Stamp Four: Connect Four with the Basic Stamp

ME 5643- Mechatronics: Integrated Term Project

December 14, 2015

Kyle Daniel

Brandon Peifer

Ming Kwok

Table of Contents

Introduction

A computer controlled connect four game, Stamp Four, was produced by integrating the Basic Stamp 2 with actuators, sensors, and control algorithms. The Stamp Four consists of two unique systems. A feeder system, made up of a ping sensor and servomotor was implemented to allow the basic stamp to physically place game tokens into the Connect Four board. A remote controller system, consisting of a limit switch, a LCD, and an accelerometer made up the user interface. Mathematical theories were used to create control algorithms for each of these systems. A separate algorithm was created for the Basic Stamp to play against the player.

Design Theory

Interface: Theory behind the Controller

A controller was specifically designed for user interface with "Stamp Four". The controller consists of a LCD, a Memsic 2125 2 axis accelerometer, and a limit switch mounted on foam. The user input is determined by the orientation of the accelerometer mounted on the controller. When the user tilts the controller, the accelerometer sends pulses as an output to Basic Stamp. The pulse measurements has an initial range of 1875 to 3125. After setting an offset and scaling the measurements, the range becomes integers from 0 to 6. The calculations are shown below.

$$x_{raw} = [1875, 3125]$$

$$x_{out} = (x_{raw} - 2500) \times \frac{6}{1250} + 3$$

$$x_{out} = [0, 6]$$

The numbers of 0 to 6 represents the 7 columns of the Connect Four game. The LCD provides feedback to the user by displaying the converted output of the accelerometer. The LCD is also programed to display which player's turn it is, who won the game, or if the result of the game was a draw. When the user reaches the desired value of the accelerometer after tilting the controller, the user confirms this value by pushing the limit switch that sends a true logic value to the basic stamp. Once the basic stamp receive the logic, it stores the converted accelerometer value and moves the feeder servo to the column corresponding to the value.

Computer Player

The computer will act on the first turn of the game and play every other turn after that. The computer follows a set of rules that it performs in order and ranks each column to choose where to place its piece. The rules of the computer are written in the subprogram AI are works as follows:

1) If the computer has a winning move, place the piece in that column.

2) If the player has a winning move, move there.

3) If a move will lead to an open winning space for either player, don't go there unless forced.

4) Pick the space with the highest number of combinations of wins.

The first step is to ensure a winning move whenever possible. The computer will pick that column above all else to end and win the game. If the step turns out to be false, then the computer will look for any moves that will lead to the player winning. If this is true, the computer will pick the column chosen in order to block the player from playing in that space and winning the game. If either of these steps are true, then the program will save the column selected, jump out of the subprogram, and move accordingly. After these step, it will then look for any move that will lead to an open win for either player. If this is true, the column will become the lowest rank for play. This is to prevent opening either the player from being able to win or block a potential win for the computer. All of the first three steps are accomplished by the same subprogram labeled AIscan within the code.

Once the first three steps are completed, with step 1 and 2 being false, the program will then look at the remaining columns and give them a ranking number. The first column with the highest rank is where the computer will place its piece. The method of appointing ranks is based on the number of possible win combinations of an empty board for each place. The actual number of winning combinations is shown below:

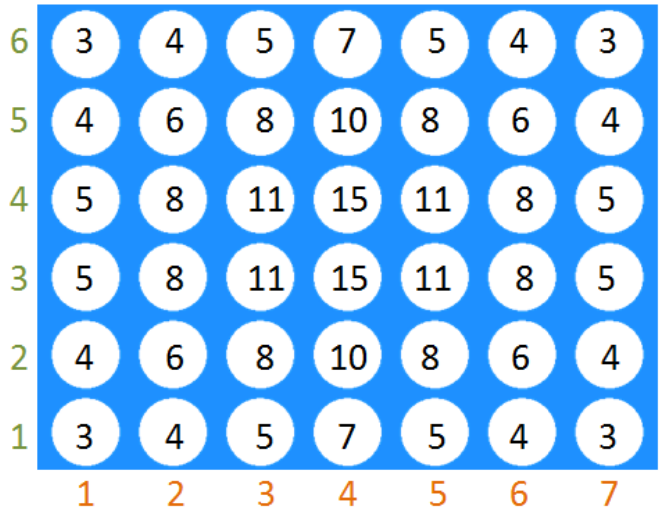

Figure 1: Probability to Win

However, this method cannot be placed into the basic stamp because it required 168 Bits of variable memory which overloads it. Therefore, this method was modified into an equation that accomplishes this method in the same way. The equation chosen is: rank=1+ (column number)*(row number). If the column number is greater than 4, then the basic stamp subtracts it from 8. Likewise, if the row number is greater than 3, then the basic stamp subtracts it from 6 to get the new row number used in the equation. The equation produces the new rank method for each spot as shown below:

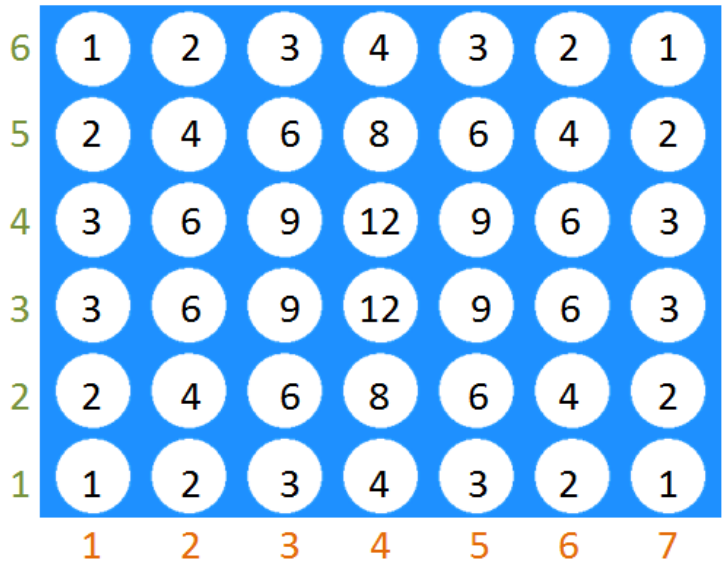

Figure 2: Modified Probability to Win

An example of what the computer would compute during a game is shown below

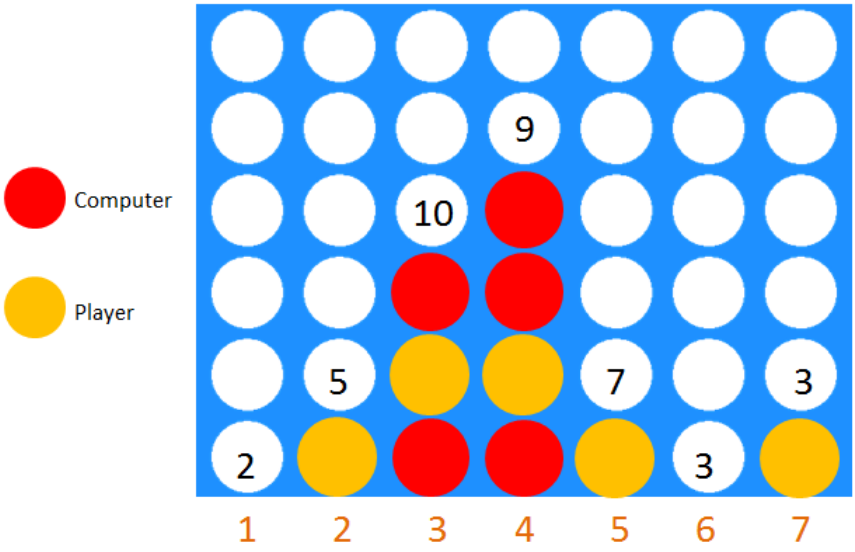

Figure 3: Example Game

This method for the computer produces a large amount of code and did overrun the memory of the EEPROM at one point. Because of this, different methods were created along the writing of it to allow for the same functionality and cut down on the amount of code needed to be stored.

Winning algorithm

The basic stamp will also find any combination of 4 of the same piece being in a row. To do this, a method for scanning vertically, horizontally, diagonally up and to the right, and diagonally up and to the left was created. Since the pieces are placed in 2 arrays of 41 bits, the basic stamp can scan both simultaneously to determine if a player has won. To start, the basic stamp scans vertically. It starts by looking to make sure the column has enough pieces to scan. This is to prevent the basic stamp from scanning a space that doesn't exist on the board. The basic stamp scans 4 spaces at a time of each of the 21 possible combinations. If each space has a piece and all of them equals the player that last played (because someone can't win that turn if they didn't play) then the basic stamp will declare the winner. The basic stamp does this the same way for all of the horizontal and diagonal combinations with the appropriate changes.

Player Movement with Controller

The player will use a controller to choose their desired column. The controller has a button and an accelerometer which are used in combination. The player tilts the controller to the left and right to change the column highlighted. The basic stamp takes in the pulse and converts it in the column and projects it on the LCD display. The button is pressed when the correct column is displayed and then the basic stamp will look up whether the column is full. If it is not, the basic stamp will save it and move to the rest of the code.

Placing the Pieces in the Connect 4 Board

The basic stamp will take the decided move and adjust the ramp to place the ball in the correct column. The basic stamp will use one standard servo, one continuous servo, and an ultrasonic sensor to achieve this. The basic stamp will first use the ultrasonic sensor to find the current position of the ramp (there was no more variable space to save the position last used). Then the basic stamp will decide which direction to move the ramp. That will turn the continuous servo until the ramp is just past the desired position where it will stop. Then the standard servo will turn one way to release a ball and then turn back to accept one ball. The ball released then rolls to the end of the ramp where it falls through the hole and into the correct position.

Mechanical Design

The mechanical design of the Stamp Four is divided into three main components: The feeder system, the ramp system, and the structure base. These three components are integrated to feed colored wooden balls to the proper columns of a purchased connect four board.

The feeder system holds the colored wooden balls and releases one ball to the ramp when either the player or the computer selects a column. The balls are held within a 4 ft. ½ in diameter vinyl tube and are purposely placed so the colors alternate. The tube is anchored to a vertical truss constructed out of bamboo skewers with hot glue. The truss is constructed of 6in bamboo skewers that are fastened together using hot glue. It spans five truss sections vertically (or about 30 in) and has a single bamboo skewer on either side of the base to stabilize the structure. Figure 4 below displays a side view of one truss section.

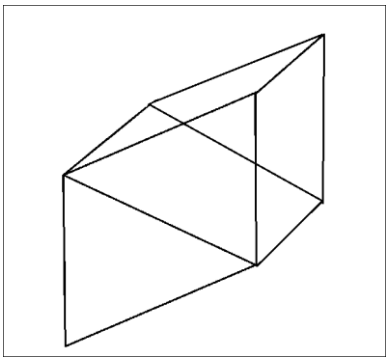

Figure 4: Single Vertical Truss Section

The vinyl tube is anchored to the truss at the sixth horizontal rear truss, the fourth horizontal rear truss, and the third horizontal forward truss. The feeding system is able to release one ball at a time from the vinyl tube. This process is possible through a mechanical device inserted into the bottom end of the feeding tube. The device consists of a bamboo skewer with two half disks fasted along its lengths. This device is picture in figure 5 below.

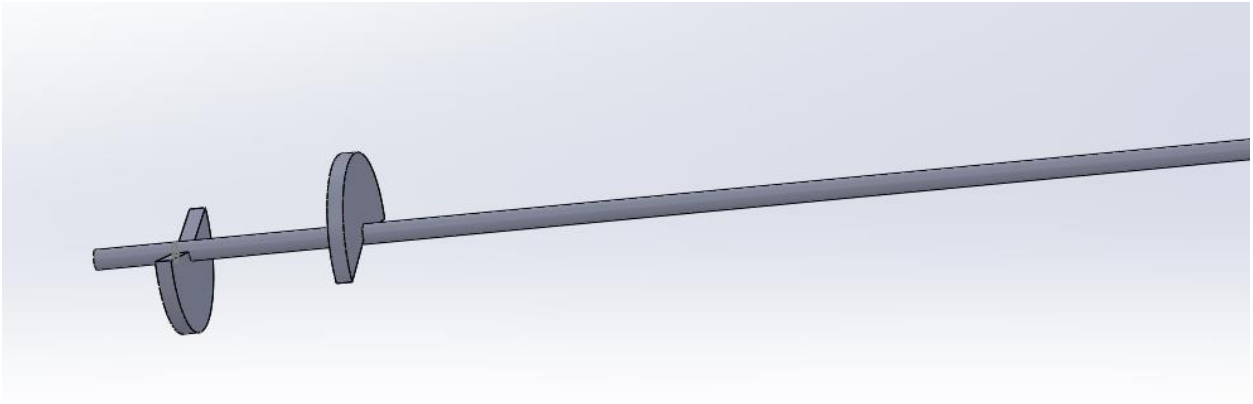

Figure 5: Feeder Device

The device is implanted into the bottom side of the vinyl tube right before the ramp. Two small slits are cut into the tube to allow the disks to enter the tube and interrupt the flow of wooden balls.

In the starting position the aft disk (furthest to the right) holds wooden balls back from entering the ramp. As the rod turns, the disks turn along with it. The aft disk turns, allowing balls to enter the space in between the disks, while the forward most disk blocks the balls from entering the ramp. The rod continues to turn allowing the ball in the chamber to be released while adding a second ball to the chamber. The feeder device is controlled by a standard servo that rotates the rod back and forth. The servo is mounted in the vertical truss and attached to the feeder device rod using hot glue. The feeder system of the prototype is picture below.

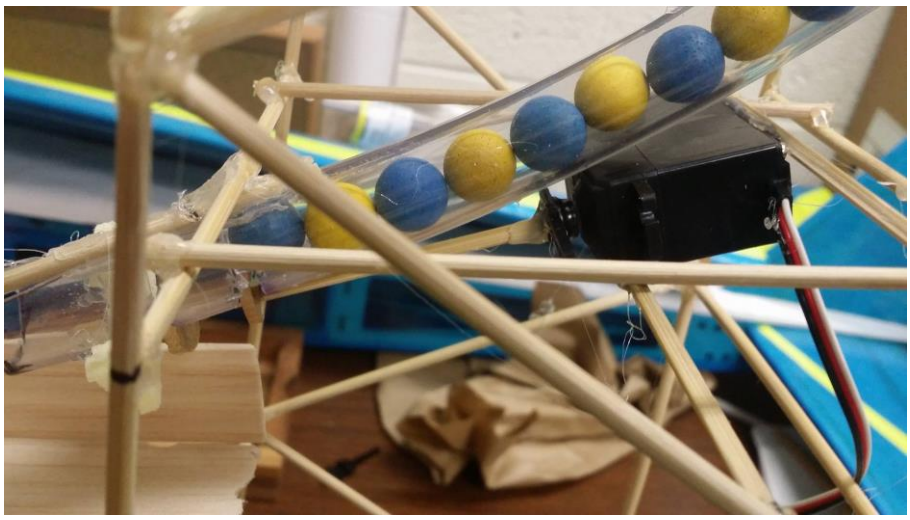

Figure 6: Prototype Feeder Device

The ramp system consists of a channel that receives the wooden ball from the feeder system and deposits it in the correct column of the game board. The channel is made of balsa wood and is approximately 3in wide by 1.5 ft. long. The channel is fasted with hot glue to a piece of hard insulation foam cut into a wedge shape. In this configuration the ramp sits and an angle so balls released from the feeder system roll down the ramp and through a hole drilled into the forward end. The insulation foam is subsequently fasted with hot glue to a platform that can be driven forward and backward with a gear system housed in the structure base. The platform is made of LEGO pieces and has a rack that interfaces with the pinion gears below it. The platform has raised edges that act as rails and keeps the system straight. Further, the platform has a vertical blocker at the aft end which the ping sensor reads to determine its position. The ramp system has safety stops at the far aft and forward ends. This stops the ramp system from going too far forward or backwards and falling off of the structure base.

The structure base is made primarily of LEGOS and houses the ping sensor, the continuous servo, the basic stamp, and the gear train that drives the ramp system. Two towers approximately 5 in tall house to pinion gears on which the ramp system sits. One of these gears is driven by a continuous servomotor that is similarly housed in a tower of LEGO pieces. This housing fastens the servomotor into place. The servomotor it fastened to the gear rod using hot glue. The ping sensor sits on a 5 in tower of LEGO pieces behind the ramp system. The ping sensor is at the same level as the vertical blocker on the back of the ramp system, and uses this marker to sense where the ramp is in space. The basic stamp is housed directly in front of the ping sensor tower. The integration of all these systems is shown below in the picture of the prototype.

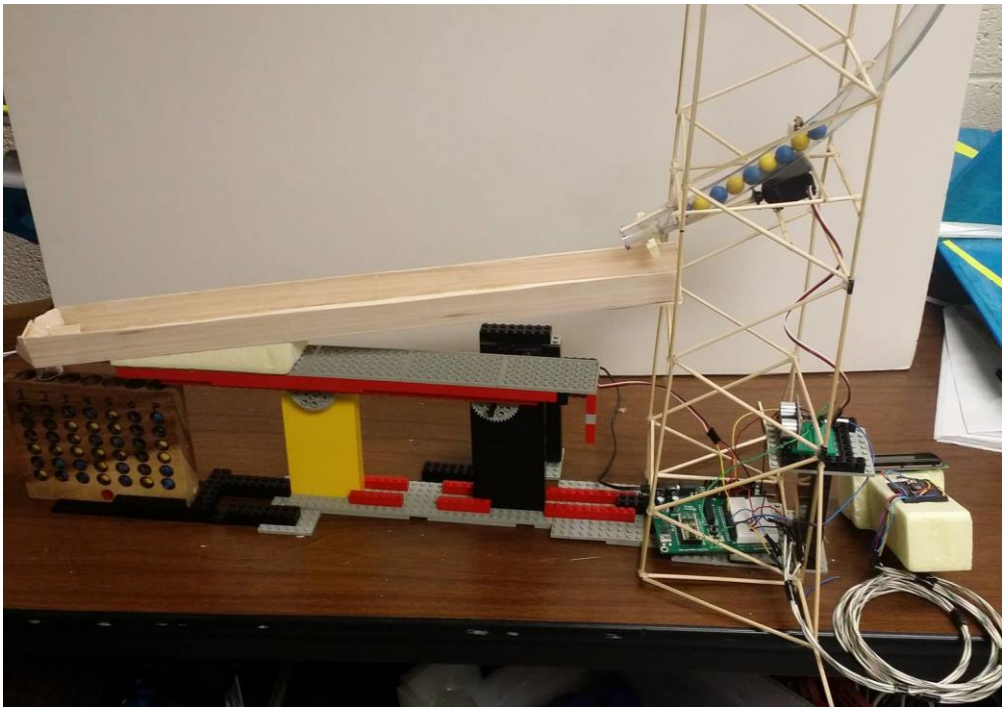

Figure 7: Prototype

Bill of Materials

The materials used in the design of the stamp four prototype are as follows.

Prototype Cost Analysis		
A. Mechanical Components	Description	Cost
Bamboo Skewers	1 Pack 6in (set of 100)	4.00
Hard insulation foam	1 2in x 2ft x 8ft board	19.65
Lego Set	1 Basic Bricks deluxe kit	40.00
Balsa Wood Sheet	1 4in x 36in x 1/8in sheet	10.00
Connect Four Game	Game with pieces included	12.00
Vinyl Tubing	1 length 1/2in x 4 ft	4.00
Hot Glue	-	6.00
	Subtotal	$95.65
B. Electronics	Description	Cost
Microcontroller	Basic Stamp w/ board of education	99.00
Ping Sensor	Parallax Brand	22.49
Continuous Servo	Parallax Brand	13.99
Standard Servo	Parallax Brand	12.99
Accelerometer	Memsic 2125 Dual Axis	22.49
Serial LCD	Parallax 2in x 16 in	29.99
Pushbutton Switch	Digikey	1.00
Misc	Wires, resistors, ect.	5.00
	Subtotal	$206.95
	Total	$302.60

Cost Analysis for Mass Production

The major costs incurred in the prototype were through the sensors, LEGO set, and the sensors used. If the stamp four were to be mass produced the cost of the sensors would drastically decrease when bought in bulk. Further, the LEGO set could be replaced with 3D printed parts made specifically to house the various components. This would greatly reduce cost and streamline the design of the system. Also if mass produced, the board of education could be replaced by a circuit designed just with the basic stamp. This would further reduce the cost of design and would simply the electronic system used. Below is a cost analysis for mass production.

Mass Production Cost Analysis		
A. Mechanical Components	Description	Cost
Bamboo Skewers	1 Pack 6in (set of 100)	4.00
Hard insulation foam	1 2in x 2ft x 8ft board	19.65
3d printed components	-	10.00
Balsa Wood Sheet	1 4in x 36in x 1/8in sheet	10.00
Connect Four Game	Game with pieces included	12.00
Vinyl Tubing	1 length 1/2in x 4 ft	4.00
Hot Glue	-	6.00
	Subtotal	$65.65
B. Electronics	Description	Cost
Basic stamp 2	-	17.00
Ping Sensor	Parallax Brand	15.00
Continuous Servo	Parallax Brand	10
Standard Servo	Parallax Brand	10.00
Accelerometer	Memsic 2125 Dual Axis	18.00
Serial LCD	Parallax 2in x 16 in	25.00
Pushbutton Switch	Digikey	1.00
Misc	Wires, resistors, ect.	2.00
	Subtotal	$98.00
	Total	$163.65

Design Advantages & Disadvantages

The purpose of the project is to create an interactive game that requires only one player and offers a unique experience outside the realm of digital devices. The computer player offers a challenge and is not easily beaten. This can attract players of all ages and can act as a great family game. The design gives a unique experience where users can see device actuate. Further it offers an interactive interface where users choose their position using a motion detecting controller.
However, there are disadvantages to the design. The system is large, bulky, and not easily transportable. Further the cost of the system, around $160, is rather high.

Appendix

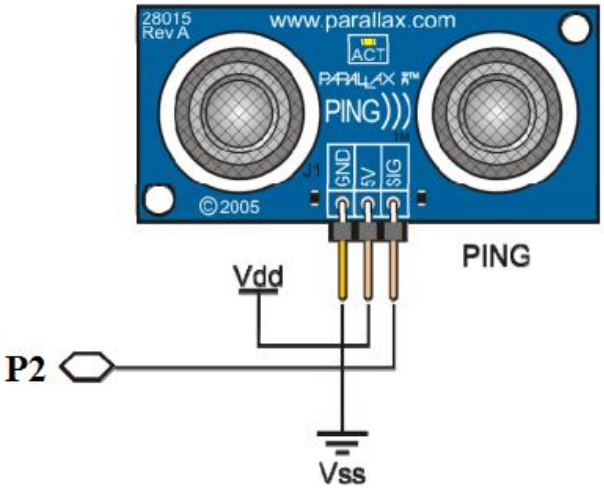

Figure 8: Ping Circuit

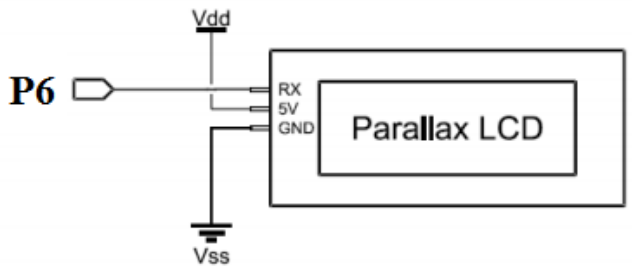

Figure 9: LCD Circuit

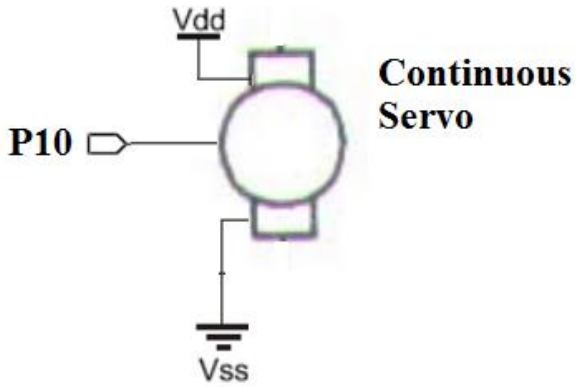

Figure 10: Servo 1 Circuit

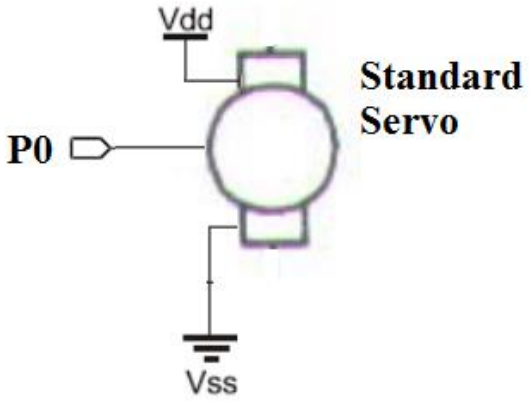

Figure 11: Servo 2 Circuit

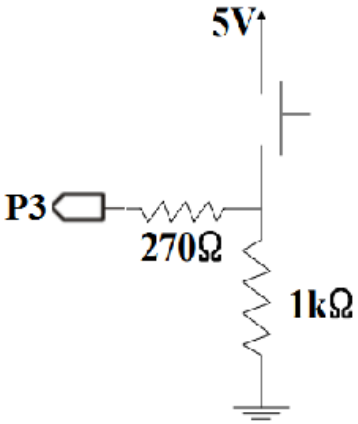

Figure 12: Limit Switch Circuit

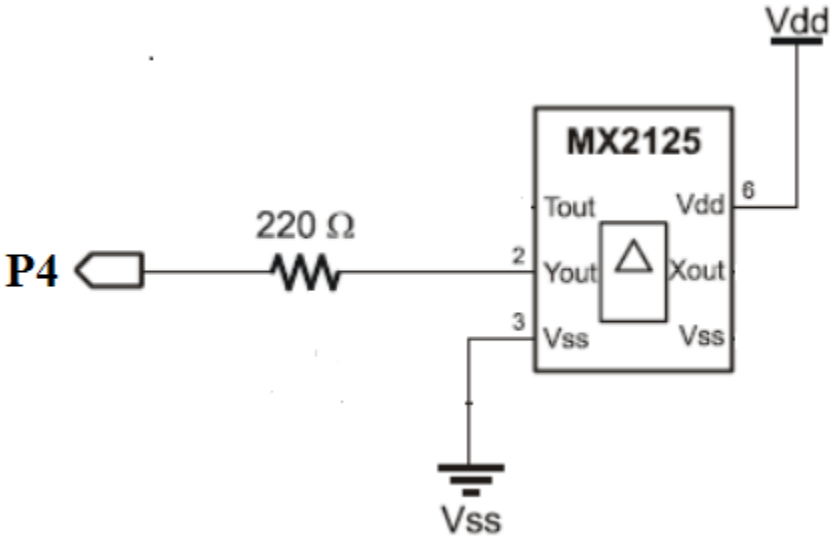

Figure 13: Accelerometer Circuit

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

'Print Screen
cou VAR Byte         'counting variable
FOR cou=1 TO 6
 DEBUG " | | | | | | |",CR
NEXT
DEBUG " ---------------"

'Initialize pins
OUTPUT 0      'feeder servo, holds peices
OUTPUT 1      'feeder servo, determines column
INPUT 3       'button for player 1
INPUT 4       'accelerometer for player 1
INPUT 6       'button for player 2
INPUT 7       'accelerometer for player 2


'initialize variables
player VAR Bit      'current players move
board VAR Nib(42)   'status of the board
cols VAR Nib(7)     'number of pieces in each column
win VAR Bit         'true when someone wins
cou1 VAR Byte       'second counting variable
lastmove VAR Nib    'stores column last played in
nextmove VAR Nib    'move made next
variable VAR Byte   'extra variable

FOR cou=0 TO 41     'initialize board
 board(cou)=0
next

'main loop
Main:



GOTO move
GOTO check
GOTO full
player=player^1   'changes player
GOTO Main


'player 1s move
move1:


return


'computer places piece
place:



return
```

```
'move has been selected
move:
 cols(lastmove)=cols(lastmove)+1
 IF(player=0) THEN              'If player 1's turn
  DEBUG CRSRXY, 2*lastmove, 6-cols(lastmove), "X"    'Places X for player 1
  board(lastmove+(cols(lastmove)-1)*7)=1              'stores player 1's move
  RETURN
 ENDIF
  'If player 2's turn
 DEBUG CRSRXY, 2*lastmove, 6-cols(lastmove), "O"    'Places X for player 1
 board(lastmove+cols(lastmove)*7-7)=2               'stores player 1's move
 RETURN

'checks if all spaces are full to end game
full:
 cou1=0
 FOR cou=0 TO 41
  IF(board(cou)!=0) THEN cou1=cou1+1
 NEXT
 IF cou1=42 THEN finish
RETURN

'check if someone has won
check:
 win=1
 'vertical scan
 FOR cou=0 TO 6      'horizantal scanning
  FOR cou1=0 TO 2    'vertical scanning
   variable=board(cou1*7+cou)
   IF
(variable=board(cou1*7+cou+7)&variable=board(cou1*7+cou+14)&variable=board(cou1*7+cou+21)&variable
!= 0) THEN finish
  NEXT
 NEXT

 'horizontal scan
 FOR cou=0 TO 3      'horizantal scanning
  FOR cou1=0 TO 6    'vertical scanning
   variable=board(cou1*7+cou)
   IF (variable=board(cou1*7+cou+1)&variable=board(cou1*7+cou+2)&variable=board(cou1*7+cou+3)&variable
!= 0) THEN finish
  NEXT
 NEXT

 'right&up scan
 FOR cou=0 TO 3      'horizantal scanning
  FOR cou1=0 TO 2    'vertical scanning
   variable=board(cou1*7+cou)
   IF
(variable=board(cou1*7+cou+8)&variable=board(cou1*7+cou+16)&variable=board(cou1*7+cou+24)&variable
!= 0) THEN finish
  NEXT
 NEXT

 'down&right scan
 FOR cou=0 TO 3      'horizantal scanning
  FOR cou1=3 TO 5    'vertical scanning
   variable=board(cou1*7+cou)
```

```
   IF (variable=board(cou1*7+cou-6)&variable=board(cou1*7+cou-12)&variable=board(cou1*7+cou-
18)&variable != 0) THEN finish
  NEXT
 NEXT
 win=0
RETURN     'end of check


'end game
finish:
 IF win=0 THEN
  DEBUG CRSRXY,0,9, "DRAW"
  end
 endif
 IF(player=0) then
  DEBUG CRSRXY,0,9, "PLAYER 1 WINS"
  END
 ENDIF
 IF(player=1) THEN
  DEBUG CRSRXY,0,9, "PLAYER 2 WINS"
  END
 ENDIF
end
```