Servo Motor-Propelled Ball Transfer System

Jack Kingsman (9 hours), Parker Newton (9 hours), Arun Sidhu (9 hours)

March 17, 2014, Team no. 206, ENGR 1 - Friday, Santa Clara University

Initial Strategy:

The initial plan was to receive the ball on a landing surface angled downwards, direct it onto a catapult, and automatically propel it to just above the height of the box (12"), delivering it to the next system. The landing ramp was to be constructed from a foam board, and the catapult was to be constructed from Popsicle craft sticks (for the base and lever arm), rubber bands (to secure the lever arm to the base and provide elastic potential energy), and a plastic spoon (to catch and propel the ball). To adjust the range of the ball propelled from the catapult, the launch velocity and angle were considered. A change in angular momentum imparted to the ball is responsible for the ball's launch velocity. A greater lever arm length will produce a larger torque, providing a larger change in angular momentum and thus a larger launch velocity. The number of Popsicle sticks inserted between the catapult's base and lever arm determine the initial angle of launch. After conducting several trials to test for optimal range, the lever arm was determined to be the length of 1 Popsicle stick (6"), and the initial angle was chosen to be 20 degrees. The primary impediment at this point was designing a method of automatically launching the cannonball. Several ideas were considered, including a counterweight which would be displaced by the ball rolling down the ramp. This would trigger the catapult and launch the ball. However, implementing this method proved to be extremely complicated and unrealistic.

Final Strategy:

The final strategy involves the use of an Arduino Starter Kit and dual ramps. The first ramp acts as the landing platform for the ball. Guides constructed from Popsicle sticks direct the ball as it moves down the ramp to the center of the platform. The second ramp is an incline placed opposite to the first and angled slightly lower than the first, so that the ball drops from the first and is caught in the space between the ramps. A light sensor (included in the Arduino kit) is mounted on the landing platform, and the guides direct the ball so that it rolls over the sensor. The light sensor triggers the servo motor by returning a value that indicates that the ball has rolled across it. The servo motor has an extension arm constructed from Popsicle sticks attached to it. When triggered, the extension swings through an angle of 120 degrees, colliding with the ball and propelling it up the incline and over the top of the system. The servo motor and light sensor are mounted underneath the ramp, and connected via a breadboard to an Arduino UNO board. The electrical system is powered by a 9-volt battery.

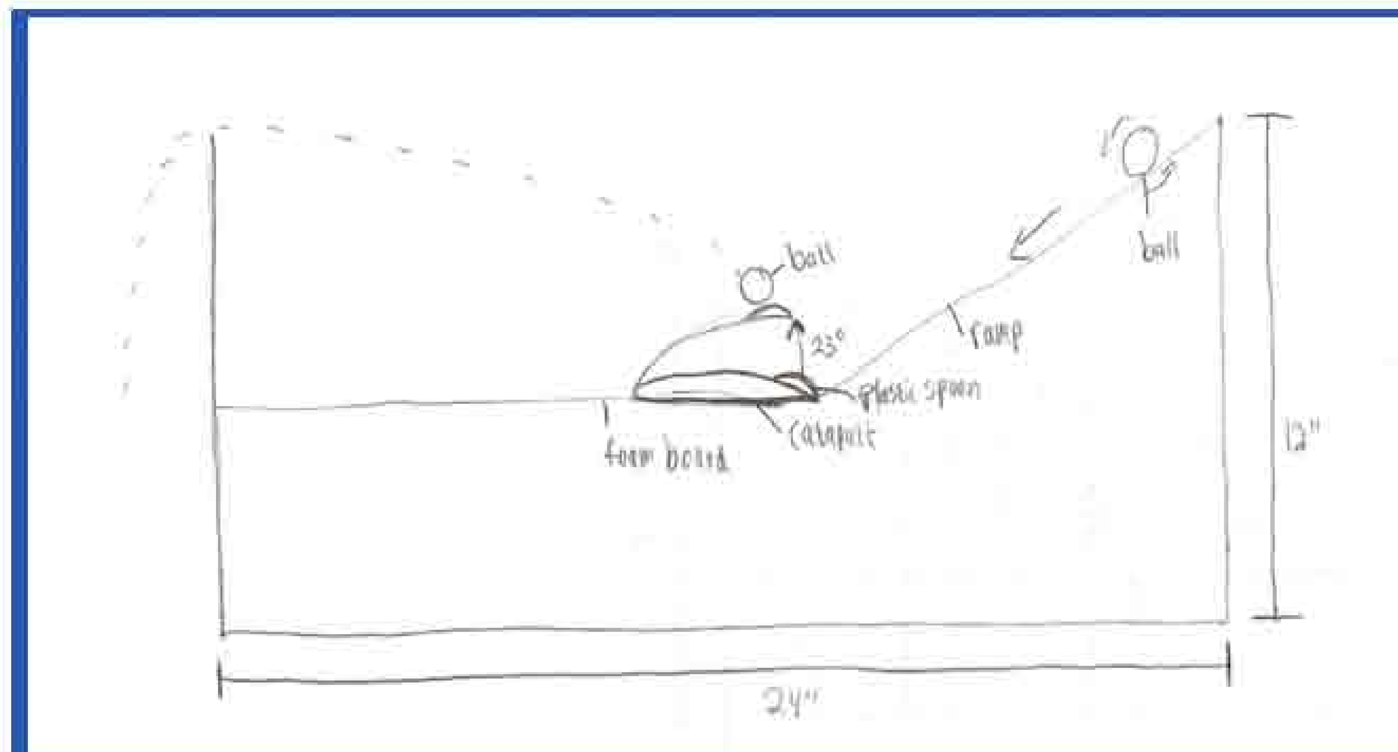

Figure 1: Initial Sketch

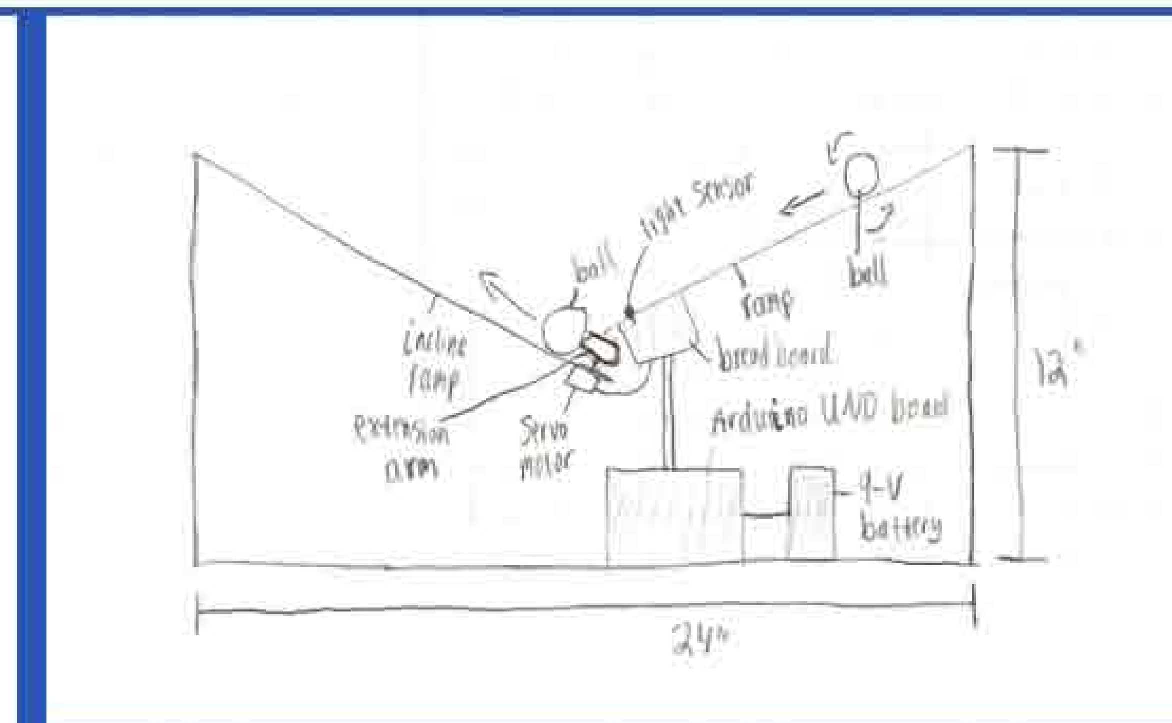

Figure 2: Final Sketch

Part	Price
(2) Foam boards	$14.55
(1) Craft sticks package	$3.99
(1) Rubber bands package	$1.99
(1) Duct tape roll	$4.99
(1) Arduino Starter Kit	$25.00
(1) Breadboard	$3.39
Total	*$53.91*

Figure 3: Expenses

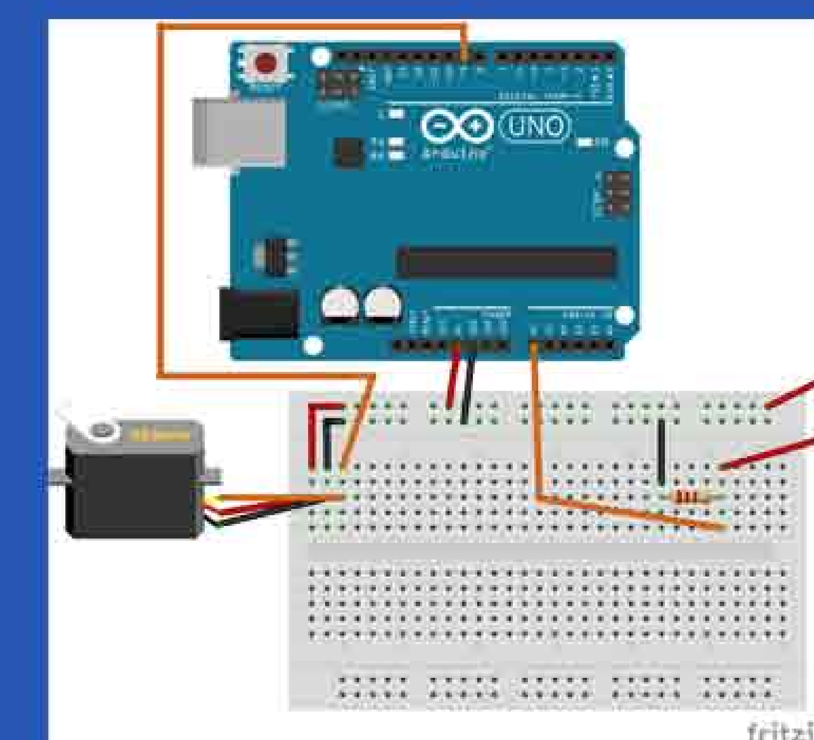

Figure 4: UNO Board and Breadboard Diagram

```c
#include <Servo.h>
#define MINDELTA 20


Servo myservo;  // create servo object to control a servo
                // a maximum of eight servo objects can be created

int pos = 0, baseval, basevalset = 0, sensorValue, i, consecutiveruns, first;



void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);  // binds to the serial port for logging
}


void loop()
{
    if(!basevalset || consecutiveruns > 3){
        Serial.println("Setting base value...");
        for(i=0; i<10; i++){
            baseval += .1*analogRead(A0); //get 10 values over 150ms for our average
        delay(15);
        }

        Serial.print("Base value set to ");
        Serial.println(baseval);
        basevalset=1;
    }else{

        // Read the input on analog pin 0:
        sensorValue = analogRead(A0);
        Serial.println(sensorValue);
```

Figure 5: Excerpt from Source Code (Programmed in C)

Figure 6: Final Product

Acknowledgements:

Thank you to Dr. Ali Abrishamchi, Dr. Tonya Nilsson, and Dr. Sally Wood.