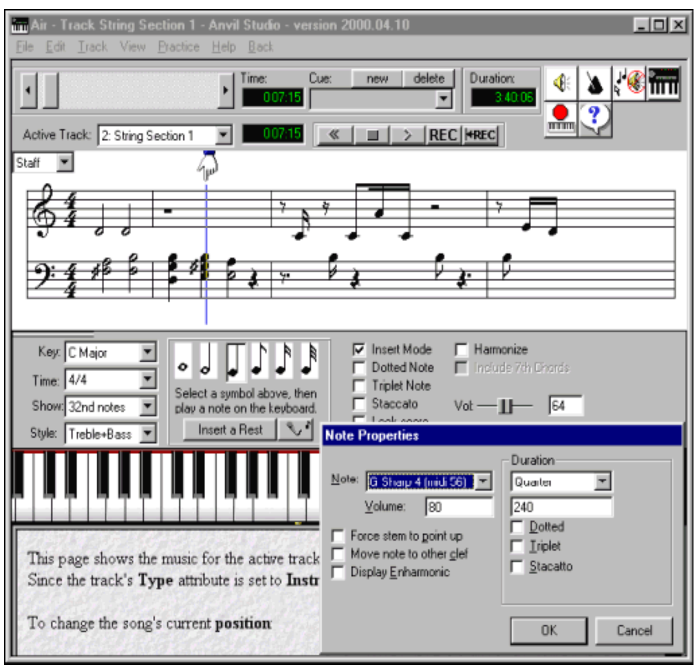
ANVIL STUDIO SOFTWARE
APPENDIX I

Anvil Studio is a free Windows 95/NT program for multi-track recording, composing, and editing of music using audio, MIDI, and sampled percussion sounds. Audio effects include delay, pitch change, volume change, filtering, and reverse.

Anvil Studio is designed to work with several Optional Accessories that increase functionality and performance. Examples of accessories include Sheet Music printing, piano exercises, ear training exercises, and voice-pitch analyzer.

Features

- Multiple Editors including Audio, Events, Score, Drum, Piano Roll
- Piano Roll and Drum editor allow measures to loop (repeat a number of times)
- Virtually unlimited number of MIDI Tracks
- MIDI Quantize, diatonic and chromatic transpose, auto-harmonize
- Rhythm tracks can include sampled audio percussion sounds
- Record and edit a single stereo/mono, one minute audio track with the free version or up to 8 stereo/mono tracks with the optional $19 accessory
- Audio effects: delay, pitch change, volume change, pan, filtering, and reverse.
- Mix audio tracks and sampled rhythms to new .WAV file.
- Perform page lets you wire together on-screen modules for real-time manipulation of MIDI sounds
- Control Record/Play/Stop functions remotely using any MIDI controllers such as a Foot Switch, Modulation wheel.
- Includes extensive integrated help



EMBEDDEDBLUE EB500 APPMOD TRANSCEIVER
APPENDIX J

For this project, one of the major components that we are dealing with is the wireless transceiver. We are using two EmbeddedBlue 500 transceivers for this project: one for the switch button inputs and the second one for the audio output. Below are the dimensions of each transceiver. This is useful information to make sure that the transceiver would fit inside the enclosed "instruments."

Dimensions

The dimensions of the eb500 are shown below in Table 3. Please reference Figure 18 to locate the referenced dimension on the eb500.

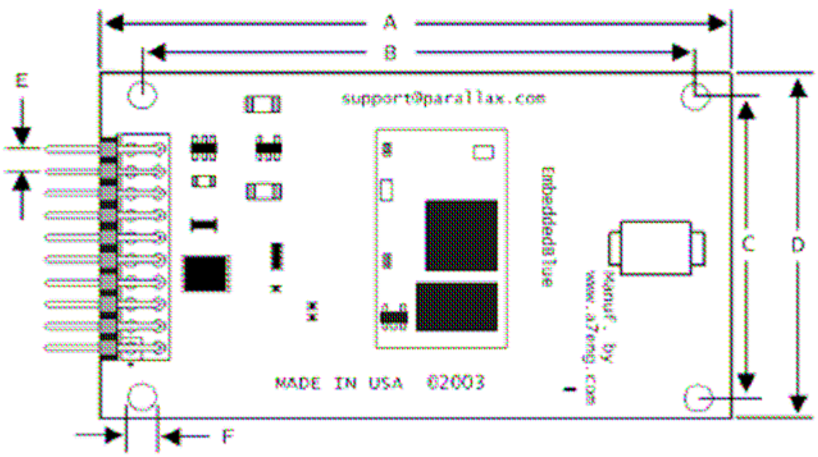


Figure 18: eb500 Dimensions

Dimension	inches	mm
A	2.75	69.85
B	2.40	60.96
C	1.30	33.02
D	1.60	40.64
E	0.10	2.54
F	0.125	3.20

Table 3: eb500 Dimensions

**From the Embedded Blue 500 User Manual*

Below is the specification that would be useful for our project. It is important that we are aware of these specifications to know the power and current consumptions. We found out that running the BASIC Stamp II with the transceiver requires more power than without. We need to analyze and test out a way to reduce the power consumption.

Operating Parameters

The operating parameters of the eb500 are shown below in Table 2.

Transmit Power	4dBm (max) class 2 operation	
Open Field Range	eb500-AHC-IN (surface mount antenna) – 100 meters (328 feet) *(Actual range is dependent upon location and environment.)*	
Receiver Sensitivity	-85dBm	
Operating Temp.	-15° to 70°C	
Supply Power	5 to 12VDC	
Current Consumption	115.2kbps data transfer: 35mA 38.4kbps data transfer: 30mA 9.6kbps data transfer: 25mA	connected and idle: 8mA no connection: 3mA
Interfaces	5V logic level UART or RS232 with optional eb600 adapter Baud rate 9.6k – 230.4k Flow control: RTS/CTS or none	
Connector	One 10x2 AppMod compatible 20 pin 0.1" header	
Antenna	Matched internal surface mount	
Bluetooth Support	Version 1.2 compliant with profiles L2CAP, RFCOMM, SDP, SPP	
Firmware	Upgradeable via PC application with eb600 adapter	

Table 2: eb500 Operating Parameters

**From the Embedded Blue 500 User Manual*

In the chart below, are the pins that we need to connect from the transceiver and the parallax BASIC stamp board. In order to transmit and receive data, we need to be sure that pins 0, 1, 2, 5, and 6 are connected. This leaves us pins 4 and 7-15 to use for the switches for one transceiver. For the audio end, we do not need as many pins, so the transceiver would not be a problem.

Pin out

The eb500 module features a 20 pin connector with 0.1" spacing for direct connection to a Parallax AppMod header. Currently, nine of the pins are in use (seven when flow control is set to none). The other pins are reserved for future use.

Pin	Parallax Pin	Function	Description	Usage
CN1 - 1	GND	GND	Ground	Required
CN1 - 2	GND	GND	Ground	Required
CN1 - 3	P0	TX	Serial Transmit line from eb500	Required
CN1 - 4	P1	RX	Serial Receive line to eb500	Required
CN1 - 5	P2	RTS	Request-to-Send on the serial port interface between the eb500 and the BASIC Stamp	Optional
CN1 - 6	P3	CTS	Clear-to-Send on the serial port interface between the eb500 and the BASIC Stamp	Optional
CN1 - 8	P5	Status	Bluetooth connection status (0 = not connected, 1 = connected)	Required
CN1 – 9	P6	Mode	Command/data mode toggle (0 = command, 1 = data)	Required
CN1 - 20	VCC	VCC	Power	Required

Table 4: eb500 Pin out Description

**From the Embedded Blue 500 User Manual*

Throughout the time we were working on the transceiver of the wireless, we have encountered several problems: connection failed, connection dropped, data not transmitted/received, etc. In the table below, there are lists of errors that we have faced or might possibly face while working with the wireless transceiver.

While using the eb500 you may encounter an error. Below is a listing of all eb500 error codes with a description of what causes the error to occur.

Error Code	Description
1	General connection failure.
2	Connection attempt failed. This error occurs when attempting to connect with an invalid Bluetooth address or a device that is not available.
3	Command not valid while active. This error occurs when there is an active connection and a command is issued that is not valid while connected with a remote device.
4	Command only valid while active. This error occurs when there is not an active connection and a command is issued that is only valid while connected with a remote device.
5	An unexpected request occurred. This error occurs when the remote device makes an invalid request. This is typically seen with older Bluetooth devices that may have errors in their firmware.
6	Connection attempt failed due to a timeout.
7	Connection attempt was refused by the remote device. This error typically occurs when the security settings of the remote and local device are incompatible. It can also occur when establishing a connection with security set to open if the remote and local passkeys do not match.
8	Connection attempt failed because the remote device does not support the Serial Port Profile.
9	An unexpected error occurred when deleting trusted devices.
10	Unable to add a new trusted device. This error will occur if you attempt to have more than twenty five simultaneously trusted devices.
11	Trusted device not found. This error occurs when the trusted device address is not recognized.
12	Command not valid during startup. This error occurs when a command has been issued before the EmbeddedBlue module is fully powered up and initialized.

In addition, earlier in the project we have encountered problems that may have "killed" the transceivers. We figured out that using the adapter would be too much power for the transceiver. In addition, we had to make sure that pins were connected to corresponding pins on the OEM module in a fashion similar to the BOE PC board layout.

Board Of Education Board

Figure 3: Board of Education Board

Prior to installing the eb500 module into the AppMod header, please ensure that the Stamp IO line P5 is not configured as an output. Failure to do so may result in damage to the eb500 module.

The Board Of Education (BOE) contains an AppMod header and supports a direct connection with the eb500 module. On the Board of Education, the AppMod header is labeled X1. When inserting the eb500 module into the Board of Education AppMod header, assure that you insert Pin 1 of the eb500 module, marked with a white dot and a square, into the VSS pin of the AppMod header as pictured in Figure 3.

Step 1: Insert the eb500 Modules into the BOE and SumoBoard Boards
In this step we will insert the eb500 modules into the Board of Education (BOE) and SumoBoard boards.
1. Insert an **eb500** module into the **AppMod header** of the Board of Education board; assuring that Pin 1 of the eb500 module is inserted into the VSS pin of the AppMod header.
2. Insert an **eb500** module into the **AppMod header** of the SumoBoard board; assuring that Pin 1 of the eb500 module is inserted into the VSS pin of the AppMod header.

Step 2: Write a BASIC Stamp Application to Get the eb500 Address
In this step we will write a BASIC Stamp application to interrogate an eb500 for its unique Bluetooth address.
1. Open the **BASIC Stamp Editor**.
2. Enter the following **program code** into the editor. This application is available in electronic form on the accompanying CD, in the Samples folder, in the file GetAddress.bs2.

```
'{$STAMP BS2}
szData VAR BYTE(20)
'Wait for the eb500 radio to be ready
PAUSE 1000
'Get the eb500 Bluetooth Address
SEROUT 1,84,["get address",CR]
SERIN 0,84,[WAIT("ACK",CR)]
'Read the local address from the get command
SERIN 0,84,[STR szData\17]
SERIN 0,84,[WAIT(CR,">")]
szData(17) = 0
DEBUG "Local eb500 address: ",STR szData\17,CR
```

The BASIC Stamp application issues an eb500 Get Address command and then reads and displays the response in the debug window. The response is the Bluetooth address of the local eb500 module.
3. On the **File** menu, click **Save As**.
4. In the **File name** box, enter a file name to which to save the program just created. For example, GetAddress.bs2.
5. Click **Save**.

Step 3: Get the Address of the eb500 on the Board of Education Board
In this step we will get the Bluetooth address of the eb500 module on the Board of Education board. We will then use this address in the next step.
1. Connect the **Board of Education board serial port** to the **PC**.
2. Apply **power** to the Board of Education board.
3. On the **Run** menu, click **Run**.
The Bluetooth address for the eb500 on the Board of Education board is shown in the debug window (Figure 9).
4. On the **Debug Terminal #1** dialog click **Close**.
5. Disconnect the **power** from the **Board of Education board**.
6. Disconnect the **Board of Education board serial port** from the **PC**.

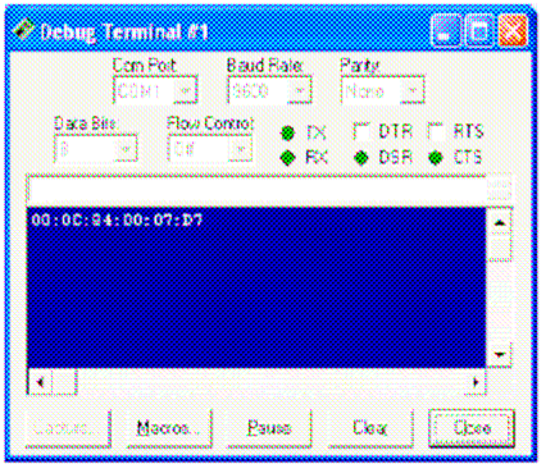


Figure 9: eb500 Bluetooth Address Output

Step 4: Connect the eb500 on the SumoBoard to the eb500 on the BOE
In this step we will develop and run a BASIC Stamp application on the SumoBoard to establish a connection with the Board of Education.
1. Using the BASIC Stamp Editor; on the **File** menu, click **New**.
This will create a new project window within the BASIC Stamp Editor.
2. Enter the following **program code** into the editor, replacing the Bluetooth device address with the device address of the eb500 on the Board of Education board, which we obtained in the previous step. This application is available in electronic form on the accompanying CD, in the Samples folder, in the file Connect.bs2.

```
{$STAMP BS2}
'Wait for the eb500 radio to be ready
PAUSE 1000
'Connect to the remote device
SEROUT 1,84,["con 00:0C:84:00:05:29",CR]
SERIN 0,84,[WAIT("ACK",CR)]
'Wait for the connection to be established and switch to data mode
WaitForConnection:
IF in5 = 0 THEN WaitForConnection
HIGH 6
PAUSE 300
'Wait for 20 seconds
PAUSE 20000
'Switch to Command Mode
LOW 6
SERIN 0,84,[WAIT(CR,">")]
'Disconnect from the remote device
SEROUT 1,84,["dis",CR]
SERIN 0,84,[WAIT(CR,">")]
```

The BASIC Stamp application establishes a connection with the remote Bluetooth device, waits twenty seconds, switches back to command mode and then disconnects from the remote device.
3. On the **File** menu, click **Save As**.
4. In the **File name** box, enter a file name to which to save the program just created. For example, Connect.bs2.
5. Click **Save**.
6. Apply **power** to the **Board of Education board**.
7. Apply **power** to the **SumoBoard board**.
8. On the **Run** menu, click **Run**.
The Connection Status LED (see Figure 1 on page 15) on both eb500 modules will turn on when a connection is established between the two eb500 modules.
9. Disconnect the **power** from the **Board of Education board**.
10. Disconnect **power** from the **SumoBoard board**.

Communicating between Two eb500 Modules

After we are able to connect the two eb500 modules, we went through the following steps to be able to have the transceivers to communicate. One eb500 module is connected to the OEM with the switches, and the other transceiver is connected to the BASIC stamp with the audio output.

We used the procedures of Monkey-See and Monkey-Do as a template.

Step 1: Create a Monkey-See Application for the SumoBot

In this step we will create a BASIC Stamp application that will use the infrared sensors of the SumoBot to follow an object and transmit its movements to a remote eb500.

1. Open the **BASIC Stamp Editor**.
2. Enter the following **program code** into the editor, replacing the Bluetooth device address with the device address of the eb500 inserted into the Boe-Bot robot. This application is available in electronic form on the accompanying CD, in the Samples folder, in the file MonkeySee.bs2.

```
'{$STAMP BS2}
'{$PBASIC 2.5}
'I/O Line 5 provides the connection status
INPUT 5

'-------[I/O]--------------------------------------------------
Switch1 PIN 11
Switch2 PIN 12
Switch3 PIN 13
Switch4 PIN 14
Switch5 PIN 15
WhichSwitch VAR Byte

'-----[Variables]----------------------------------------------
SwitchBits    VAR  Nib
bBuffer    VAR  Byte(4)
bErrorCode  VAR  Byte

'------[Initialization]-----------------------------------------
'Wait for the eb500 radio to be ready
PAUSE 1000
Connect:
  'Connect to Monkey-Do
  SEROUT 1,84,["con 00:0C:84:00:0C:90",CR]
  SERIN 0,84,[WAIT("ACK",CR)]
  'Either an Err #<CR> or a ">" will be received
  SERIN 0,84,[STR bBuffer\6\">"]
  IF bBuffer(0) = "E" THEN ErrorCode

WaitForConnection:
  IF IN5 = 0 THEN WaitForConnection

'-----[Main Code]----------------------------------------------
Main:
  'Verify the connection is still up before each loop

  IF IN5 = 0 THEN Connect

  GOSUB PlaySwitch
  BRANCH SwitchBits,[Hold, PlaySwitch5_1, PlaySwitch5_2, PlaySwitch5_3, PlaySwitch5_4,
PlaySwitch1, PlaySwitch2, PlaySwitch3, PlaySwitch4, PlaySwitch5]

'PlayBoth:
'  SEROUT 1,84,["1"]
'  SwitchBits=0
'  GOTO Main
```

```
'PlayBoth1:
'  SEROUT 1,84,["2"]
'  SwitchBits=0
'  GOTO Main

PlaySwitch5_1:
  SEROUT 1,84,["1"]
  SwitchBits = 0
  GOTO Main

PlaySwitch5_2:
  SEROUT 1,84,["2"]
  SwitchBits = 0
  GOTO Main

PlaySwitch5_3:
  SEROUT 1,84,["3"]
  SwitchBits = 0
  GOTO Main

PlaySwitch5_4:
  SEROUT 1,84, ["4"]
  SwitchBits = 0
  GOTO Main

PlaySwitch1:
  SEROUT 1,84,["5"]
  SwitchBits = 0
  GOTO Main

PlaySwitch2:
  SEROUT 1,84,["6"]
  SwitchBits = 0
  GOTO Main

PlaySwitch3:
  SEROUT 1,84,["7"]
  SwitchBits = 0
  GOTO Main

PlaySwitch4:
  SEROUT 1,84,["8"]
  SwitchBits = 0
  GOTO Main

PlaySwitch5:
  SEROUT 1,84,["9"]
  SwitchBits = 0
  GOTO Main



Hold:
  GOTO Main


'-----[Subroutines]---------------------------------------------
PlaySwitch:

'  IF (Switch1=1 AND Switch2=1) THEN
'    SwitchBits = 1
 ' ELSEIF (Switch2=1 AND Switch3=1) THEN
'  SwitchBits = 2
  IF (Switch5=1 AND Switch1=1) THEN
    SwitchBits = 1
  ELSEIF (Switch5=1 AND Switch2=1) THEN
    SwitchBits = 2
  ELSEIF (Switch5=1 AND Switch3=1) THEN
    SwitchBits = 3
  ELSEIF (Switch5=1 AND Switch4=1) THEN
    SwitchBits = 4
  ELSEIF (Switch1 = 1) THEN
    SwitchBits = 5
  ELSEIF (Switch2 = 1) THEN
    SwitchBits = 6
  ELSEIF (Switch3 = 1) THEN
    SwitchBits = 7
  ELSEIF (Switch4 = 1) THEN
    SwitchBits = 8
  ELSEIF (Switch5 = 1) THEN
    SwitchBits = 9
```

```
  ELSE
    DEBUG "button not pressed"
  ENDIF
RETURN


BadCommand:
  'DEBUG "A bad command was received."
  'END

ErrorCode:
  bErrorCode = bBuffer(4)
  'DEBUG "An error was received: ",STR bErrorCode,CR
  END
```

3. On the **File** menu, click **Save As**.
4. In the **File name** box, enter a file name to which to save the program just created. For example, MonkeySee.bs2.
5. Click **Save**.

Step 2: Create a Monkey-Do Application for the Boe-Bot

In this step we will create a BASIC Stamp application that will receive information from the remote SumoBot and perform movements based on that information.
1. On the **File** menu, click **New**.
This will create a new project window within the BASIC Stamp Editor.
2. Enter the following **program code** into the editor. This application is available in electronic form on the accompanying CD, in the Samples folder, in the file

```
'{$STAMP BS2}
'{$PBASIC 2.5}

'-----[ Select STAMP for Adjustments]--------------------------------------------------
'TimeAdj (constant to adjust time)
'FreqAdj (constant to adjust frequency)
#SELECT $STAMP
#CASE BS2, BS2E
TimeAdj CON $100 ' x 1.0
FreqAdj CON $100 ' x 1.0
#CASE BS2SX
TimeAdj CON $280 ' x 2.5
FreqAdj CON $066 ' x 0.4
#CASE BS2P
TimeAdj CON $3C5 ' x 3.77
FreqAdj CON $044 ' x 0.265
#CASE BS2PE
TimeAdj CON $100 ' x 1.0
FreqAdj CON $0AA ' x 0.665
#ENDSELECT

Time CON 150
'-----[I/O Definitions]-----------------------------------------
Speaker PIN 15

'-----[Variables]-----------------------------------------------
CmdData    VAR  Byte

'-----[ Note Frequencies ]-------------------------------------------------
'Declare constant (CON) frequency for each note.
C  CON  65    'Ridiculously low notes
Db CON  69
D  CON  73
Eb CON  77
E  CON  82
F  CON  87
Gb CON  92
G  CON  97
Ab CON  103
A  CON  110
Bb CON  117
BE CON  124

C1  CON  131    'Very low notes
Db1 CON  139
D1  CON  147
```

```
Eb1 CON  154
E1  CON  165
F1  CON  175
Gb1 CON  185
G1  CON  195
Ab1 CON  207
A1  CON  220
Bb1 CON  234
BE1 CON  248

C2  CON  262    'Low notes
Db2 CON  278
D2  CON  294
Eb2 CON  307
E2  CON  329
F2  CON  350
Gb2 CON  370
G2  CON  391
Ab2 CON  414
A2  CON  439
Bb2 CON  467
BE2 CON  495

C3  CON  521    'Middle 'C'
Db3 CON  554
D3  CON  588
Eb3 CON  623
E3  CON  658
F3  CON  694
Gb3 CON  737
G3  CON  781
Ab3 CON  829
A3  CON  877
Bb3 CON  928
BE3 CON  980

C4  CON  1042    'High notes
Db4 CON  1102
D4  CON  1163
Eb4 CON  1240
E4  CON  1316
F4  CON  1389
Gb4 CON  1476
G4  CON  1563
Ab4 CON  1658
A4  CON  1754
Bb4 CON  1856
BE4 CON  1960

C5  CON  2084    'Very high notes
Db5 CON  2204
D5  CON  2326
Eb5 CON  2480
E5  CON  2632
F5  CON  2778
Gb5 CON  2952
G5  CON  3126
Ab5 CON  3316
A5  CON  3508
Bb5 CON  3712
BE5 CON  3920

C6  CON  4168    'Ridiculously high notes
Db6 CON  4408
D6  CON  4652
Eb6 CON  4960
E6  CON  5264
F6  CON  5556
Gb6 CON  5904
G6  CON  6252
Ab6 CON  6632
A6  CON  7016
Bb6 CON  7424
BE6 CON  7840

'-----[Initialization]------------------------------------------
Initialize:
  'Wait for the eb500 radio to be ready
  PAUSE 1000
  'Set the initial state to hold
  CmdData = 0
```

```
'-----[Main Code]-----------------------------------------------
Main:
  'Wait for a command
  SERIN 0,84,[DEC1 CmdData]

  'Process the command
   BRANCH CmdData,[Hold, PlaySwitch5_1, PlaySwitch5_2, PlaySwitch5_3, PlaySwitch5_4,
PlaySwitch1, PlaySwitch2, PlaySwitch3, PlaySwitch4, PlaySwitch5]

  'If the command was invalid, just loop again
  GOTO Main

'PlayBoth:
'  FREQOUT Speaker, Time */ TimeAdj, Db4 */ FreqAdj
'  GOTO Main

'PlayBoth1:
'  FREQOUT Speaker, Time */ TimeAdj, Eb4 */ FreqAdj
'  GOTO Main

PlaySwitch5_1:
  FREQOUT Speaker, Time */ TimeAdj, A4 */ FreqAdj
  GOTO Main

PlaySwitch5_2:
  FREQOUT Speaker, Time */ TimeAdj, BE4 */ FreqAdj
  GOTO Main

PlaySwitch5_3:
  FREQOUT Speaker, Time */TimeAdj, C5 */ FreqAdj
  GOTO Main

PlaySwitch5_4:
  FREQOUT Speaker, Time */ TimeAdj, D5 */ FreqAdj
  GOTO Main

PlaySwitch1:
  FREQOUT Speaker, Time */ TimeAdj, C4 */ FreqAdj
  GOTO Main

PlaySwitch2:
  FREQOUT Speaker, Time */ TimeAdj, D4 */ FreqAdj
  GOTO Main

PlaySwitch3:
  FREQOUT Speaker, Time */TimeAdj, E4 */ FreqAdj
  GOTO Main

PlaySwitch4:
  FREQOUT Speaker, Time */ TimeAdj, F4 */ FreqAdj
  GOTO Main

PlaySwitch5:
  FREQOUT Speaker, Time */ TimeAdj, G4 */ FreqAdj
  GOTO Main

HOLD:
  GOTO Main
```

3. On the **File** menu, click **Save As**.
4. In the **File name** box, enter a file name to which to save the program just created. For example, MonkeyDo.bs2.
5. Click **Save**.

Step 3: Download the Applications to the Robots

In this step we will download the applications we just created to the respective robots.
1. Click the **MonkeySee.bs2** tab in the BASIC Stamp Editor.
2. Connect the **SumoBoard board serial port** to the **PC**.
3. Apply **power** to the **SumoBoard board**.
4. On the **Run** menu, click **Run**.
5. On the **Debug Terminal #1** dialog click **Close**.
6. Disconnect the **power** from the **SumoBoard board**.
7. Disconnect the **SumoBoard board serial port** from the **PC**.

8. Click the **MonkeyDo.bs2 tab** in the BASIC Stamp Editor.
9. Connect the **Board of Education board serial port** to the **PC**.
10. Apply **power** to the **Board of Education board**.
11. On the **Run** menu, click **Run**.
12. Disconnect the **Board of Education board serial port** from the **PC**.

Step 4: Run the Monkey-See / Monkey-Do Applications

In this step we will run the Monkey-See / Monkey-Do applications.
1. Apply **power** to the **SumoBoard board**.
2. Make the **Boe-Bot** robot **mimic the movements** of the SumoBot by putting your hand in front of the SumoBot IR sensors.

PUSHBUTTON CIRCUITRY
APPENDIX K

How the Pushbutton Circuit Works

The left side of Figure 3-4 shows what happens when the pushbutton is not pressed. The LED circuit is not connected to Vdd. It is an open circuit that cannot conduct current. By pressing the pushbutton, you close the connection between the terminals with conductive metal making a pathway for electrons to flow through the circuit.

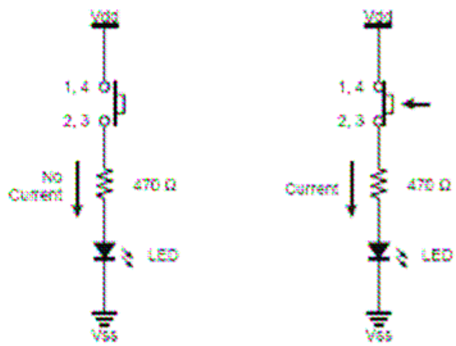


Figure 3-4
Pushbutton Not Pressed, and Pressed

Pushbutton circuit open (left) and closed (right)

Page 76 · What's a Microcontroller?

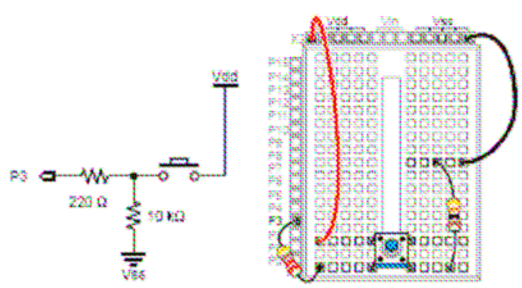


Figure 3-6
Pushbutton Circuit Connected to I/O Pin P3

On the wiring diagram, the 220 Ω resistor is on the left side connecting the pushbutton to P3 while the 10 kΩ resistor is on the right, connecting the pushbutton circuit to Vss.

Figure 3-7 shows what the BASIC Stamp sees when the button is pressed, and when it's not pressed. When the pushbutton is pressed, the BASIC Stamp senses that Vdd is connected to P3. Inside the BASIC Stamp, this causes it to place the number 1 in a part of its memory that stores information about its I/O pins. When the pushbutton is not pressed, the BASIC Stamp cannot sense Vdd, but it can sense Vss through the 10 kΩ and 220 Ω resistors. This causes it to store the number 0 in that same memory location that stored a 1 when the pushbutton was pressed.

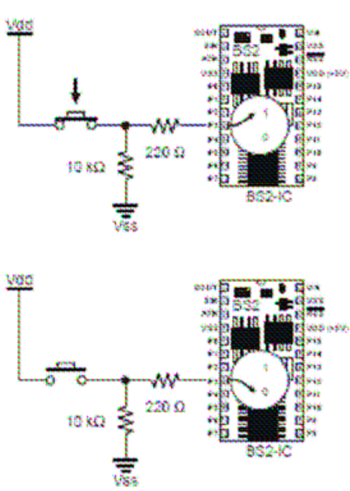


Figure 3-7
BASIC Stamp Reading a Pushbutton

When the pushbutton is pressed, the BASIC Stamp reads a 1 (above). When the pushbutton is not pressed, the BASIC Stamp reads a 0 (below).

GANTT CHART
APPENDIX L

GANTT CHART **Winter 2006**												
	Task	Week 1 Jan 8-Jan 14	Week 2 Jan 15-Jan 21	Week 3 Jan 22-Jan 28	Week 4 Jan 29-Feb 4	Week 5 Feb 5-Feb 11	Week 6 Feb 12-Feb 18	Week 7 Feb 19-Feb 25	Week 8 Feb 26-Mar 4	Week 9 Mar 5-Mar 11	Week 10 Mar 12-Mar 18	Week 11 Mar 19-Mar 25
1	Test out OEMs											
2	Explore transceiver											
3	Plan out connections											
4	Research press buttons											
5	Prototype 1 pusbutton and Sound											
6	Order press buttons											
7	Get wireless address											
8	Tech Support (problem)											
9	Circuitry for buttons											
10	Pseudocode for buttons											
11	Crimp wires for OEM											
12	Wireless transceiver communicate											
13	Wireless Interface											
14	Update Thesis Report			due 1/26/2006								
15	Smoothning the Sound											
16	One Octave Prototype											
17	Three Octave Switch Prototype											
18	Test one octave Wireless											
19	Analysis Work											
20	Prepare for Presentation											
21	Obtain materials for actual project											
22	Test three octave Wireless											
23	Analysis Presentation											
24	Familiarize MIDI											
25	Coding for MIDI											
26	Convert Sound to MIDI											
27	Multiple Instrument Audio Options											
28	Headphone option											
29	Test Everything Together											
30	Button Arrangements											
31	Case/Enclosure											
32	Start putting everythign together											
33	Work on Documentations											
34	Demonstration of Initial Implementation											
35	Detailed Design Documentation											
36	Test Everything (after put together)										stuff to do prior to Spring 2006 quarte	
37	Debugging										(spring break)	

BUDGET
APPENDIX M

Senior Design Project Budget (2004-2005)				
Conceptual Design Build Components:	**Qty**	**Price/Item ($)**	**Total Cost ($)**	**Budget Balance ($)**
				$400.00
Microprocessor Kit	1	$119.00	$128.82	$271.18
Flexi Force	2	$25.00	$59.63	$211.55
Design Build Components				
Microprocessor Kit	0	$0.00	$0.00	$211.55
Audio Software	1	$0.00	$0.00	$211.55
Flex Sensor	6	$10.00	$72.50	$139.05
Other Components:				
USB to Serial Adapter	1	$29.99	$32.39	$106.66
VELCRO	1	$6.43	$6.96	$99.70
Gloves	2	$25.00	$27.06	$72.64
Kids Gloves	1	$2.99	$3.24	$69.40

Senior Design Proposal Budget for (2005-2006)					
Components:		**Qty**	**Cost Range**	**Expected Cost**	**Note**
Microcontroller Kit		1	$150-$170	$175.00	*Kit includes electrical components*
Wireless Interface			$125-$160	$155.00	
Sensory Interface					*We have 6 flex sensors from last year, we just need 4 more.*
	Flex Sensor	10	$10-15	$55.00	
	Vibration Sensor		$95-$120	$125.00	
Audio Interface					
	Speaker	1	$10-25	$20.00	
	Amplifier	1	$65-$75	$70.00	
Power Management			$15-$25	$20.00	*battery regulation/power supply*
Other Components:					*For the other components, we will use the same components as last year.*
	USB to Serial Adapter	1	$32.39	$0.00	
	Velcro	1	$6.96	$0.00	
	Gloves	2	$27.06	$0.00	
	Kids Gloves	1	$3.24	$0.00	
Total			**$540-$659**	**$620.00**	

PARENT SURVEY ABOUT MUSIC IS POSSIBLE
APPENDIX N

***Music is Possible*:**
An Introduction

Music is Possible was a project first conceived by the Kevin Ip and Luis Vicencio for Santa Clara University's College of Engineering Senior Design Program. Their intention was to design a device or product that could be used by people with motor impairments or limited motor function. They effectively created—in conjunction with The Avalon Academy and with the advice of Dr. Shoba Krishnan of the Electrical Engineering Department—a device capable of producing music through electromechanical sensors and software. This device was inspected and tested by Jeremy and Joshua F., two boys attending The Avalon Academy, who are themselves coping with the symptoms of cerebral palsy. During the two visits made to The Avalon Academy by Luis and Kevin, the device was considered to have met their criteria for functionality and usability, though still limited in comfort and adaptation.

As a continuation for this new school year, our project team, Trang Phan and Joshua Yee, have further improved and optimized the design for the *Music is Possible* device. The intention was to alter the design of the cumbersome and sometimes difficult glove-type interface that was previously used. The device will now be a standalone, enclosed project, akin to a handheld electronic piano keyboard, capable of playing music either by itself or through a computer and wireless communication system.

A Parent's Survey on Expected Child Involvement in Music

*Please circle *one* answer that you think best fits the behavior of your child.

1. My child likes music.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

2. My child will be or is interested in playing an instrument.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

3. My child would need a specially-designed, adaptive instrument such as the *Music is Possible* device if he/she were to play.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

4. My child would appreciate or benefit from the design of the *Music is Possible* device or similar device.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

5. I would purchase or encourage my child to use such a device.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

----

Please write your responses to these questions and any additional comments on the back of this sheet.

6. How can the *Music is Possible* device meet your child's therapeutic needs and how would such a device specifically stimulate your child effectively?

7. What are your child's symptoms that such therapy would address?

SURVEY RESULTS

APPENDIX O

Survey Results

Seven parents of children from The Avalon Academy took the test.

1. My child likes music.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

One hundred percent of the parents strongly agree with the statement:

- *7 out of 7 strongly agree*

2. My child will be or is interested in playing an instrument.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

More than half of the parents strongly agree with the statement:

- *4 out of 7 strongly agree*
- *3 out of 7 agree*

3. My child would need a specially-designed, adaptive instrument such as the *Music is Possible* device if he/she were to play.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

More than half of the parents strongly agree with the statement:

- *5 out of 7 strongly agree*
- *1 out of 7 agrees*
- *1 out of 7 disagree*

*Note: Parents may like a device similar to the *Music is Possible* device, but their children may need some other device that is specially designed for their symptoms. Or the parents may feel that their children can use the *Music is Possible* device already without changing the design.

4. My child would appreciate or benefit from the design of the *Music is Possible* device or similar device.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

More than half of the parents strongly agree with this statement:

- *6 out of 7 strong agree*
- *1 out of 7 agrees*

5. I would purchase or encourage my child to use such a device.

a) Strongly agree b) Agree c) Disagree d) Strongly disagree e) Not sure

More than half of the parents strongly agree with this statement:

- *5 out of 7 <u>strong agree</u>*
- *2 out of 7 <u>agree</u>*

<u>Parents' Quotes</u>

6. How can the *Music is Possible* device meet your child's therapeutic needs and how would such a device specifically stimulate your child effectively?

7. What are your child's symptoms that such therapy would address?

Quotes of parents of children with cerebral palsy:

- "My son would be unable to use a regular instrument because his poor fine motor skills."
- "My child has cerebral palsy with limited use of his hands….The music is possible device can help my child in various ways. It has an immediate cause and effect response. It can teach my child notes (letters) and different sounds corresponding to the notes."
- "My child has Cerebral Palsy so the movement in his finger is limited. A device such as Music is Possible will able[sic] a New World of opportunities for him."
- "Matthew has Cerebral Palsy. He has hypecfonice [sp] and his extremities are really spastic."

Quotes of parents of children of other motor impairments:

- "I am interested in this device and would like to check it out."
- "She has a difficult time using her hands."
- "Having the cause and effect aspect of the device would empower my child. It would make him proud of himself and it would entertain him, which is important to me."
- "Gabbie loves music; she mimics the singing and moves to the beat. She does not have the motor ability for a regular instrument so she would greatly enjoy being able to produce music."
- "It could encourage him to move, focus on something he wants to accomplish and produce music independently.
- "From the information written, The Music Device would be incredible device for Bentley-Grace to be able to use an instrument and or intract [sic] in the playing of music."

STAKEHOLDERS
APPENDIX P

- Avalon Academy
  - *We will be working directly with them for our senior design. They probably care about how we can help their kids.*
- Children with Cerebral Palsy (i.e., Joshua & Jeremy)
  - *This is our target group. They would be our ideal customers. Our project is to help them have some substitute for music since they cannot control their muscles as well. Not only does it serve as music entertainment but also some form of therapy.*
- Adults with Cerebral Palsy
  - *They may care because providing a product for the children with cerebral palsy would means one step closer for them to have a similar product.*
- Their caretaker or whoever the person depends on
  - *Their caretaker may care because they probably want the best for those they are taking care of.*
- Their parents/families/friends
  - *Parents and families would care because they also want what is best for their family member. They would probably like to see those with cerebral palsy be happy and be able to play instruments with others. They would probably want to help their family member to learn to be independent and help alleviate the condition.*
- Physical therapy
  - *They may care to see some different form of physical therapy for those who lost control of muscle movements. The device would be another method for them to use.*
  - *American Academy of Orthopedic Surgeons [http://orthoinfo.aaos.org/fact/thr_report.cfm?Thread_ID=318&topcategory=Children]*
- Neurology Groups:
  - *They would probably support us because they would care about any updates/research on cerebral palsy (where there are damages in the nervous system)*
  - Society for Neuroscience [http://web.sfn.org/Splash.cfm]

- National Institute of Neurological Disorder
  [http://www.ninds.nih.gov/disorders/cerebral_palsy/cerebral_palsy.htm]
- Psychologist/Psychotherapy
  - *This group of people may support the device because it would be a form of music as psychotherapy. It may support their theory of how music would help those with injured/damaged brain be able to "increase ability in emotional empathy, increased lucidity" [The Brain by Howard Pierce]*
- Vibroacoustic Therapy
  - *They would probably support this design because it would support their research in how vibroacoustic therapy (surrounded by speaker and vibration) would help alleviate the symptoms for those with cerebral palsy. [The Brain by Howard Pierce]*
- Hypersense Complex
  - *They would care because they are creating some music interface with flex sensors. [http://createdigitalmusic.com/index.php?option=com_content&task=view&id=770&Itemid=44]*
- Biotechnology Industries
  - *These industries may support this because they might be interested in some way of helping those with cerebral palsy. (i.e.,* Stimtech Inc)
- Cerebral Palsy Institutions
  - *They would care about what can be done to help those with cerebral palsy. They would care about entertainment and therapy for cerebral palsy, etc.*
- Assistive Technology
  - *They would support this device because it deals with helping their targeted group [http://www.pluk.org/AT1.html#2]*
- Professor Krishnan
  - *our advisor for the Sr. Design; will be helping us*
- School of Engineer
  - *Sr. Design distinguishes the School of Engineer apart from the rest of the university*
- Santa Clara University distinction for our school

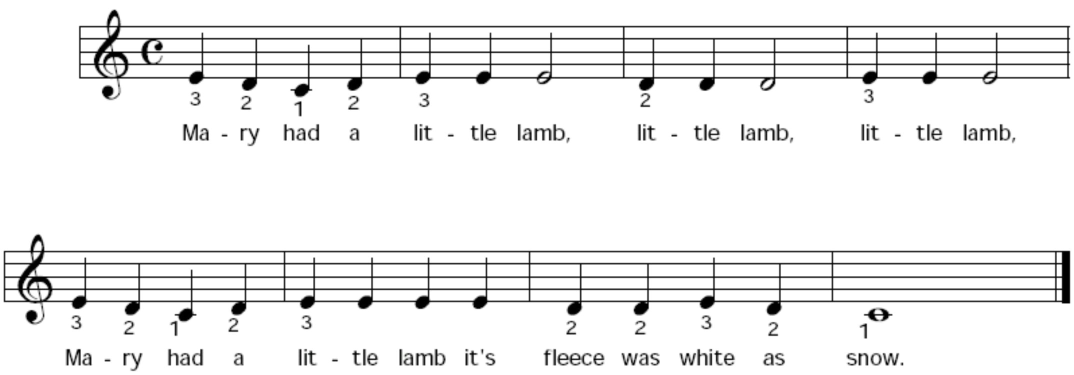
SHEET MUSIC
APPENDIX Q

Color-coded music sheet based on children request. These are color-coded music sheet to make reading the notes easier to read.

Mary Had a Little Lamb

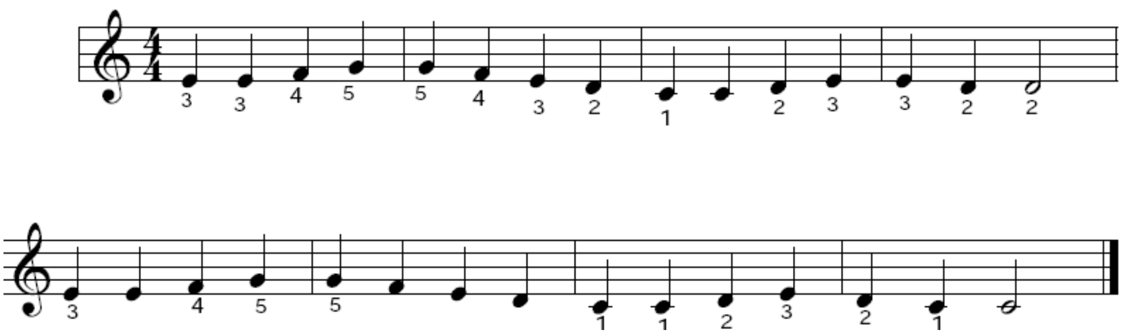
First Pieces

Arr: Gilbert DeBenedetti



Ode to Joy
First Pieces

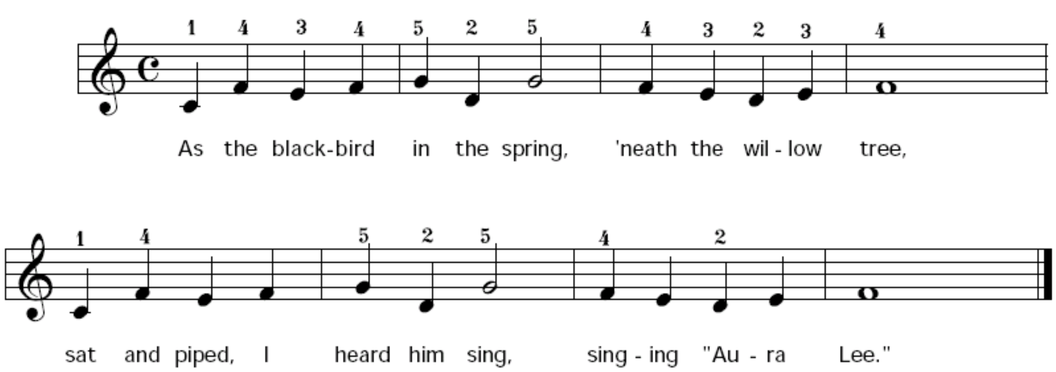
Beethoven
Arr: Gilbert DeBenedetti



Popularized by Elvis Presley as

"Love Me Tender"
First Pieces

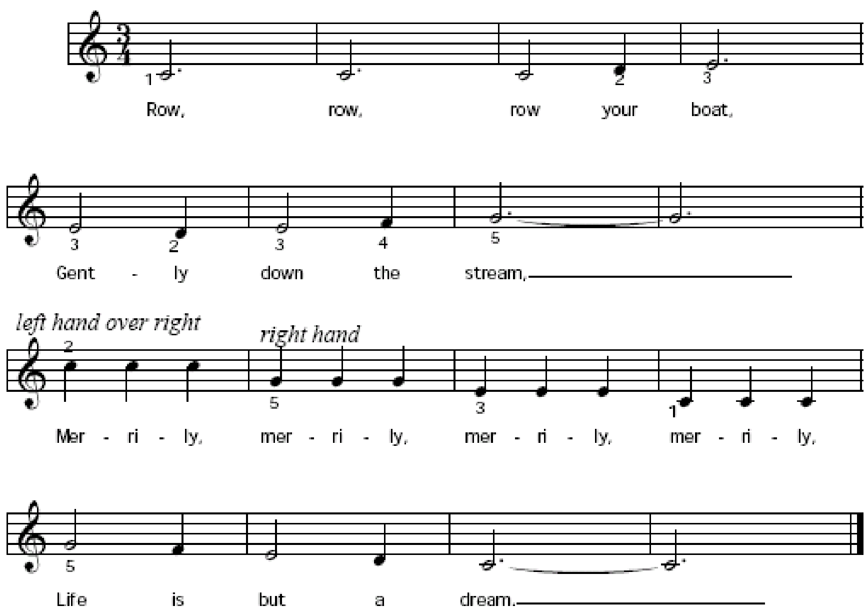
Arr: Gilbert DeBenedetti



Row, Row, Row Your Boat

First Pieces

Arr: Gilbert DeBenedetti



Jingle Bells

Short Version

First Pieces

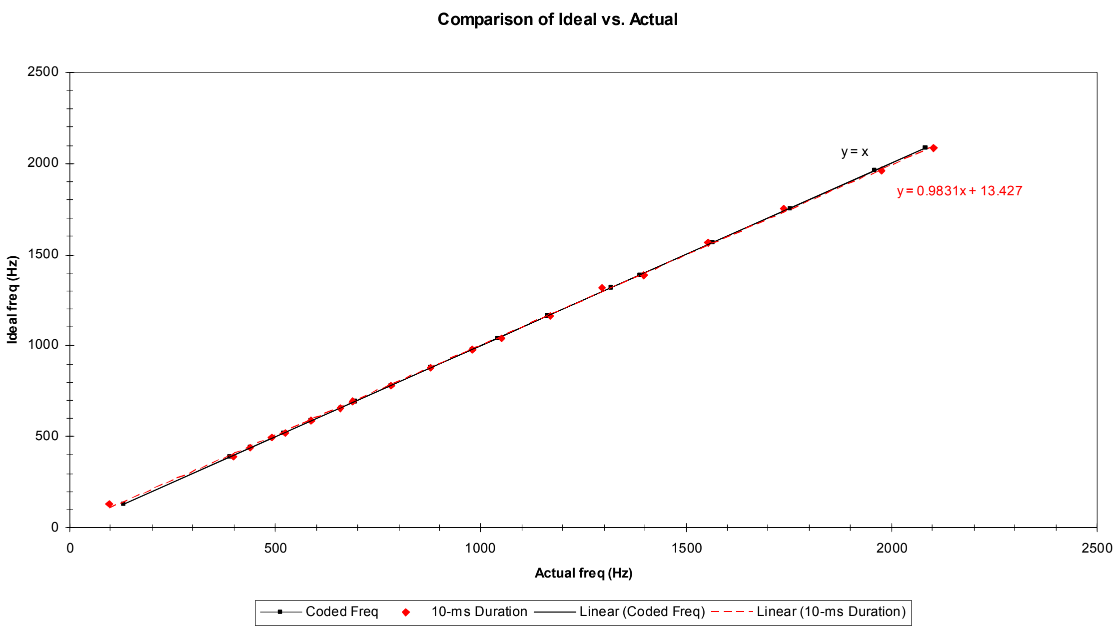
J. Pierpont

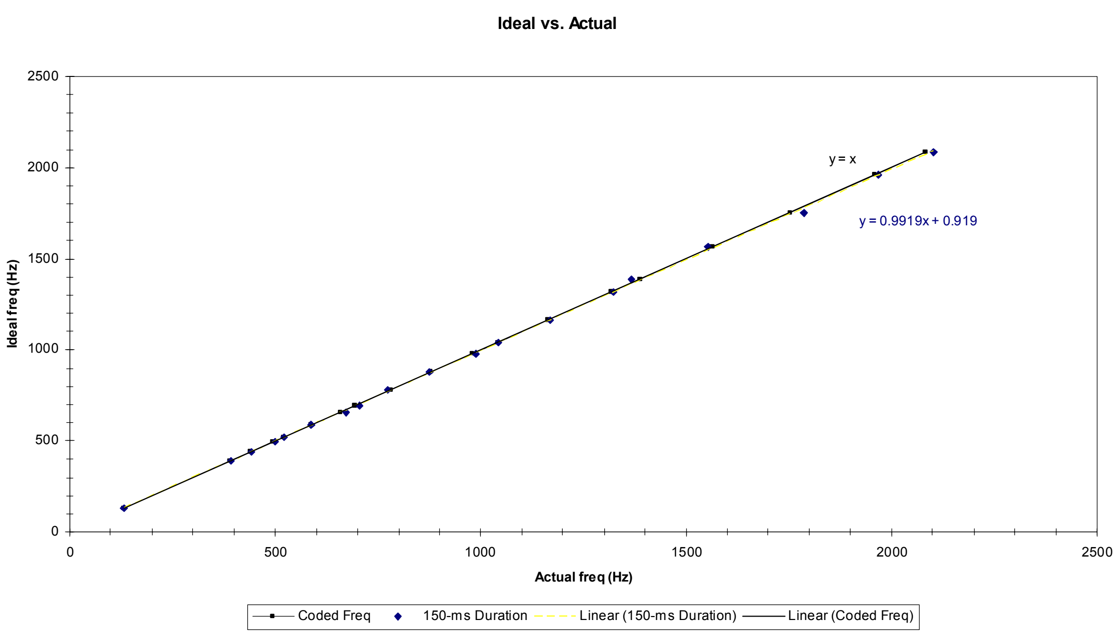
Arr: Gilbert DeBenedetti

DATA ANALYSIS

APPENDIX R

Note	Frequency (Hz) Expected			
	for C	for B		
C (ridiculously low note)	65	124		
C1 (very low note)	131	248		
C2 (low note)	262	495		
C3 (middle C)	521	980		
C4 (high note)	1042	1960		
C5 (very high note)	2084	3920		
C6 (rediculously high note)	4168	7840		
Note	Program-med	Duration 10	Duration 150	Duration 350
	Hz	Hz	Hz	Hz
C3	521	523.56	520.833	526.316
D3	588	588.235	588.235	588.235
E3	658	657.895	671.141	649.351
F3	694	689.655	704.225	704.225
G3	781	781.25	775.194	787.402
A3	877	877.193	874.126	877.193
BE3	980	980.392	988.142	992.063
C4	1042	1050.42	1041.67	1041.67
D4	1163	1168.22	1168.22	1168.22
E4	1316	1295.34	1322.75	1322.75
F4	1389	1396.65	1366.12	1396.65
G4	1563	1552.8	1552.8	1552.8
A4	1754	1736.11	1785.71	1760.56
BE4	1960	1976.28	1968.5	1968.5
C5	2084	2100.84	2100.84	2100.84
C1	131	95.2381	131.579	131.579
G2	391	396.825	393.701	390.625
A2	439	438.596	442.478	434.783
B2	495	490.196	500	495.05


Comparison of Ideal vs. Actual
2500
2000
1500
1000
500
0
Ideal freq (Hz)
0
500
1000
1500
2000
2500
Actual freq (Hz)
y = x
y = 0.9831x + 13.427
Coded Freq
10-ms Duration
Linear (Coded Freq)
Linear (10-ms Duration)


Ideal vs. Actual
2500
2000
1500
1000
500
0
Ideal freq (Hz)
0
500
1000
1500
2000
2500
Actual freq (Hz)
y = x
y = 0.9919x + 0.919
Coded Freq
150-ms Duration
Linear (150-ms Duration)
Linear (Coded Freq)

Ideal vs. Actual