

Automated Basketball Return System

by

Victor Mendoza

Chris Chamberlain

Senior Project

Electrical Engineering

California Polytechnic State University

San Luis Obispo

Fall 2009

TABLE OF CONTENTS

Section		Page
	List of Figures	i
	Acknowledgments	ii
	Abstract	iii
I	Introduction	1
II	Background	2
III	Requirements	3
IV	Design	3
V	Test Plans	9
VI	Development and Construction	10
VII	Integration and Test Results	10
VIII	Conclusion	14
IX	References	15
Appendices		
A	Part List and Cost	16
B	Hardware and Schematics	17
C	Source Code	19
D	Senior Project Analysis	24

LIST OF FIGURES

Figure Title	Page
SHARP GP2Y0A700K distance module	4
Distance Sensor Object Detection Method	5
Parallax PIR Motion Detector	5
Method of Detection by Pyroelectric Sensor	6
Fresnel Lens Condensing Light to a Focal Point	7
Scattering Detection Areas by the Fresnel Lens	7
Software Flowchart of Overall System	8
Excel Plots of IR Distance Sensor	11
External Resistors and Capacitors Controlling T_x and T_1	12

ACKNOWLEDGMENTS

We would like to thank Dr. Harris for his patience and motivation in the completion of our senior project.

ABSTRACT

The project is meant to assist a basketball player in their practicing. The project hangs directly on the basketball rim and automatically rotates to find the player on the court. Once the player is detected, the automated basketball return system (ABRS) stays focused in the direction of the player so that any baskets successfully made will be funneled back to the player. When the player moves from center, IR motion detectors find which direction you moved to and rotates accordingly, until the player is focused in front of the funnel again. If at any time the player is absent from the court or cannot be located, the rotation continues in a 180 degree sweeping motion until the player is picked up again.

I. INTRODUCTION

Basketball is a sport that allows a person's individual talents to help support a team as a whole. One of the difficult aspects of playing basketball is learning to shoot with good form while making consistent baskets from every location on the basketball court. Although basketball is a team sport, practicing alone is typically the easiest method to perfect an individual's shooting ability. The problem with practicing alone is the need to collect rebounds from both missed and made shots. A person wastes time and energy chasing after the ball instead of preparing for their next shot.

For our senior project, we wanted to eliminate the need to collect stray basketballs in order to allow for more repetitious shooting and game-pace movement while practicing. We combined an infrared (IR) sensor module, two passive infrared (PIR) sensors, and a Digilent Nexys board microcontroller to track and locate a single person on a basketball court. The overall goal is to funnel a basketball back to the person after shooting a ball and moving to different locations. Due to mechanical and financial restraints, we aimed our project towards "proof of concept" rather than producing a marketable product. We intended to build a prototype that can be used for future designs to make our idea possible.

The following report will explain the process that we took in order to complete our project. In the background section, we present how we chose our project and describe some benefits. The requirements section will list our minimum requirements for the overall project. Then, the design section will explain how the individual components work and how we combined them. The test plan section will describe how we will test our project and the results are described in the integration and test results section. The development and construction

section described how we connected the entire project together. All of our concluding thoughts are in the conclusion section of our report.

II. BACKGROUND

Basketball return machines already exist and are used by high school, college, and professional basketball teams. The machines used are very expensive and are not made for the general consumer. Also, the machines only work for designated locations on the court and do not follow the person shooting.

Our Basketball Return System will be low-cost and add the tacking feature. The system consists of two main modules, the basketball funnel, a dc motor, and the Nexys development board. The first stage of the system is the dc motor. At power-up, the motor will begin to move until it is stopped by a signal from the second stage, the IR object detector. As long as the person remains in the IR object detector's line of sight, the motor will not move. Once the person is no longer detected by the IR detector, the second stage, two PIR motion sensors, turns the motor in the direction of the detected movement. The motion sensors are placed on either side of the IR detector. The motor will turn either clockwise or counter-clockwise depending on which side detects movement. The motor will continue to move until the IR detector locates the person once again and stops the motor. The person shooting the basketball will be able to shoot and move to a new location without chasing rebounds. The entire system is programmed in C and downloaded onto the Nexys board via Export. All the outputs of the three sensors are inputs to the Nexys board which then outputs the appropriate response to the motor.

III. REQUIREMENTS

Power

The ideal system will be a stand-alone system that runs solely on rechargeable batteries for easy use indoors as well as outdoors. For demonstration purposes, the system will run on external power of 120V AC. The internal modules must run on 5V DC.

Motor

The motor used must have enough torque to turn the funnel while using a small amount of DC Power and low voltage.

IR Object Detector

The IR object detector must detect an object at least a distance of 19'9"; the distance to the three point line on the basketball court.

Motion Detector

The motion detectors must detect motion at a distance of at least 19'9" and provide a quick response for multiple detections under short periods of time.

IV. DESIGN

We wanted a system that stayed above the ground and yet easy to install. Quikshot USA provided us with the funnel that allowed for a 360 degree rotation. It hangs from the basketball rim and returns the ball through the funnel attached. The ball does not return with great speed but the funnel does get the job done. Although the funnel did rotate manually, we needed a way to turn the funnel 180 degree only by an attached motor. The motor we chose is a Ryobi 4V hand-drill that has enough torque to turn our system. The drill's motor is small and inexpensive which makes it an ideal choice for our project. Initially, the motor will begin to move and search for you. The person must put themselves in the funnel's line of sight to begin. If the person is not detected in front of the funnel, then the motor will continue to rotate back and forth along the

180 degree angle until the person is detected. In order to keep the motor from turning the unnecessary 180 degrees behind the rim, two touch sensors are added. The touch sensors will be simple switches that will stop the motor from turning in one direction and allow the motor to spin in the opposite direction.

In order to detect that the person is in the funnel's line of sight, we chose to use the SHARP GP2Y0A700K distance sensor shown below in Figure 1.



Figure 1: SHARP GP2Y0A700K distance module

The module is low-cost and very simple to use. This distance module provides an easy object detection method mainly used in robotics projects, but will work great for our idea. It requires 5V DC to operate and its output is an analog voltage that is used to detect the distance of an object from the sensor. Depending on the distance of the detected object, the module outputs a voltage which can then be converted to distance. For our project, we were only concerned with the detection of a person, not so much how far away the person is from the basket. The module has a narrow detection width and detects up to 21 ft, both of which are necessary in order to stop the motor at the exact moment the person is detected. The module sends out an IR beam and waits for a reflected beam to return. If there is no object, the light is never reflected back. Figure 2 displays the method in which the module detects an object's distance using angles within a triangle. When light bounces back from an object, the light enters an enclosed charge-coupled device (CCD) array at a triangulated angle which allows for the distance calculation. We only used it for detecting an object in its line of sight.

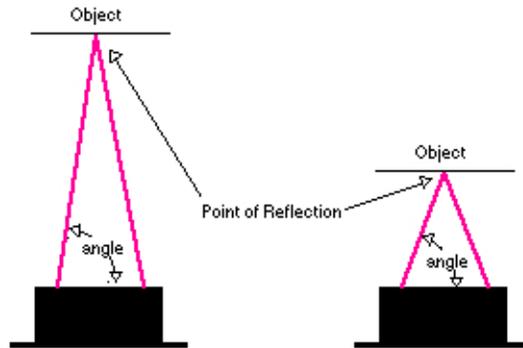


Figure 2: Distance Sensor Object Detection Method

After the person moves from the IR sensor’s line of sight, our system will detect the movement and turn the funnel in that direction. The motor will continue to move until the IR sensor detects the person once again. We chose to use the Parallax PIR (passive IR) motion detector module to detect motion because it is also is low-cost and very easy to use.

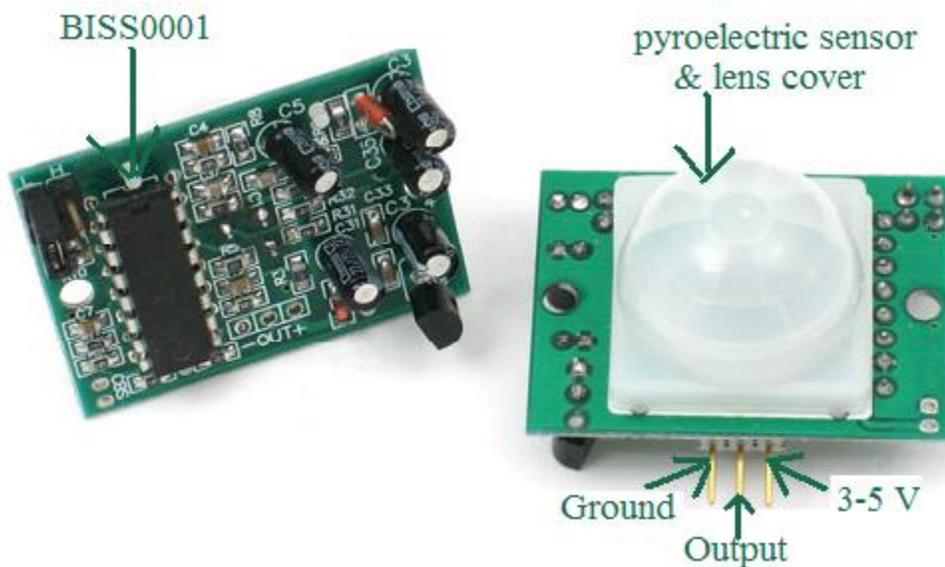


Figure 3: Parallax PIR Motion Detector

Figure 3 above displays the PIR motion detector. The module consists of several supporting resistors and capacitors, a BISS0001 chip, and a pyroelectric sensor. The pyroelectric sensor

detects levels of infrared radiation from its surroundings. The sensor is split into two halves in order to use it as a motion detector. The output of the sensor is connected to the BISS0001 chip. The chip takes in both the outputs from both halves and does a little bit of logic. When one half of the sensor “sees” more or less radiation than the other, it causes a positive differential change between the two halves and the output swings high. When the other half “sees” the change, the opposite happens and the output swings low as seen in Figure 4.

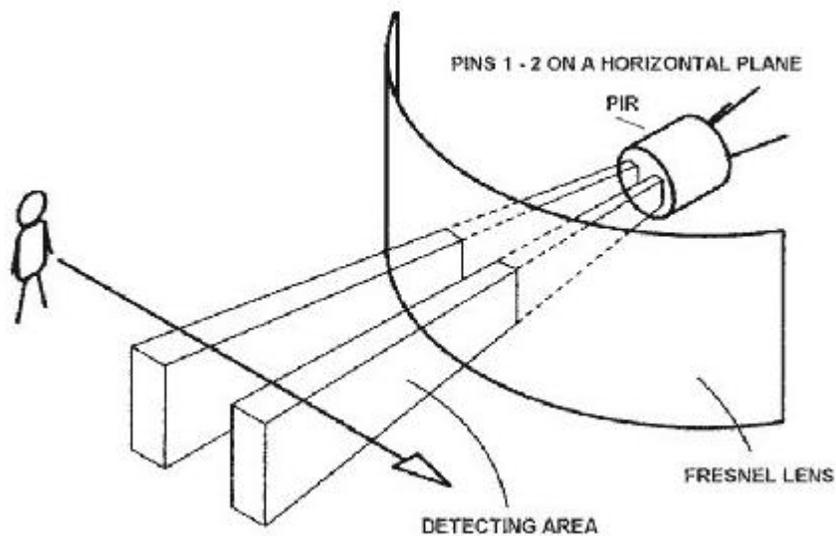


Figure 4: Method of Detection by Pyroelectric Sensor

The chip accounts for these cancellations and is labeled “movement” across the sensor, giving a high value to the output of the module. The BISS0001 chip is also responsible for not allowing multiple detection at once by keeping the module’s output high for a period of time, T_x , and not allowing for another detection for another period of time, T_i .

The pyroelectric sensor detects radiation in two narrow paths which does not allow for wide range of detection. In order to enlarge the detection range, a simple lens is placed over the sensor as seen in Figure 3. The lens is a Fresnel lens that condenses light providing a larger range of detection as depicted in Figure 5.

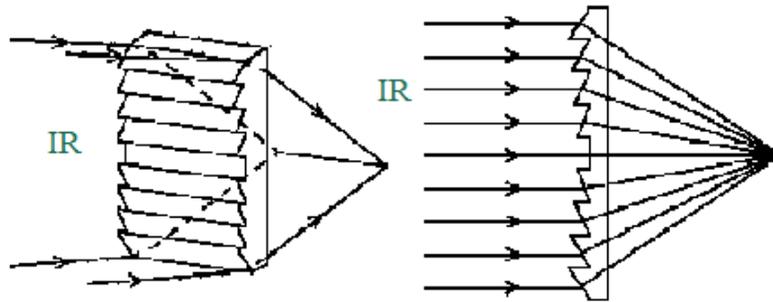


Figure 5: Fresnel Lens Condensing Light to a Focal Point

The lens is then split into multiple sections each a smaller Fresnel lens. This allows for a larger area with scattering detection areas as depicted in Figure 6 below.

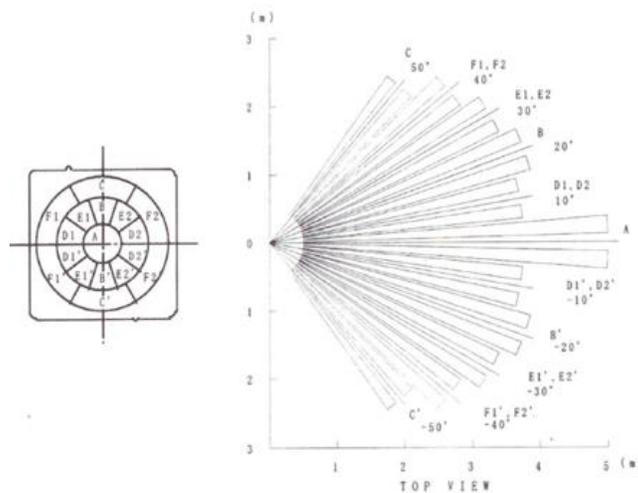


Figure 6: Scattering Detection Areas by the Fresnel Lens

In order to combine all of the modules together, we decided to use the Digilent Nexys board. The board is centered on the Spartan 3 FPGA and allows for quick setup of design circuits. It has many input and outputs, an A/D and D/A converter, sensor input, motor control and many other built-in features. The Nexys board's on-board oscillators are useful when needing an internal clock. We will use the clock to have our code check the sensor inputs every rising edge to continuously and rapidly ensure that the sensor are working correctly. Figure 7 displays the software flow chart of the overall system.

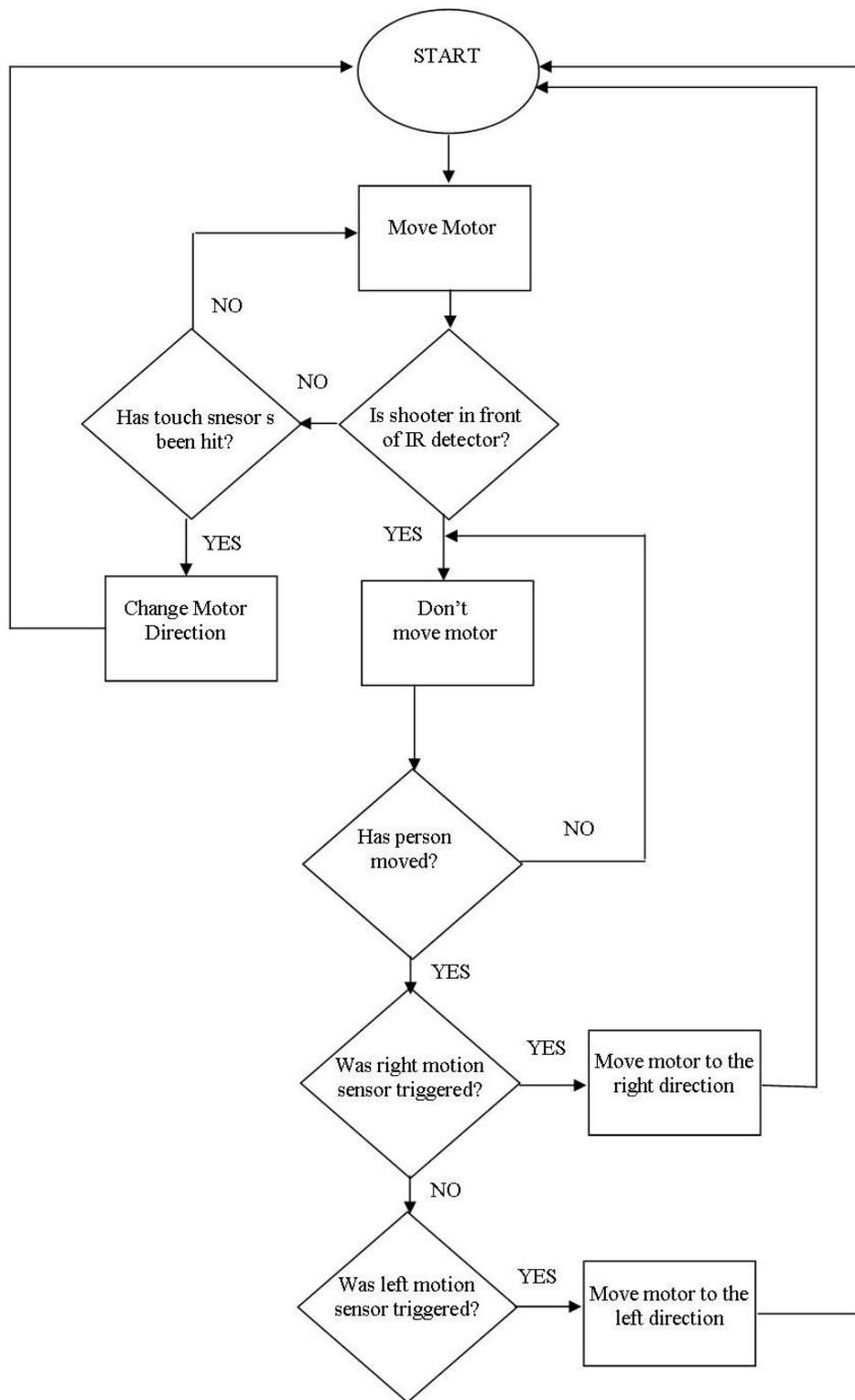


Figure 7: Software Flowchart of Overall System

V. TEST PLANS

In order to ensure the proper functionality of the overall project, each module will be tested individually. To begin, the IR distance sensor will be placed on a work bench pointed in the direction of open space. We want to ensure that the sensor is able to work properly at the necessary distance range. We will provide it with 5V DC and measure the voltage for the required distance. We will measure the voltage using a digital multimeter provided in the lab. The motion sensors will also require testing for its required range. In order to ensure that the detectors only detect motion that we provide, the tests must be done in an area where there are no other heat sources that can be sensed by the detector. Once again, the Parallax sensor was chosen for its convenience of requiring the same amount of power as the IR sensor. We will provide the module with 5V DC and measure its output for the required distance range. We will connect a light emitting diode (LED) to illuminate when detection occurs. The module provides a digital output of either high or low. We will use an oscilloscope to measure the time it takes between detections.

After writing the source code for the project, we will use a small motor that will be connected to the Nexys board. By simulating digital highs and lows to the board, we intend to prove the source code is working properly before we interface all the modules to the Nexys board. After ensuring that the code is functioning properly, we will interface all of the modules and the touch sensors to the Nexys board and validate the smaller motor is responding correctly to the program. The final test will be to connect the larger drill motor and ensure that it turns the funnel correctly according to all sensors.

VI. DEVELOPMENT AND CONSTRUCTION

The overall development and construction was done on a Quikshot USA basketball return chassis, which all of the sensors and motors are attached. The IR object location sensor, which is attached to the bottom of the funnel, is attached using pre-existent holes. In order to adjust the height of the sensor, we used metals rods with adjustable lengths. This allowed us to get the correct angle for the shooter detection. The PIR sensors are in charge of locating the shooter when they move from the center, and are attached on the right and left sides of the QuikShot chassis. These sensors have IR blocking enclosure in order to focus the IR detection to 90 degrees on each side.

The main challenge was the adding the mechanical movement to the chassis. We utilized a bike chain and a high torque 4V DC motor attached to a bike gear. Using metal fencing fasteners the motor was attached to the chassis. Attaching the bike gear to a drill bit was done with tack welding in order to try and ensure perfect perpendicular placement. We used extra strength industrial hot glue to glue the bike chain around the circumference of the funnel. All other non moving parts we screwed into the chase using previously drilled holes.

VII. INTEGRATION AND TEST RESULTS

Before integrating all the modules together, each was tested to ensure they worked as we expected. First, we test the distance sensor. As previously mentioned, we wanted the sensor to detect a person a distance of at least 20ft. To test, we connected the sensor to a bread board and applied 5V DC to its input. We then measured its output with a digital multimeter to make sure its output had enough voltage to be used as a recognizable detection for the Nexys board. We pointed the sensor to open air and measured a voltage of only a few mV. We stood in its line of

sight and stepped backwards continuously measuring its output voltage and plotting the results in Excel. The sensor did not reach the distances we had expected, dropping its output voltage near 0V after only 8ft. We realized that the IR light emitted by the room and the open air was causing interference to the reflected IR beam that we wanted the sensor to detect. In order to fix the problem, we added a cardboard tube around the module to have a focused beam returning to the sensor in the direction we wanted. Retesting the module, we were able to reach distances that we hoped for. Both Excel plots are shown in the Figure 8 below.

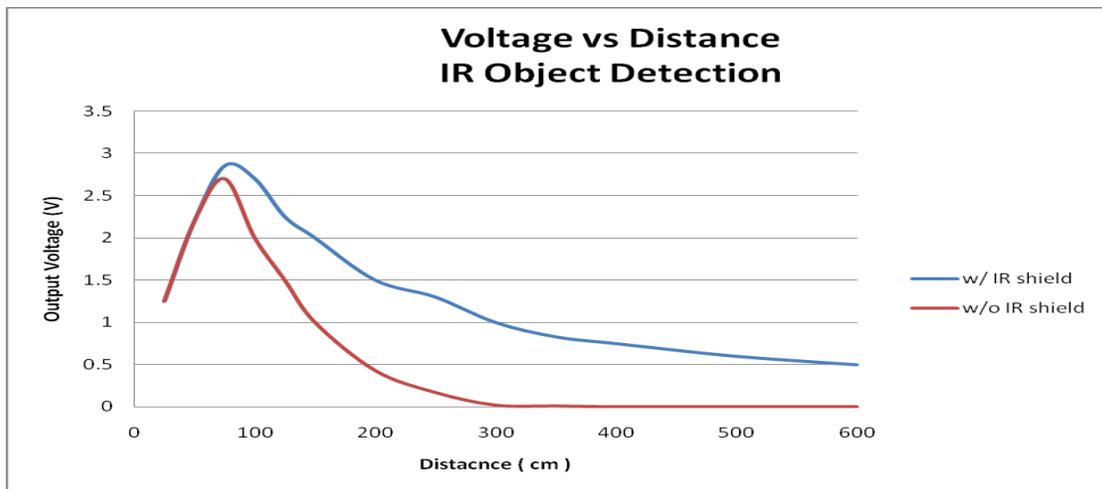


Figure 8: Excel Plots of IR Distance Sensor

The next module we tested was the Parallax motion sensor. We isolated ourselves to make sure that the module only picked up motion that we intended it to. We provided the module with 5V and measured its output using an oscilloscope. First, we wanted to see what the output looked like. The modules had very high sensitivity and worked very well. We had to use pieces of cardboard to direct the modules PIR sensor to a designated location. Then, we moved ourselves in its line of sight at different distances. The sensor worked extremely well all the way to the required distance of 20ft. The output was a digital value of low, 0V for no detection, and high, 3V when detection occurred. We used a LED to visual see its affect. The LED would light

up when detection occurred. The next requirement we had to test was the speed of the sensor between detections. Using the oscilloscope, we could see how long the sensor output a high value and how long before the sensor allowed another detection to occur. Adding both times together, the time between detections was estimated at 3sec. We thought this time was much too high considering that a person could stop moving within the time frame and the motor would not move. Looking at the data sheet of the BISS0001 chip, we realized that the time was controlled by external resistors and capacitors located on the module as seen in Figure 9.

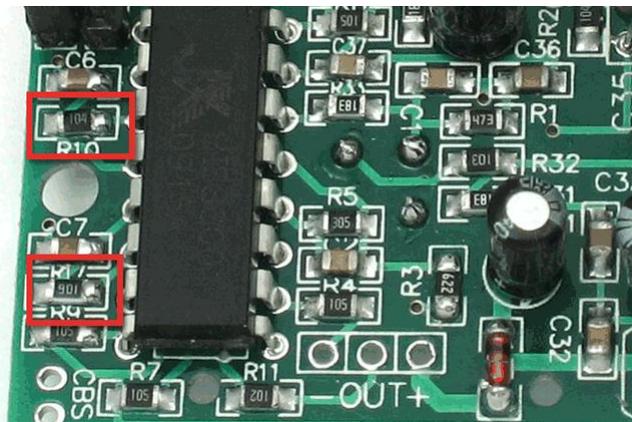


Figure 9: External Resistors and Capacitors Controlling T_x and T_i

Resistor R_{10} and capacitor C_6 controlled time T_x , which is the amount of time the output stays high when detection occurs, and resistor R_9 and capacitor C_7 controlled time T_i , which is the amount of time where detection is guaranteed not to occur. We noticed that R_9 and R_{17} mislabeled on the module, but looking at the datasheet cleared the problem up. Knowing that the time constant is equal to the product of the resistance and the capacitance and including two constants provided by the datasheet, we concluded that lowering the resistance of both times would essentially lower the entire detection rate.

$$T_x \approx 24576 \times R_{10} \times C_6 \quad (1)$$

$$T_I \approx 24 \times R_9 \times C_7 \quad (2)$$

Using a multimeter, we measured R_{10} and R_9 to equal $4.7\text{K}\Omega$ and $470\text{K}\Omega$, respectively. We measured the time of T_x and T_I to equal 1.2s, giving a total time of 2.4s (even though we measured 3sec with a timer). Using equations 1 and 2, we calculated capacitor values of $C_6 \approx 10\text{nF}$ and $C_7 \approx 0.1\mu\text{F}$. By using a desired result of $T_x + T_I = 1\text{sec}$, we calculated that the resistor values needed to be $2\text{K}\Omega$ and $208\text{K}\Omega$. We soldered two resistors, $3.3\text{K}\Omega$ and $330\text{K}\Omega$, in parallel with resistors R_{10} and R_9 , respectively. After retesting the time, we were able to lower the detection rate from 3sec to 1.2sec, which worked even better.

After testing all the modules, it was time to write the code in order to integrate everything together. We tested the code with a small motor and simulated highs and lows with external signals. Once we were satisfied that the code was working correctly, we connected all the modules and sensor to the Nexys board. The motor performed as expected and all that was left was to connect the larger drill motor and our funnel to the Nexys board.

The final part of our project was to test the DC motor. We took the drill apart and looked at its circuitry. Using the lab equipment, we had trouble getting the motor to run without the use of the battery provided by the drill. After careful troubleshooting methods and several blown chips, we determined that our problem is high start-up current for the DC motor. In order for the motor to begin rotating, it requires a current of at least 20 A. What we would need to do, if time permitted, is to build extra circuitry to allow for the 20 A to start our motor. One option is to use two high current single pole double throw (SPDT) relays that will be controlled by the outputs of the Nexys board. Another option is to build an H-bridge with high current transistors. The Nexys board outputs will also control which transistors turn on to control the motor direction.

VIII. Conclusion

Although our idea seems simple enough, the complexities in the details of the project allowed us to use what we have learned in our student careers. This senior project allowed us to combine our love for basketball with Electrical Engineering, making it both fun and challenging at the same time. The Automated Basketball Return System provides a practical application that can be used by anyone, not just Electrical Engineers. If we had more funds we might have been able to make this project a marketable item, but we are happy showing off what we have learned any way. The mechanical aspects of the project gave us a small taste of Mechanical Engineering as well. All our sensor and modules performed as expected after several alterations and tweaks. We are glad to have the opportunity to combine, both programming, which is not our strongest aspect of Electrical Engineering, and circuit theory in our project. We have not tested our design in a gym setting so we are not sure that the environment will change the overall effect. Either way, we are confident that our design will hold true after small alterations to its new environment.

Unfortunately we did not have the large motor running to turn our funnel. Even though we did not link the large motor to the output of the Nexys board, we are still able to manually turn the funnel using the motor's battery and show the 180 degree rotation of the funnel. We can also show that the source code works as expected by using the much smaller motor that does not require the large amount of current. Other than the large motor, our project will show proof of concept for our idea and hopefully allow for future Electrical Engineering students to modify the design.

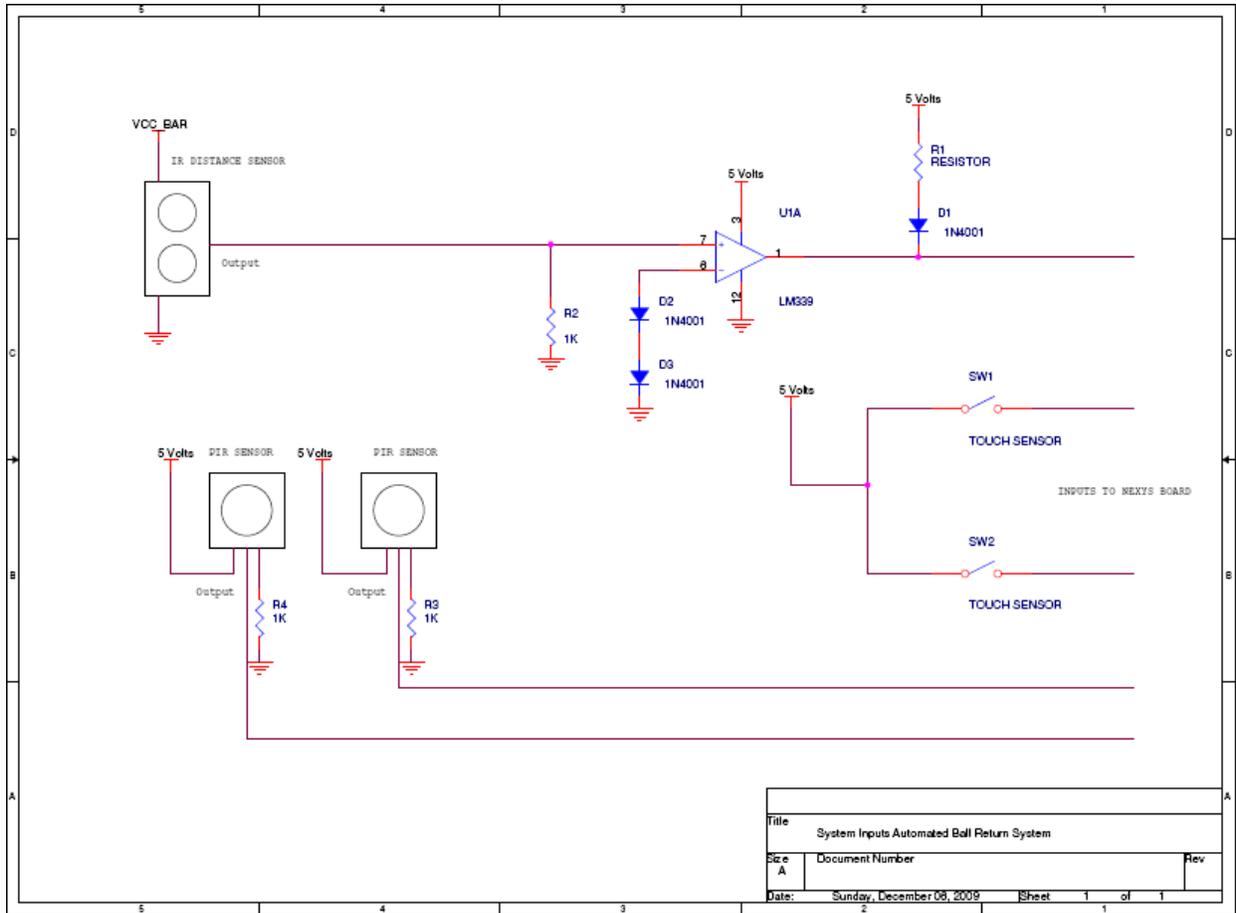
IX. References

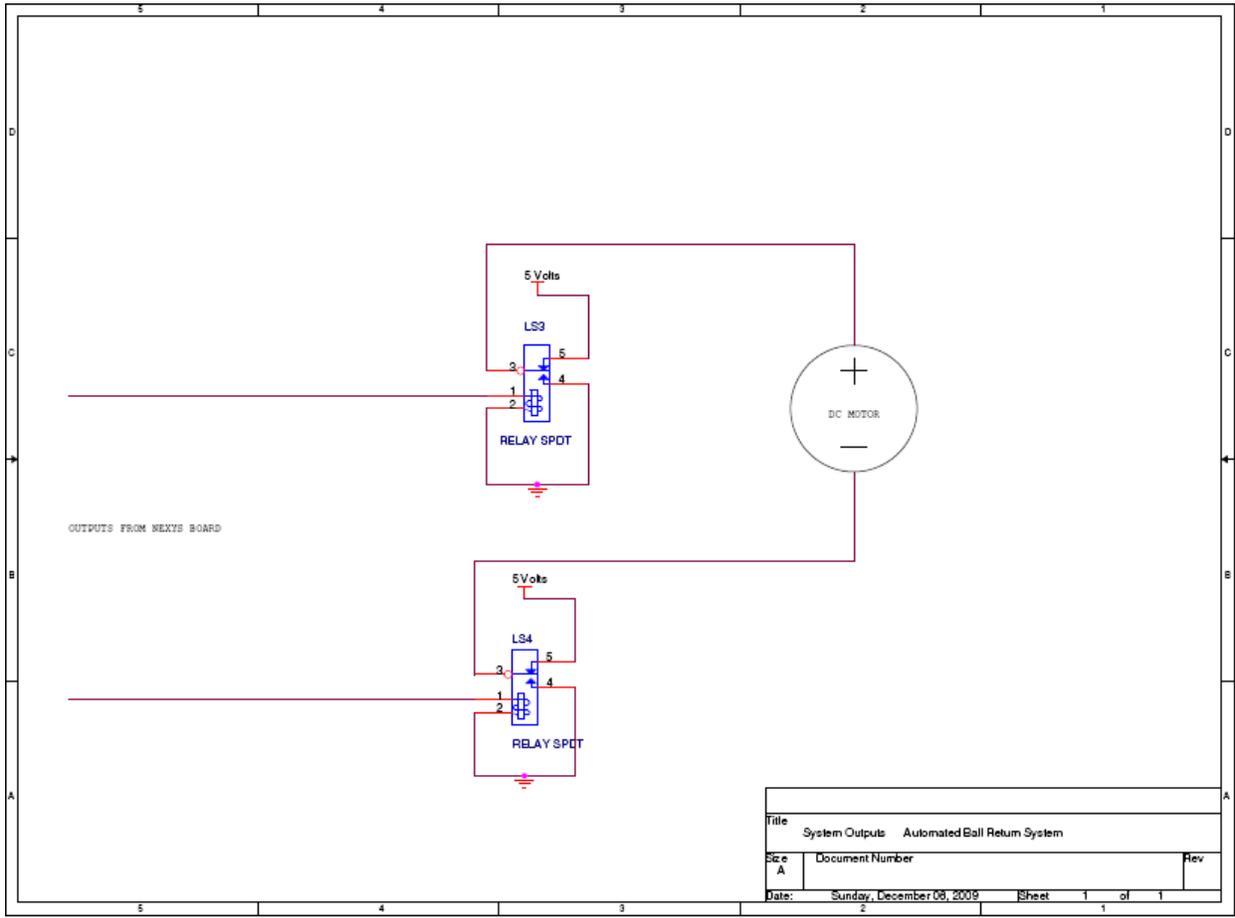
1. Franco, Sergio [Design with Operational Amplifier and Analog Integrated Circuits](#) McGraw-Hill, Inc 1988.
2. Cook, David [Intermediate Robot Building](#) Apress, 2004
3. Scary Terry [Report on Parallax PIR Sensor](#), last update 8/2009
<http://www.scaryterry.com/itw/pirsensor/pirsensor.htm>
4. Acroname Robotics [Sharp IR Rangers Information](#), last update 1/10/2009
<http://www.acroname.com/robotics/info/articles/sharp/sharp.html>

APPENDIX A: PART LIST AND COSTS

PART	QTY	COST	TOTAL
QuickShot Basketball Funnel	1	\$100 each	\$100
Sharp IR distance sensor	1	\$20 each	\$20
Parallax PIR motion sensor	2	\$10 each	\$20
Ryobi 4V Hand-Drill	1	\$30 each	\$30
Miscellaneous Components			\$5
		Overall	~\$175

APPENDIX B: HARDWARE AND SCHEMATICS





Title		
System Outputs Automated Ball Return System		
Size	Document Number	Rev
A		
Date:	Sunday, December 08, 2009	Sheet 1 of 1

APPENDIX C: SOURCE CODE

```
#include "xparameters.h" //Import the parameters created by the EDK
#include "xio.h" //Import the functions created by EDK

#define ASCIIOFFSET 0x30
#define SSRREG 0x070
#define CTRLREG 0x060
#define DATAREG 0x068

int leftSensor = 0;
int rightSensor = 0;
int rightMotor = 0;
int leftMotor = 0;
int bothSensor = 0;
int distance = 0;
/**
 * This function sends the given character out to the display and waits enough time for it to actually display.
 * @param char The character to print.
 * @return Always returns 0
 */
int printChar(char x)

{
    int i; //Counter Variable
    XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x04); //Set E Low
    XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x05); //Set E high
    XIo_Out32(XPAR_LCD_DATA_BASEADDR, x); //Output the Character
    XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x04); //Send E low, writing out character
    for(i = 0; i < 1700000; i++); //Loop for a 1.52ms to allow character to hit LCD
    return 0; //Return 0 for technicality
}
/**
 * This function clears and initializes the LCD, preparing it for output.
 * @return Always returns 0
 */
int initializeLCD()
{
    int i; //Delay Counter
    int GPOffset = 4; //Offset for Register to control Data Direction

    // Initialize Everything - Set both GPIO's to Output Mode
    XIo_Out32(XPAR_LCD_DATA_BASEADDR + GPOffset, 0x00); //Set Data bus to Output
    XIo_Out32(XPAR_LCD_CTRL_BASEADDR + GPOffset, 0x00); //Set CTRL bus to Output

    //Display Control
    XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x01); //Read OE
    //No Cursor: XIo_Out32(XPAR_LCD_DATA_BASEADDR, 0x0C);
    XIo_Out32(XPAR_LCD_DATA_BASEADDR, 0x0C); //Output '00001100' Display On cursor&blink off
    XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x00); //Write OE
    for(i = 0; i < 55000; i++); // Delay 37us

    //Display Clear
    XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x01); //Read OE
    XIo_Out32(XPAR_LCD_DATA_BASEADDR, 0x01); //Output '00000001' Clear the display
    XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x00); //Write OE
    for(i = 0; i < 1700000; i++); // Delay 1.52ms
```

```

//Entry Mode
Xlo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x01); //Read OE
Xlo_Out32(XPAR_LCD_DATA_BASEADDR, 0x06); //Output '00000110' for Right moving cursor, no display shift
Xlo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x00); //Write OE
    for(i = 0; i < 55000; i++); // Delay 37us

// Function Set
Xlo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x01); //Read OE
Xlo_Out32(XPAR_LCD_DATA_BASEADDR, 0x38); //Output '00111000' for 8 Bit, 2 Lines, 5x8Dot font
Xlo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x00); //Write OE
    for(i = 0; i < 55000; i++); // Delay 37us

return 0; //For tehcnicality

}
/**
 * This function controls the motors movement based upon sensor inputs
 * @return Always returns 0
 */
void motorControl()
{
    if ( bothSensor == 1 )
    {
        Xlo_Out32(XPAR_MOTOR_BASEADDR, 0x00); //Set Both Motor to Output low
    }

    else if (rightSensor == 1)
    {
        Xlo_Out32(XPAR_MOTOR_BASEADDR, 0x01); //Set Right Motor to Output high
    }

    else if (leftSensor == 1)
    {
        Xlo_Out32(XPAR_MOTOR_BASEADDR, 0x02); //Set Left Motor to Output high
    }

    else
    {
        Xlo_Out32(XPAR_MOTOR_BASEADDR, 0x00); //Set Both Motor to Output low
    }
}

void printStatus()
{
    int i;

    //Display Clear
    Xlo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x01); //Read OE
    Xlo_Out32(XPAR_LCD_DATA_BASEADDR, 0x01); //Output '00000001' Clear the display
    Xlo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x00); //Write OE
    for(i = 0; i < 1700000; i++); // Delay 1.52ms

    if ( bothSensor == 1)
    {
        char tempstring1[14] = "BOTH SENSORS!!";
        for(i = 0; i < 14; i++)

```

```

        {
            printChar(tempstring1[i]);
        }
    }

    else if ( rightSensor == 1 )
    {
        char tempstring2[14] = "SENSOR RIGHT!!";
        for(i = 0; i < 14; i++)
        {
            printChar(tempstring2[i]);
        }
    }

    else if ( leftSensor == 1 )
    {
        char tempstring3[13] = "SENSOR LEFT!!";
        for(i = 0; i < 13; i++)
        {
            printChar(tempstring3[i]);
        }
    }
else
{
    char tempstring4[13] = "NOTHING!!!!!!";
    for(i = 0; i < 13; i++)
    {
        printChar(tempstring4[i]);
    }
}
// Second Line
XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x01); //Read OE
XIo_Out32(XPAR_LCD_DATA_BASEADDR, 0xC0); //Try to goto 2nd line
XIo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x00); //Write OE
for(i = 0; i < 55000; i++); // Delay 37us

if ( distance == 1 )
{
    char tempstring5[13] = "DISTANCE!!!!!!";
    for(i = 0; i < 13; i++)
    {
        printChar(tempstring5[i]);
    }
}
else
{
    char tempstring6[13] = "NO DISTANCE!!";
    for(i = 0; i < 13; i++)
    {
        printChar(tempstring6[i]);
    }
}
}

void printSearch()
{

```

```

int i;
//Display Clear
Xlo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x01); //Read OE
Xlo_Out32(XPAR_LCD_DATA_BASEADDR, 0x01); //Output '00000001' Clear the display
Xlo_Out32(XPAR_LCD_CTRL_BASEADDR, 0x00); //Write OE
for(i = 0; i < 1700000; i++); // Delay 1.52ms

char tempstring0[14] = "SEARCHING!!!!";
    for(i = 0; i < 14; i++)
    {
        printChar(tempstring0[i]);
    }
}

/**
 * This main functon is the one that actually operates upon poweron.
 */
int main()
{
    int i;
    //Call the initialize Functions for LCD
    initializeLCD();

    Xlo_Out32(XPAR_IR_SENSOR_BASEADDR + 0x04, 0xFF); //Set Sensors to Input
    Xlo_Out32(XPAR_BUMP_SENSOR_BASEADDR + 0x04, 0xFF); //Set Sensors to Input
    Xlo_Out32(XPAR_DISTANCE_SENSOR_BASEADDR + 0x04, 0xFF); //Set Sensors to Input
    Xlo_Out32(XPAR_MOTOR_BASEADDR + 0x04, 0x00); //Set Motors to Output

    leftSensor = 1;
    bothSensor = 0;
    rightSensor = 0;
    distance = 0;
    motorControl();
    control();
    printStatus();

    while(1)
    {
        if((Xlo_In32(XPAR_DISTANCE_SENSOR_BASEADDR) & 0x01) == 0x01) // Check is Distance Sensor
        {
            leftSensor = 0;
            rightSensor = 0;
            bothSensor = 0;
            distance = 1;
        }
        else
        {
            if((Xlo_In32(XPAR_IR_SENSOR_BASEADDR) & 0x03) == 0x03) // Check if both sensors
            {
                rightSensor = 0;
                leftSensor = 0;
                bothSensor = 1;
                distance = 0;
            }
            else if((Xlo_In32(XPAR_IR_SENSOR_BASEADDR) & 0x01) == 0x01) //Check right sensor
            {
                rightSensor = 1;
            }
        }
    }
}

```

```

        leftSensor = 0;
        bothSensor = 0;
        distance = 0;
        motorControl();
        control();
    }
    else if((XIo_In32(XPAR_IR_SENSOR_BASEADDR) & 0x02) == 0x02) //Check left sensor
    {
        leftSensor = 1;
        rightSensor = 0;
        bothSensor = 0;
        distance = 0;
        motorControl();
        control();
    }
}
motorControl();
printStats();
}

return 0; //For tehcnicality
} //End Main

int control()
{
    while(1)
    {
        if((XIo_In32(XPAR_DISTANCE_SENSOR_BASEADDR) & 0x01) == 0x01) // Check is Distance Sensor is hit
        {
            leftSensor = 0;
            rightSensor = 0;
            bothSensor = 0;
            distance = 1;

            return 0;
        }
        else if((XIo_In32(XPAR_BUMP_SENSOR_BASEADDR) & 0x01) == 0x01) // Check is Left Touch
        {
            rightSensor = 1;
            leftSensor = 0;
            motorControl();
        }
        else if((XIo_In32(XPAR_BUMP_SENSOR_BASEADDR) & 0x02) == 0x02) // Check is Right Touch
        {
            rightSensor = 0;
            leftSensor = 1;
            motorControl();
        }
        else
        {
            distance = 0;
            printSearch();
        }
    }
}

return 0;
}

```

APPENDIX D: ANALYSIS OF SENIOR PROJECT DESIGN

Project Title:		
Student's Name:	Student's Signature:	
Student's Name:	Student's Signature:	
Advisor's Name:	Advisor's Initials:	Date:

• Summary of Functional Requirements

The project is meant to assist a basketball player in their practicing. The project hangs directly on the basketball rim and automatically rotates to find the player on the court. Once the player is detected, the automated basketball return system (ABRS) stays focused in the direction of the player so that any baskets successfully made will be funneled back to the player. When the player moves from center, IR motion detectors find which direction you moved to and rotates accordingly, until the player is focused in front of the funnel again. If at any time the player is absent from the court or cannot be located, the rotation continues in a 180 degree sweeping motion until the player is picked up again.

• Primary Constraints

Challenges in implementation of the ABRS came mostly with the motor control. Using a 4V high torque motor we did not foresee the need for such extremely high current requirements. The current spike at start of the motor exceeds 20 Amperes, which our system was not set up for. Also, when implementing the IR motion sensors into the overall design, they needed a 4 second delay between adjacent signals. This was inefficient and would halt the basketball player when running around the court. In order to make our sensors work, we were able to reverse engineer the design and change the resistor values of the circuit in order to get the time down to under 1 second.

• Economic

- The original estimated cost of component parts was roughly \$150.
- The final total came to \$175.
- The original estimated development time was estimated 6 months.
- Actual development time equal to 8 months, although ours was a proof of concept model and not a full prototype.

- **If manufactured on a commercial basis:**

- Estimated number of devices to be sold per year = 10,000
- Estimated manufacturing cost for each device = \$50
- Estimated purchase price for each device = \$199.99
- Estimated profit per year = \$1,500,000

- **Environmental**

Environmental concerns are of extreme concern with every type of new electrical product. An assembly of our size uses high amounts of plastics and aluminum. The advantage of our product is its longevity and usage of reusable materials. Because of the heavy use of plastics involved with our product we would suggest a highly recyclable material. Structural and clarity concerns are usually the drawbacks in development when choosing recyclable plastic materials. In our case, these would not be an issue and could help with the environmental concern of massive production of the ABRS.

- **Manufacturability**

Challenges in manufacturability would be to get the most efficient design possible, in order to keep power consumption at a minimum. The overall product could be done in many separate parts and assembled easily together before being sold in market. The efficiency in system design is needed far more than any new manufacturing technique.

- **Sustainability**

Sustainability issues would most likely be due to rust and engine wear if the product was to be used outside. Also, while being battery operated, battery wear and longevity would be an issue. If upgrades were developed for the product, it would most likely be done in order to more perfectly control the rate at which the ball is returned to the shooter. In doing this, there would be an additional motor and most likely increase the failure rate of the system in whole.

- **Ethical**

New products usually involve some sort of ethical issues, either with competition or marketing strategy. The goal of our product entails only fun and exercise, and the advancement of skills in a sport we enjoy. We hope that the use of our product relates strictly to our purpose.

- **Health and Safety**

The Health and Safety issues with our product are limited to the motors movement and the high wattage battery used. The product, if manufactured, would be much safer if eliminating the gears and chains and switching to a belt driven system. Most likely, the product would be used to encourage good health with exercise with sports. The overall product development only had good intentions for the user.

- **Social and Political**

Any social impact with our product could only be for the positive. When people are using our product, they are using it for self improvement of their basketball skills. This keeps the user involved with a meaningful use of their time and could not be seen as detrimental in any social viewing.

- **Development**

When reverse engineering the timing component of our IR motion sensors, understanding the timing restraints associated with sensing was very difficult. We were able to reduce the resistance of the RC circuit to lower the time the signal is high and the delay before it was ready to sense another signal. Also, understanding the idea of startup current in a DC motor was unfamiliar prior to building this project.