Punching Bag Trainer

A Senior Project

presented to

the Faculty of the Electrical Engineering Department

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

by

Gian Delacruz

December, 2009

Table of Contents

List of Tables and Figures

Abstract

Martial artists; such as boxers, kick boxers, and karate practitioners; use punching bags in training all the time to work on technique, speed and power but there hasn't really been a good way to train reaction and combination speed (a combination is a string of strikes) without the aid of a training partner. Reaction and combination speed are important in competition because a faster competitor can attack successfully before his or her opponent has the time to react. This device aims to help martial artists increase reaction and combination speed through the use of randomly flashing targets attached to a punching bag. The give and take nature of the device (it gives a target and waits for the user to hit it then gives another) allows users of all levels to use the device. Novices can go at a slower pace to work on proper technique and advanced users can go as fast as they can, working on reaction and combination speed.

I. Introduction

In a match; boxing, kickboxing, etc.; a good opponent behaves in an unpredictable way; he/she may jab and open up his/her torso for attack for instance. This device uses randomly lit targets to mimic the randomness of an actual match. One target on the device lights up and the person using the device hits the target and another lights up and so forth. By using this device the user increases reaction and combination time by reacting to the random and nature of the device, he/she learns to see openings and attack them quicker and learns to transition smoother from technique to technique.

I created this device, called a Punching Bag Trainer, using a standing punching bag, and a Nexys Board microcontroller. The punching bag contains the targets and lights and the microcontroller controls the operation of the lights. The targets on the punching bag send information to the microcontroller, telling it whether or not the right the user hit the right target, and the microcontroller controls the lights based on that information, switching lights if the user hits the correct target or keeping the same light if not.

II. Background

In my experience training boxing, kickboxing, and Hapkido, I never came across a device that trained reaction and combination speed without the help of a human partner. Generally, one person wears mitts (the mitts catch the strikes) and holds them up for different strikes, such as a jab or a kick, and the other person hits them. Here, the person hitting the mitts reacts to the person holding the mitts and eventually learns to react faster and transition smoother from strike to strike. The punching bag trainer acts in a similar manner to the person holding the mitts. One target on the punching bag lights up, the user hits that target then another target lights up and the user hits that target and so on. The random changing of the targets simulates the randomness of the mitt holder or of an opponent in a match.

Although this device somewhat simulates a training partner it cannot replace a good training partner. A good training partner critiques the user's technique, offers up suggestions, and moves around the way an opponent would, the punching bag trainer cannot do those things. The person using the punching bag trainer must judge their own technique. With all that being said, the punching bag trainer still works well as a substitute when the user does not have access to training partners.

A microcontroller is a small computer that has a CPU and support functions such as timers or analog inputs and outputs. I chose to use the Nexys Board microcontroller for this project for a few different reasons; because it contains six four pin input/outputs fitting my need for six outputs for the LED’s, six inputs for the buttons, and one output for the buzzer, because I already own one, and because I have previous programming experience with it.

A standing punching bag is a punching bag supported by a plastic base filled with water or sand to stabilize it during use. The choice to use a standing punching bag for this project came about to fulfill the need for portability and convenience. A hanging punching bag requires a chain and a sturdy beam to hang it from, severely limiting the number of places someone can use it. A standing punching bag, on the other hand, is lighter (when the base is empty), more portable, and far more convenient because it works by itself without the need for hanging beams.

III. Requirements

The device will have six targets put into places most likely to be hit in a match. A competitor in a boxing, kickboxing, or karate match aims mainly for the front of the face, left side of the face, right side of the face, the front of the stomach, the left side of the stomach, and the right side of the stomach. When someone turns the device on one of the targets will light up. The user will hit the lit up area at which point a buzzer will go off signifying a successful hit and another area will light up randomly and the process continues until the user turns the device off.

IV. Design

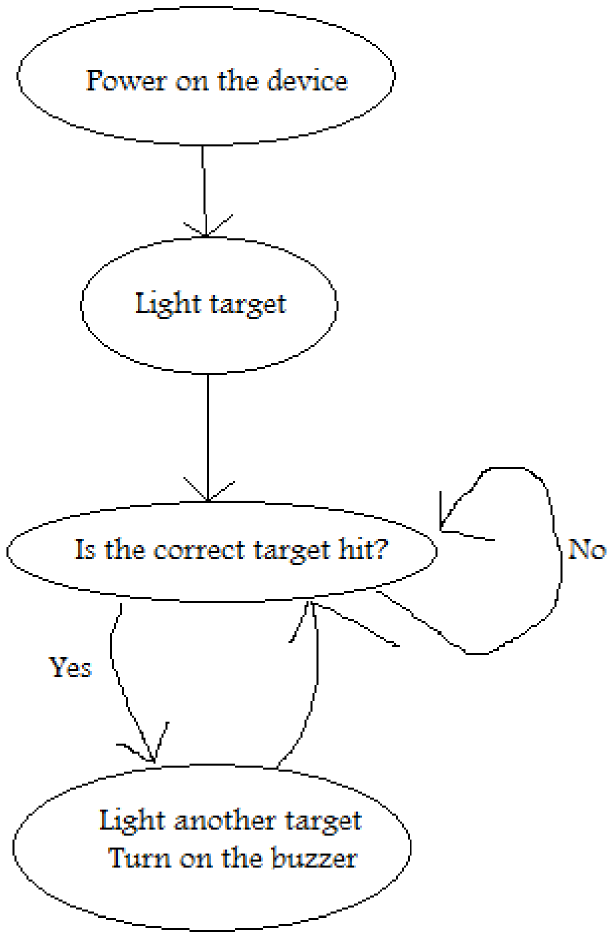


Figure 1: Block diagram of the Punching Bag Trainer

The device uses a Nexys Board to control the function of the LED’s and the buzzer. Buttons attached to the punching bag targets send information to the Nexys Board. If the user hits the correct button (i.e. the button corresponding to the currently lit LED) then the Nexys Board lights another LED at random and turns on the buzzer, continuing the process.

V. Test Plans

To test the code on the Nexys Board I connected the Nexys Board to a breadboard containing six wires and seven LED's. The six wires corresponded to the six buttons on the punching bag, six of the LED's corresponded to the six flashing LED's on the punching bag, and the last LED corresponded to the buzzer. Testing consisted of powering on the Nexys Board, loading the code onto the Nexys Board, pressing the "set" button on the Nexys Board, and seeing if one of the LED's turned on. If one of the LED's lit up I would connect the wire corresponding to the currently lit LED to ground and see if the LED turned off and another turned on and if the LED in place of the buzzer turned on. If this worked, testing continued by grounding wires while looking for any glitches in the operation of the Nexys Board (i.e. multiple LED's turning on, no LED's turning on, non-random patterns, etc.).

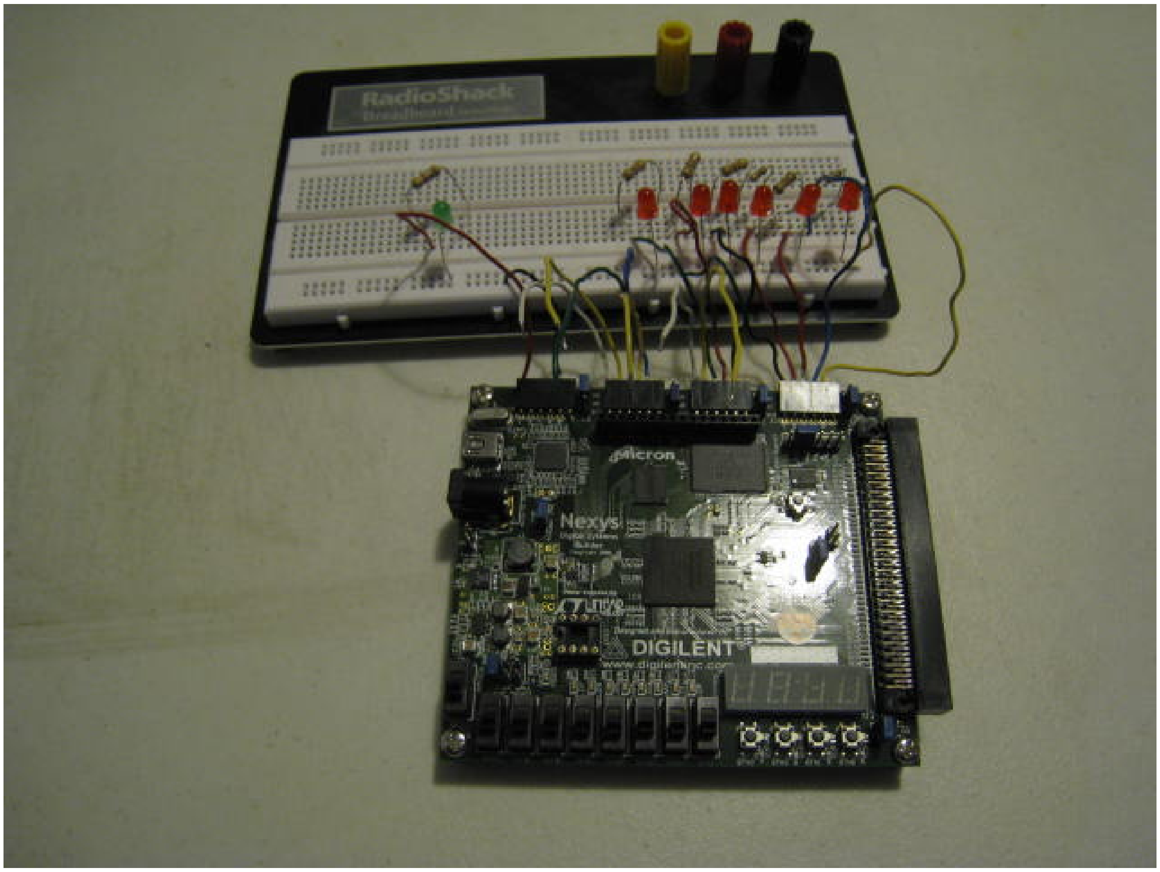


Figure 2: The test setup for the software code

Figure 2 shows the test setup for the software code on the Nexys board. Note the LED's with a 100 ohm resistor to control current. The header pins output a high value voltage of 3.3 volts so the 100 ohm resistor to keeps the current at 20 milliamps, the operating current of the LED's used in the project. The LED's on the right represent the six LED's on the punching bag, the LED on the left represents the buzzer, and the unconnected wires in the middle represent the buttons.

To test the buttons for the punching bag I attached the buttons and the LED’s to the Nexys board. Testing consisted of powering on the Nexys Board to see if the LED's lit up and if pressing the button corresponding to the currently lit LED resulted in the LED's switching. I

made sure to test the durability of the buttons; if the buttons broke or seemed too fragile to hold up to repeated use they would not pass this test.

If both the Nexys Board and the buttons worked then testing continued by integrating the two onto the punching bag and testing both together by turning on the Nexys Board and hitting the buttons on the bag all while looking for glitches in the program (multiple LED's lighting, no LED's lighting, etc.) or problems with the punching bag setup (LED's not changing, the buttons braking, etc.). The main focus of this test was to see if the buttons held up to the type of stress caused by an average punch or kick (assuming that the force of the punches and kicks I perform are average).

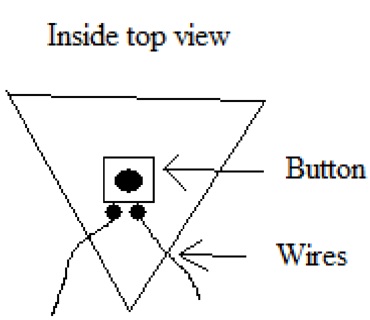
VI. Development and Construction

To develop the system to control the LED's and the buzzer I used a Nexys Board and the Xilinx ISE 9.1i development program. These components allow me to write a program in the ISE 9.1i development program and load that program into the Nexys Board. Writing a program in the ISE 9.1i development program requires learning (or rather, relearning as I had taken a class using this program some two years ago) how to write code in VHDL and integrate that code into the physical components of the Nexys Board.

To begin, I wrote a program to output a high value to one of the header pins, just to get the feel for using the header pins as outputs. Next, I outputted high values to six of the header pins using a string of six variables, preparing them for the output of the LED's. The next thing to do was work on creating an algorithm that would randomly choose one of the LED's to output a high value to. To create pseudo randomness I made another string of six variables and wrote code that would move a high value through the string at each rising edge of the clock provided on the Nexys Board. The first string for the LED's would copy the data of the random string only when the Nexys Board received a button press, creating a seemingly random output on the LED's. The clock provided on the Nexys Board functions at 25 MHz, this frequency ensures pseudo randomness each time the Nexys Board receives a button press and outputs data to the LED's. Next came the creation of code to ensure that the LED's only changed when the Nexys Board received a correct button press. To create button inputs, six of the header pins acted as inputs and an "if statement" allowed the LED's to change only when the input corresponding to the currently lit LED received a high value. This initial button input code provided strange results; powering on the Nexys Board and loading the code showed the LED's constantly

switching even when no button input values were given. Checking the voltage on the header pins revealed that setting the header pins as inputs gave them a voltage of 0.3 to 0.4 volts, enough to give a high value to the input variables. Changing the button code so that the LED's changed only when the input corresponding to the currently lit LED received a low value fixed this problem. The last thing to do was work on writing code for the buzzer. To write code for the buzzer I used another variable to output to one of the header pins for the buzzer and wrote code for a counter. The code of the counter would output a high value to the buzzer variable only if a correct LED change occurred and would start counting at each rising edge of the clock provided on the Nexys Board. When the counter reached a value of 12.5 million it would output a low value to the buzzer variable and reset. The clock frequency of 25 MHz and the limit on the counter ensured that the buzzer would receive a high value for only half a second after a successful button press. Please see Appendix E for the full software code.

Developing the setup for the buttons attached to the punching bag was much more difficult as I am not a mechanical engineer and therefore not familiar with designing and developing mechanical things. I went through several design iterations before finding a suitable design for the button setup.



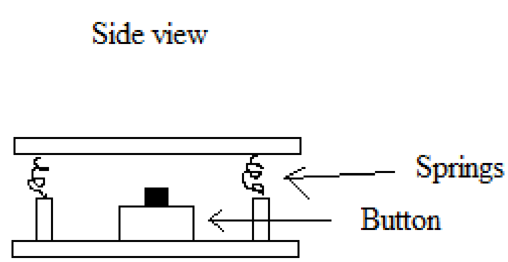


Figure 3: First button design

The first design came about after examining a button mechanism used in a stuffed toy. The button mechanism in the toy sits in a plastic case and switches when the case gets pushed. The design seen in Figure 2 is an attempt at something similar to the toy design but it got scrapped soon after I realized that creating this design would prove much too difficult. Creating this design required access to and understanding of plastic forming machinery to create the custom parts.

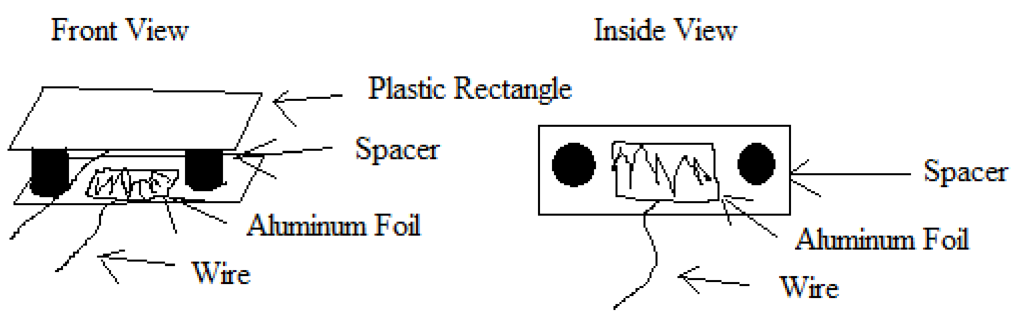


Figure 4: Second button design

After the first design failed I tried to create a simpler button design, thinking that a smaller number of parts meant a smaller number of things that could fail. As seen above in Figure 4, the second design used two rectangular pieces of plastic separated by spacers on each end. I glued wires to the inside top and bottom of the plastic and covered them with foil to make a better contact. The button activated when pressed in the middle creating an electrical connection between the two wires. This design worked well in theory but under testing proved much too fragile. The connections made with superglue on both the spacers and the foil would come apart under repeated use.

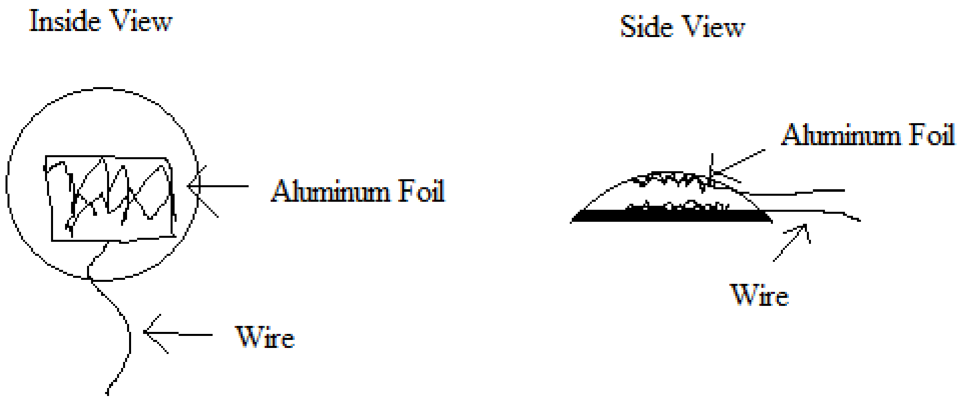


Figure 5: Third button design

The third design came about after speaking with a friend about the project. He remembered the board game *Trouble* in which players would press a plastic dome containing a die to play. We figured that we could apply that concept to my project. He happened to have some plastic googly eyes lying around for an art project he was doing so we came up with a design using those. This design required opening up the plastic eye and super gluing wires to the top and bottom of it covering the wires with foil to make a better contact as seen above in Figure 5. Activating the button meant pushing down the dome of the eye creating a connection between the two wires. Again the design was too fragile; the connections made between the wires and foil would come apart after repeated use.

In the end, I settled on a rather simple design of a button soldered to two wires and sewn into a pocket containing padding. The LED is sewed onto the outside of the pocket and Velcro attaches the pocket to the outside of the punching bag.

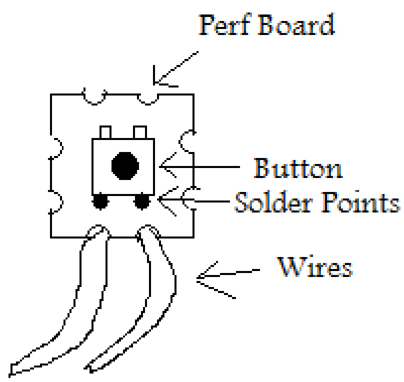


Figure 6: The final button design

Figure 6 shows the final button design. The store bought surface tact switch sticks to the cut out square of perforated board with super glue and solder connects the wires with the connections of the surface tact switch. This design held up to repeated use so I went forward and designed a padding system for it.

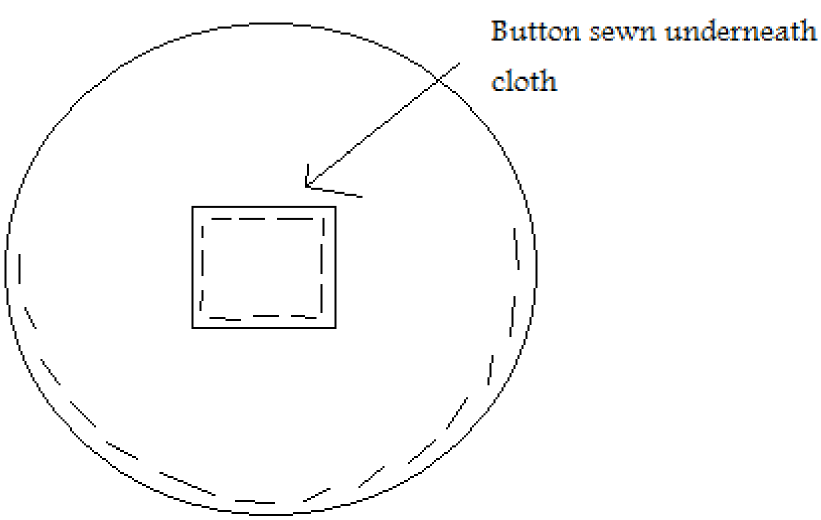


Figure 7: The half finished pocket containing the button

A pocket filled with padding holds the button. Figure 7 shows the pocket with the button sewn in. This design required two circles of cloth approximately eight inches in diameter, one

with a small hole in the middle of it. The button stays stable under a piece of cloth sewed onto the pocket and the wires of the button feed through the hole. I then sewed the two pieces of cloth halfway together, flipped the pocket inside out, and filled the pocket with padding then sewed the pocket shut. The padding used for the pocket was the same padding used to stuff dolls seeing as how toys made for children are usually very durable. The amount of padding used to was enough to fill the pocket and protect the button.

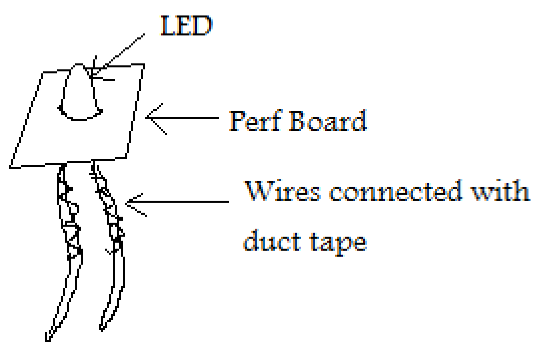


Figure 8: The LED design

Figure 8 shows the LED design. The LED connects to a cut out square of perforated board and duct tape connects the LED to two wires. The duct tape creates a tight and strong connection from the wires to the LED.

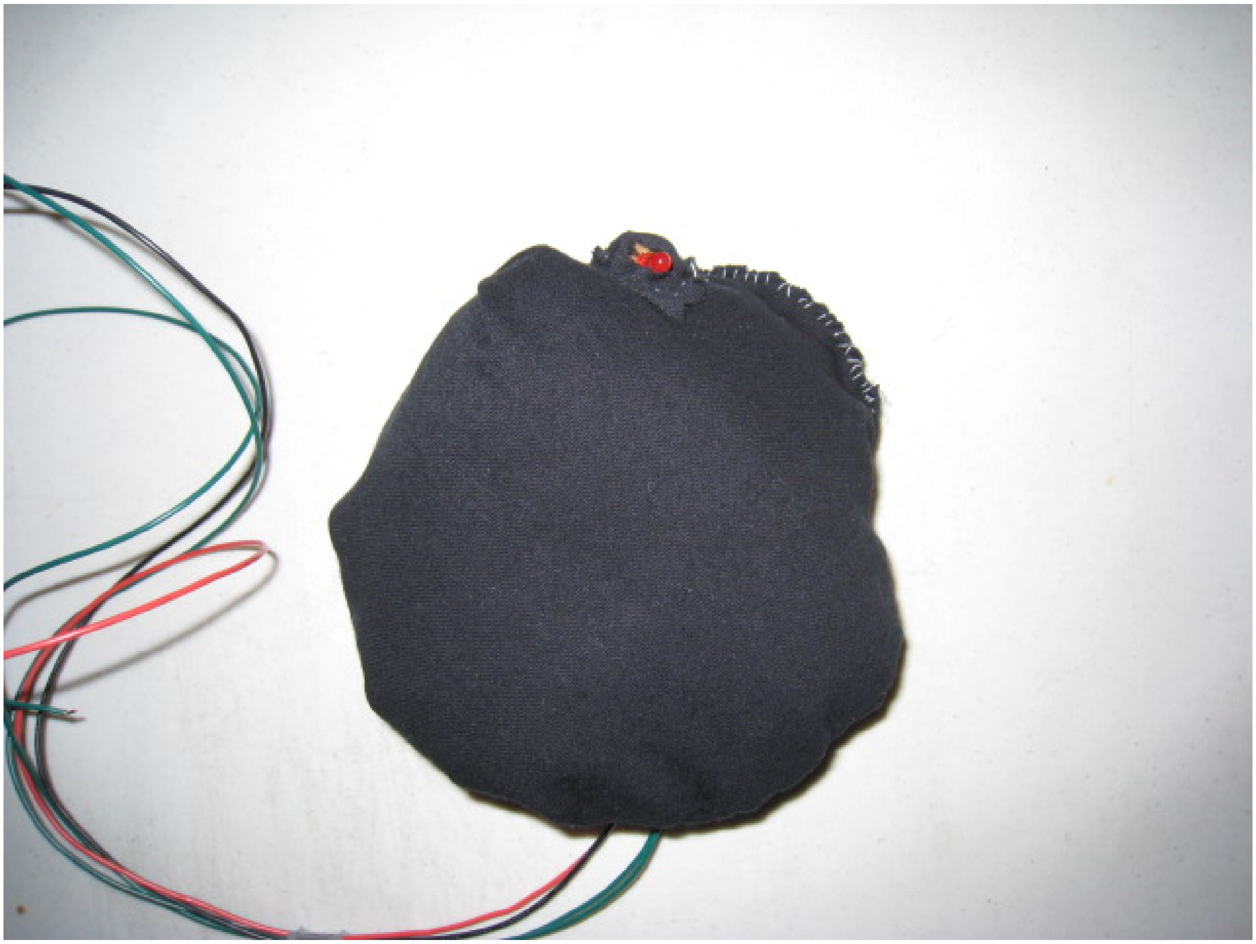


Figure 9: The pocket with the LED

Figure 9 shows the finished pocket with the LED sewn on. A small piece of cloth attaches the LED to the pocket by way of sewing. A small hole in the piece of cloth allows the LED to poke through and shine.

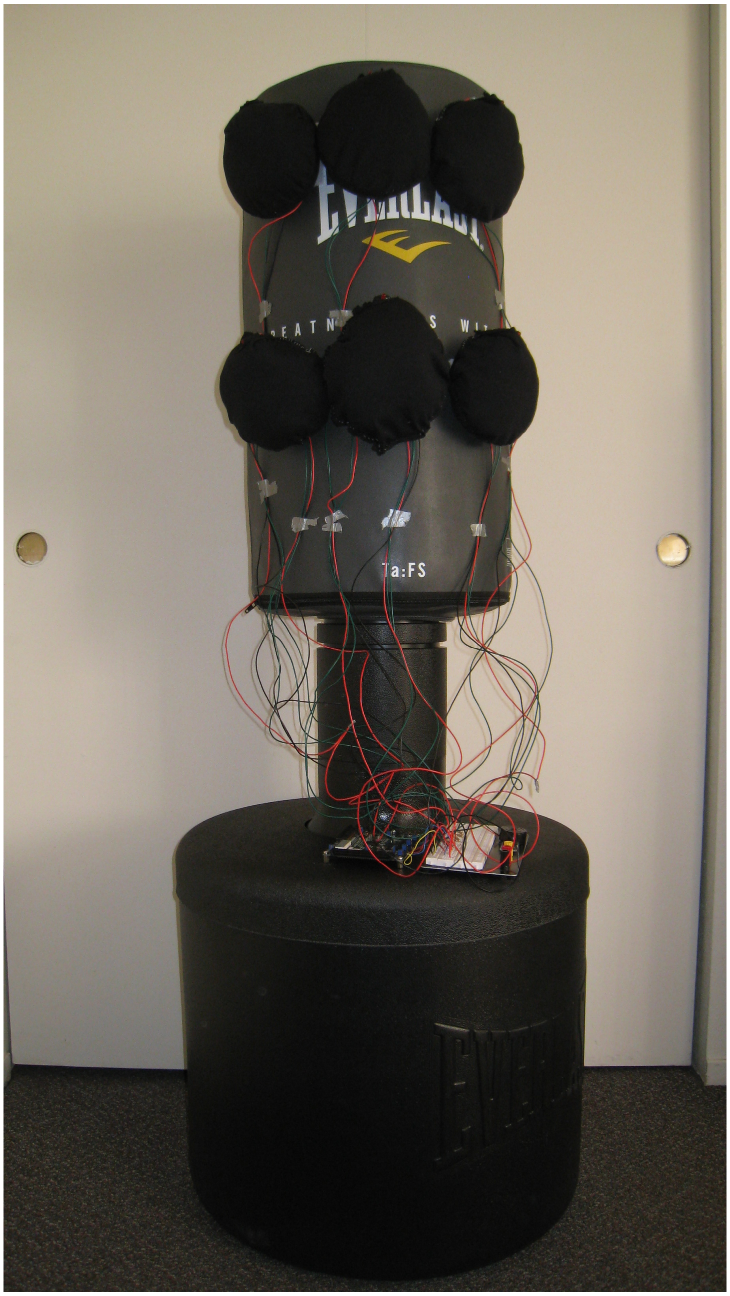


Figure 10: The final integrated product

Fig 10 shows the finished product with the pockets attached to the punching bag and the microcontroller. Velcro held the pockets to the punching bag with and duct tape held the wires to

the punching bag with to keep them from getting in the way during use. Duct tape secured the Nexys board and breadboard to the base of the punching bag.

VII. Integration and Test Results

Program Test Results:

Testing the code brought up no glitches or errors. The code performed just as expected; one LED lit up and randomly switched to another LED and the buzzer LED lit up for half a second only when the Nexys Board received a correct button press.

Button Design Test Results:

Although the designs I created all passed the test of functionality in that they all functioned as buttons creating an electrical connection when depressed, two of the designs did not pass the initial test of durability. The second and third designs proved much too fragile for use under the stress of this application. Specifically, the buttons would come apart at the connections made with super glue. This problem stumped me at first; it is called super glue for a reason; but consulting my advisor led to an explanation. The failure comes not from the super glue but from the materials that it connects. In both cases the stress from repeated use was enough to degrade the plastic and foil to the point of failure. The fourth and last design, however, did pass the initial test of durability, it held up to repeated testing and use so it went on into the final integration.

Final Integration Test Results:

The final integrated product functioned correctly; hitting the target corresponding to the currently lit LED resulted in a random change of the LED and a buzzer signal. One thing to note; a couple of times during testing I would find a button that would not work. Cutting open the pockets and taking out the button revealed that the wires would come off of the button at the solder points, losing their electrical connection. Re-soldering the buttons making sure to solder a

sturdier connection remedied this problem. Although re-soldering seemed to fix the problem as the re-soldered button would continue to work and hold up to the force of a punch or kick, the initial failings still leave questions about the final products long term durability. After consulting my advisor, I found that another way to solve the durability problem was to encase the button in glue or epoxy to stabilize the button, specifically the solder points, so that there wouldn't be so much tension in the wires, keeping them from breaking down.

VIII. Conclusion

The project went well and the finished punching bag trainer works and meets the requirements I set. The code written and implemented on the Nexys Board microcontroller switched the LED's and switched on the buzzer for half a second when the correct button was pressed and the button design continued to function under repeated testing and use.

Adding a timer and hit counter to the software code could improve the project. The timer would set the microcontroller to function for only a certain period of time, say two minutes. The hit counter would count the number of successful hits in that time period and display them on the Nexys Board after the timer period. The hit counter gives the user a quantitative way of measuring his or her progress with the device.

If I were to do the whole project over again I would first choose a more electrical engineering project than this one. Half of the work this project required was electrical engineering focused, the programming and interfacing with the Nexys Board, the other half, the design and construction of the button setup, was mechanical engineering focused. Working on the button setup was very frustrating as I felt directionless without any prior mechanical engineering experience to draw from. Much of the design iterations for the button setup came about from trial and error. With all that being said, this was a good learning experience in that I had to try something new, do research on my own, and think about things in a way I'm not used to.

Doing this project taught me a lot about programming on the Nexys Board, interfacing the Nexys Board with outside devices, and creating durable physical devices. More importantly,

I gained valuable experience with the whole design, construct, and test process. My experiences doing this project will definitely serve me well in the engineering industry.

IX. Bibliography

"Microcontroller." Wikipedia. 20 April 2009. <http://en.wikipedia.org/wiki/Microcontroller>

Karate Depot. 30 September 2009. <http://www.karatedepot.com/tr-bg-20.html>

"Global Warming." Wikipedia. 25 November 2009. <http://en.wikipedia.org/wiki/Global_warming>

"Polystyrene" Wikipedia. 25 November 2009. <http://en.wikipedia.org/wiki/Polystyrene>

Appendices

A. Project Specifications

-Adjustable height, 52” to 70” tall

-Six circular target areas with an LED above each target

-Enough padding in the targets to ensure operation for at least two weeks at half an hour of use a day

-Microcontroller and circuit board connected to the base of the punching bag with duct tape

-A switch on the microcontroller powers the device on and off

B. Estimate of Parts and Costs

Table 1: Estimate of Parts and Costs

Part	Cost (dollars)
Microcontroller	100
Breadboard	15
Punching bag	140
Cloth	1.5
Padding	8
Various Electronics (LED's, wires, buttons etc)	30
Velcro	8
Total Cost	302.5

C. Timeline

Table 2: Gantt chart for spring quarter 2009

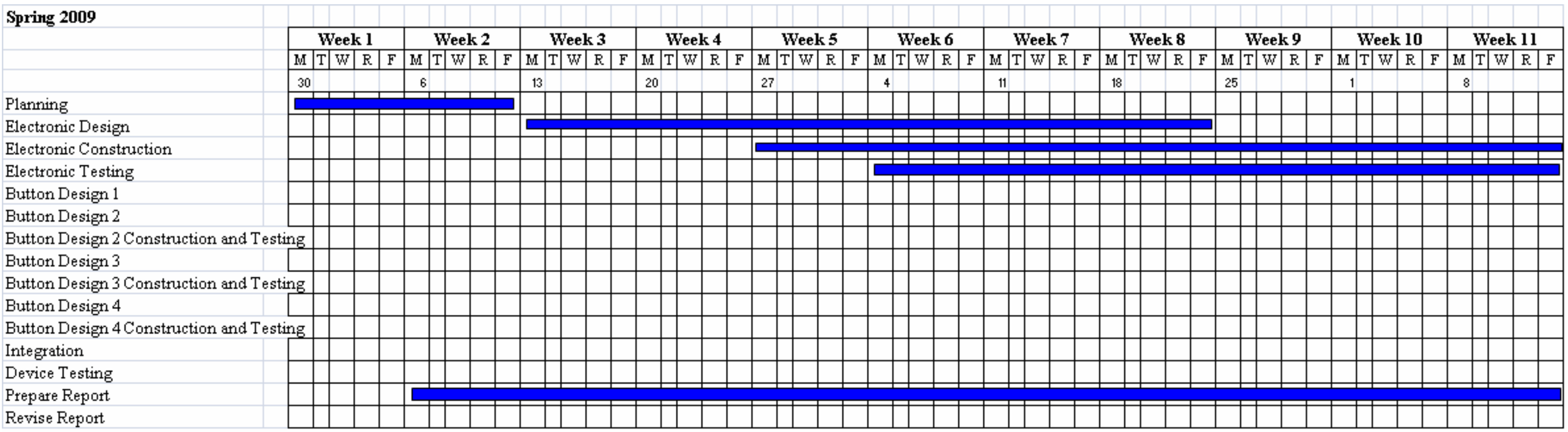
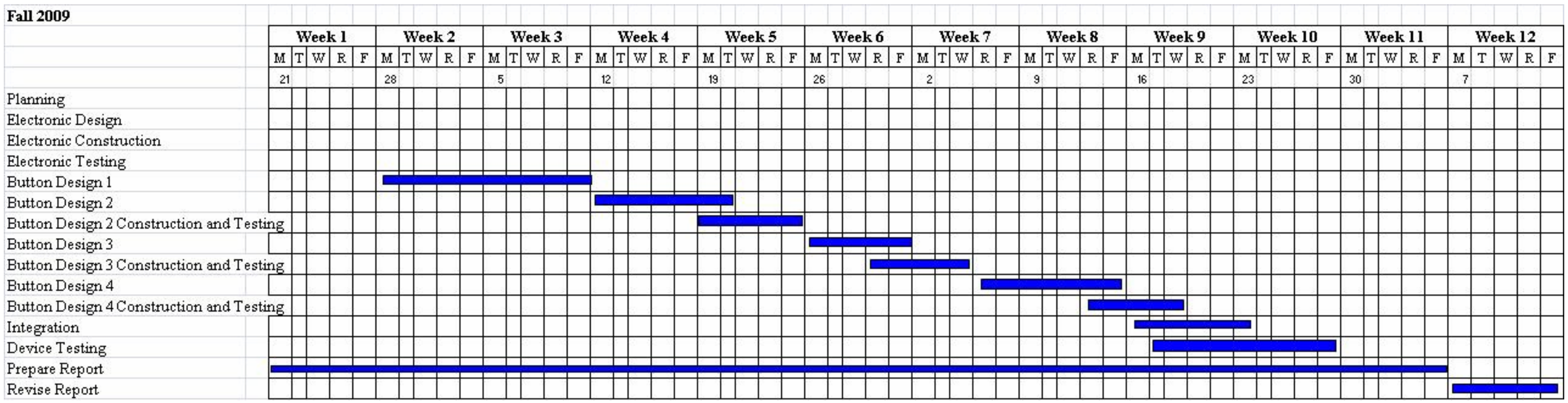


Table 3: Gantt chart for fall quarter 2009



D. ABET

Economic

This device does not replace any other training tools, it is something used in addition to them, so it would not put any products out of business but it may take some of their market share. A company that produces punching bags and training tools could mass produce this product cheaply. Standing punching bags sell for around $140 so something like this could probably get away with being sold for $200. The manufacturer could use a cheaper microcontroller in the mass produced version, I used the Nexys Board mainly because I already own one and was somewhat familiar with it but there are many cheaper microcontrollers on the market, some microcontrollers sell for less than ten dollars. Using one of these cheap microcontrollers on the Punching Bag Trainer would significantly decrease costs. Also, mass production allows companies to create products cheaper than if they are not mass produced.

Environmental

Gasoline used in the manufacturing and delivery of this product releases greenhouse gases speeding up global climate change. Climate change affects the environment in negative ways; melting the icecaps and flooding the land, causing extreme temperature shifts, leading some species to extinction, in essence, destroying the environment and its delicate ecosystem. The packaging may also have a negative effect on the environment if the manufacturers use polystyrene foam to package the device. Polystyrene foam does not biodegrade so any polystyrene that gets thrown out stays in the environment as pollution.

Sustainability

This device contains a lot of plastic, and creating plastics requires gasoline. Gasoline is a non-renewable energy resource so this device does not promote sustainability. The manufacturers of this device could promote sustainability by using recycled plastic to produce it, avoiding the waste spent in creating plastics. Also, using renewable energy sources, like solar energy or wind power, in the production of this device would cut down on wastes. One point to make is that someone who purchases this product needs to only do so once; I designed this device to last, so the product does cause much waste in the sense that people wouldn't buy it, throw it away, and buy another.

Manufacturability

A device like this only contains simple circuitry and mechanical parts. A company already involved in manufacturing punching bags could manufacture this product easily. The circuit manufacturing could be outsourced to an electronics company, they would program the microcontroller and prepare it for production, and the device could be put together in a factory that produces punching bags by adding a few extra steps to the process, placing the targets and such on the punching bag and making the connections to the microcontroller. Many of the choices I made when choosing the materials to use with this device came about from a convenience standpoint not a cost standpoint. As I said above in the Economic section, manufacturers could cut the cost of making the Punching Bag Trainer by using a cheaper microcontroller, cheaper products to create the buttons, and by buying the items in bulk.

Ethical

The ethical benefits of using this device include teaching police how to better fight crime, teaching normal people how to defend themselves from attackers, increasing the lifespan through

better health, and teaching the general populous how to fight in case of a revolution against the government. One ethical question some might ask is, “will this device be used to do evil?” Some may argue that would be evil-doers will obtain this device and learn how to fight so they could attack people but this is simply not the case. This device can't teach you how to fight on its own; you cannot learn how to properly attack another human being by only hitting a punching bag. This device is targeted at people who train martial arts as a supplement to their training. The users of this device are already training; they would train with or without this device. If a user of this device decides to go out and beat someone up or some other unethical activity that person would do that regardless of whether or not they owned this device.

Health and Safety

This device acts as a training tool for martial arts but it could also act as a workout tool. Used as a workout tool, this device increases the user’s stamina and overall health. The widespread use of a device such as this could abate the obesity problem America is currently facing, increasing the overall health of the nation and, as a consequence, the safety as well since healthy people are more able to resist ailments and injuries. As a martial arts training tool, this device offers up a double edged sword for health and safety. On one hand, this device could help people increase their self defense skills, keeping them safe from attackers. On the other hand, this device could help people increase their attack skills, creating more dangerous situations.

Social and Political

If the manufacturing of this device is outsourced to some developing nation, as many of the products sold in the U.S. are, we must consider the social and political implications of that. Outsourcing would give jobs and opportunities to people in those developing nations but we

must make sure that they are not being exploited through relaxed or nonexistent job and environmental laws in that nation. We must also consider the impact outsourcing would have on citizens in the U.S. This would take away jobs from workers in the U.S. but outsourced products are usually produced cheaper and therefore sold cheaper than products made in the states so this may lead to the product reaching a wider audience through lower price.

E. Software Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity blinker is port(
	clk : in STD_LOGIC;
	set : in STD_LOGIC;
	btn : in bit_VECTOR (5 downto 0);
	--btn0 : in std_logic;
	--btn1 : in std_logic;
	--btn2 : in std_logic;
	buzz : inout STD_LOGIC := '0';
	outbuzz : out std_logic := '0';
	outled : inout std_logic_vector (5 downto 0) := "000001";
	led : inout std_logic_vector (5 downto 0) :="000001");	-- 6 led's

end blinker;

architecture Behavioral of blinker is

	signal roll : std_logic_vector (5 downto 0) :="000001";
	signal buzzcount : integer range 0 to 12500000 := 12500000;

begin

--shift register, roll, every clk pulse, button inputs, led outputs, buzzer output
shifter: process (clk, set, led, roll, btn)

begin

	if falling_edge (clk) then

--set button starts the system and sets a random led high
		if set = '1' then
			led <= roll;
			outled <= roll;
			buzzcount <= 0;
--button inputs for corresponding leds
		elsif ((btn(0) = '0') and (outled = "000001")) then
			led <= roll;
			outled <= roll;
```

```
                    buzzcount <= 0;
            elsif ((btn(1) = '0') and (outled = "000010")) then
                    led <= roll;
                    outled <= roll;
                    buzzcount <= 0;
            elsif ((btn(2) = '0') and (outled = "000100")) then
                    led <= roll;
                    outled <= roll;
                    buzzcount <= 0;
            elsif ((btn(3) = '0') and (outled = "001000")) then
                    led <= roll;
                    outled <= roll;
                    buzzcount <= 0;
            elsif ((btn(4) = '0') and (outled = "010000")) then
                    led <= roll;
                    outled <= roll;
                    buzzcount <= 0;
            elsif ((btn(5) = '0') and (outled = "100000")) then
                    led <= roll;
                    outled <= roll;
                    buzzcount <= 0;
            end if;

--buzzer buzzes for half a second after every successful button press
            if (buzzcount = 12500000) then
                    buzz <= '0';
                    outbuzz <= '0';
            else
                    buzzcount <= buzzcount + 1;
                    buzz <= '1';
                    outbuzz <= '1';
            end if;

    end if;

    if rising_edge (clk) then
            case roll is

                    when "000001" => roll <= "000010";
                    when "000010" => roll <= "000100";
                    when "000100" => roll <= "001000";
                    when "001000" => roll <= "010000";
                    when "010000" => roll <= "100000";
                    when "100000" => roll <= "000001";
```

```
                when others => null;
            end case;
        end if;
    end process shifter;

    end Behavioral;
```