# Machine- Human Tic-Tac game based on Microcontroller Technology

Alauddin Al-Omary
College of Information Technology
University of Bahrain
Al-Sukhir, Bahrain
Email: aalomary {at} uob.edu.bh

*Abstract:* **In this paper the hardware implementation of intelligent Tic-Tac toy is presented. The implementation uses Graphical LCD (GLCD) touch screen and microcontroller. The microcontroller receives the player move from GLCD (displayed as X) and uses intelligent algorithm to analyze the move and choose the best counter move. The microcontroller displays the counter move on the screen as circle (O). The algorithm decides the winner when game is finished according to the Tic-Tac playing rule. The system is implemented using cheap available off-the-shelf electronic components and tested and proved to be working fast and efficiently.**

***Keywords:*** ***Tic-Tac Game, AI, Microcontroller, Zero-sum assumption algorithm***

## I. INTRODUCTION

Games provide a real source of enjoyment in daily life. Games also are helpful in improving the physical and mental health of human. Apart from daily life physical games, people also play computer games. These games are different than those of physical games in a sense that they do not involve much physical activity rather mental and emotional activities.

### I.I TYPES OF GAME

Perfect Information Game: In which player knows all the possible moves of himself and opponent and their results e.g. Chess, Tic-Tac, etc.

Imperfect Information Game: In which player does not know all the possible moves of the opponent e.g. Bridge since all the cards are not visible to player.

### I.II TIC-TAC GAME

Among many game available, the Tic-Tac toy seems to be popular since it has very simple rules and can be played by kids as well as adults.

Tic-tac game is a pencil-and-paper game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The X player usually goes first. The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game. In this paper the X player is a Man and the O player is the machine.

### I.III RELATED WORK

Many software implementations of Tic Tac game had been reported and recently it became available for smart phone such as the one for Apple iPhone [1], and the other for Android environment [2]. However few hardware implementations were reported.

In Cornell University [3] hardware based game system with touch screen interfaces based on Atmel ATmega644 microcontroller was reported. The system implements Tic-Tac game and other two games.

A system where two players can play Tic Tac Toe with each other using their respective PC by the help of microcontroller is introduced in [4]. The objectives of the work includes developing an algorithm for Tic Tac Toe, interfacing hardware with the two PCs and to design a real world setup for Tic- Tac toe game using LED array / robotic arms. The algorithm proposed in this paper is fuzzy based and tested for the game. The researchers in [5] took advantage of earcons [6] fundamental characteristics, such as spatialization usually employed for concurrent/parallel reproduction, in order to implement a tic-tac-toe audio game prototype. The proposed sonic design is transparently integrated with a user control/interaction mechanism that can be easily implemented in mobile devices incorporating movement sensors (i.e. accelerometers and gyroscope).

In [7] a design of a parallel digital circuit that performs neural network (NN) calculations to evaluate Tic-Tac-Toe position was introduced. FPGA's are programmed to implement custom digital designs by physically mapping paths between the logic gates on each device. Using an FPGA allows the structure of the NN to be reprogrammed without any monetary cost. The author claims that NN implementation has better performance than traditional software implementations, because it takes advantage of the NN's inherent parallel structure.

Another NN application is implemented by [8] extend the game by adding two additional rows, two additional columns, and has been extended to the 3rd dimension. The paper calculate the optimum position using  the idea of creating a neural network that uses backpropagation coupled with elements of a genetic algorithm to improve the likelihood that the most optimal solution is obtained and outline our methodology at the implementation level.

However, some these implementations are expensive and some are slow and lacking the speed required by real time game response. Also getting the optimum counter move is another issue in some of these hardware implementations.

### I.IV SCOPE OF THE WORK

People generally play computer games by using the common input devices like keyboard, mouse and joystick. A real sensation is not always achieved by playing these games with these traditional devices. This is due to the fact that the buttons on the keyboard and joystick do not truly reflect the mapping between game elements and their directional movements.

In this paper the implementation of portable player Tic-Tac game machine is presented. In order to make a real sensation for player who plays the game a GLCD is used. The GLCD serves as input and output for the game. The machine is designed to be portable inexpensive and fast. To make a cheap machine, the presented Tic-Tac game uses off-the shelf cheap components such as PIC microcontroller and GLCD. The microcontroller is programmed using intelligent algorithm that implements the Tic-Tac game rules and responds to the player move fast.

There are different options for tic-tac-toe game. We can create either two human players game or AI (micro-controller) verses human player. We realized that creating AI verses human will be a challenging one, and we kept in mind AI part as the goal of our work.

## II. Game Theory

Games [9], [10], [11] are represented in the form of trees wherein nodes represent all the possible states of a game and edges represent moves between them. Initial state of the game is represented by root and terminal states by leaves of the tree. In a normal search problem, the optimal solution would be a sequence of moves leading to a goal state that is a win. Even for a simple game like tic-tac-toe is too complex to draw the entire game tree

### II.I Definition

Game playing is a search problem defined by following components [9]:

Initial state: This defines initial configuration of the game and identifies first payer to move.

Successor function: This identifies which are the possible states that can be achieved from the current state. This function returns a list of (move, state) pairs, each indicating a legal move and the resulting state.

Goal test: Which checks whether a given state is a goal state or not. States where the game ends are called as terminal states.

Path cost / utility / payoff function: This gives a numeric value for the terminal states. In chess, the outcome is winning, losing or draw, with values +1, -1, or 0. Some games have wider range of possible outcomes.

### II.II The Tic-Tac game Characteristics [11]

1- It is two players game
2- It is deterministic game which means that when you provide it a specific set of inputs you will get the exact same set of outputs [9].
3- It is a perfect information game where perfect information is available for all move [10].
4- It is tree based game
5- It is a zero sum game where the interests of the players are dramatically opposed.

## III. Hardware Components

As shown in figure (1), the system consists from the following components:

a) PIC microcontroller: it provides the intelligence to the game so it can interact with the user, by displaying crosses and circles and detecting if the game over or not.

b) GLCD: it is the game interface where the circles and crosses are displayed.

c) Touch Screen: it is located on the top of the GLCD, so the user will be able to interact with system and crosses will be displayed on the square he touches.

d) Power Supply: supplies the system with the required voltage.

e) Buzzer and LED: it alerts the user whether the game is started or it is over.

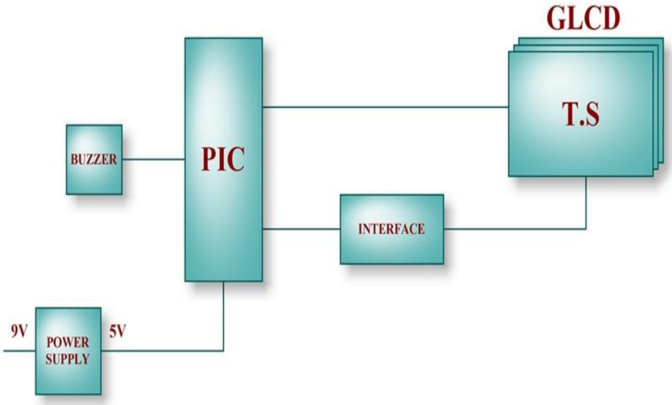

Figure (1) Tic-Tac Toy components

### III.I The GLCD

The GLCD is used to input the player move and display the counter move decided by the microcontroller. According to the Tic-Tac rules, the GLCD is divided into three rows and three columns to make (3x3) 9 squares as shown in figure (2). Each square represents one touch panel. The touch panel will be fixed on front of the graphical LCD, so when pressing on the touch panel the corresponding button on the GLCD will be active.

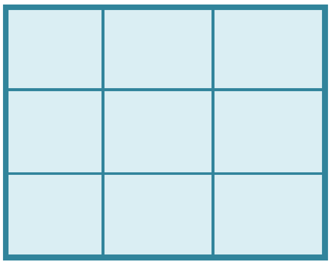

Figure (2) GLCD Division

When the player touch a particular touch panel, a cross (X) will be drawn on that part of the GLCD. The intelligent algorithm that resides inside the microcontroller will analyze the player input and decides the counter move and this move will be displayed as a circle (O) on the GLCD.

### III.II Touch Screen

Touch screen technology can be used as an alternative user interface with many applications and is used in this work to give real sensation to the Tic-Tac game. A Touch screen is a display which can detect the presence and location of a touch within the display area. The term generally refers to touch or contact to the display of the device by a finger, light pen or stylus.

The touch panel that we used is shown in the figure (3), this panel are added to a graphical LCD to display the input and the output of the game.

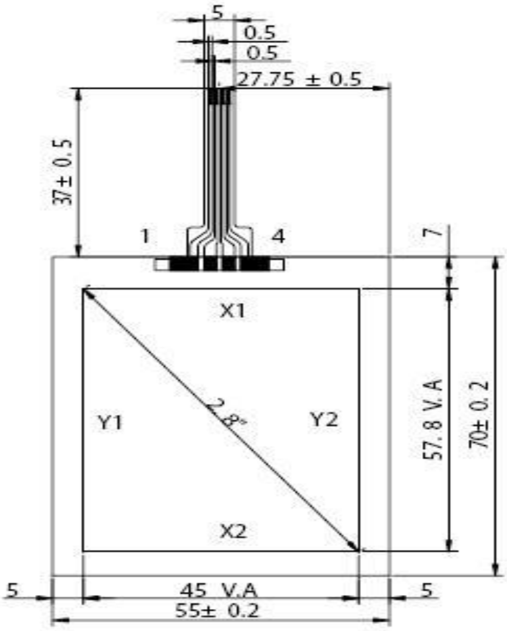

Figure (3) The touch panel

It has two analog dimensions X and Y their voltages are from (0-5)V (Volt) with tolerance and it has four pins X1, Y1, X2, Y2.

To measure a point using this coordinates X and Y:

At first if the X axis is what is wanted to be measured then:

Y1= Z state and the voltmeter is connected to it, X2 = 0V, Y2= 0V and the X1 = 5V.

If the Y what is wanted to be measured then:

X1= Z state and the voltmeter is connected to it, Y1=5V, X2= Z state and Y2 is equal to zero volt .These values for the pins is given by the PIC microcontroller and thus we have the values for a point on the touch panel.

When writing on the touch panel it is considered as a voltage change which is will be saved in a two registers called X and Y which is the two dimension of the touch panel .

### III.III PIC Microcontroller

PIC (Peripheral Interface Controller) microcontroller [14] is the intelligence provider for the system, that it will detect if the user touched a square so it will display the cross on that square, then it will detect if there a line made by three crosses if not it will determine and do calculations to choose a square where to display the circle, after displaying the circle PIC will detect if there a line drawn by the circles. All these steps will be repeated until a line is drawn by crosses or circles, or until there is no more available squares. In the two cases PIC will turn the buzzer on. And will initialize the game. PIC 16F877A [15] Microcontroller was chosen from many types of Microcontroller available in the market because of the good specification it has to meet the real time requirement of the Tic-Tac toy as well as the big control program needed to implement the game. It has 8K words Flash, 368 bytes RAM, 256 EEPROM, 1-20 MHz operating frequency. It has many input and output ports which are needed to connect the GLCD screen. Other important features are it can work at industrial temperature and its performance is about 5 MIPS which mean that 200 nanosecond are needed for one instruction execution that makes the program execution speed meet the real time requirement of the game. A 4 MHz clock is used to push the microcontroller.

### III.IV Power Supply

In order to use the PIC microcontroller it is necessary to feed it with required voltage and correct clock. We use 9 volt battery to power both the whole circuit and since the microcontroller needs 5 volts, a voltage regulator is used to supply this voltage from the 9 volts battery.

### III.V buzzer Circuit and LED

The buzzer circuit and LED are used to mark the finishing of the game when there is a complete line is drawn by crosses or circles, or if there is no more available squares. In the two cases PIC will turn the buzzer on, LED on and the game will initialized.

The complete Tic-Tac game circuit is shown in figure (4). This circuit shows the power circuit, regulator, Microcontroller, GLCD, LED and buzzer.

## IV. Software design

The microcontroller is programmed to play the Tic-Tac toy with a human player. By playing games, the machine intelligence can be revealed. The Tic-Tac game is a tree based game. Tree searching will be time consuming even for a few plies. Hence, an efficient searching algorithm is an important issue. The problems are solved by forming a possible set of solutions based on the endgame condition, or searching for the set of solutions based on the current game condition. The machine cannot learn to play the games by itself. In this work an evolutionary approach was employed to evolve and to learn for playing Tic-Tac-Toe without the need of a database. The complete flowchart of the microcontroller program is shown in figure (5). An algorithm based on zero-sum assumption concept is used. The machine is first initialized and waits for player move. When player touch a box it will display X on that box. If a line is created vertically, horizontally or diagonally (V, H, D), the game is over. If not the algorithm will calculate the counter move by selecting a box (x,y) which has best position using Zero-Sum assumption algorithm [11] and then display O on that box. Again, if a line is created vertically, horizontally or diagonally (V, H, D), the game is over and if not the process is repeated and the player should make his next move. For dealing with such types of games, all the legal moves that can be made from the current position should be considered. Computing the new position resulting from each move and evaluating each resulting position and determine which is best position is necessary to make the counter move. Wait for the player to move and repeat the procedure. But for this procedure the main problem is how to evaluate the position? Evaluation function or static evaluator is used to evaluate the 'goodness' of a game position. The zero- sum assumption allows us to use a single evaluation function to describe the goodness of a position with respect to both players. Consider, f(n) is the evaluation function of the position 'n'. Then,

– $f(n) >> 0$: position n is good for machine and bad for player

– $f(n) << 0$: position n is bad for machine and good for player

– $f(n)$ near 0: position n is a neutral position

Therefore the evaluation function for Tic- Tac- Toe used is:

f( n) = [no. of 3- lengths open for machine] - [no. of 3- lengths open for player]

Where a 3- length is a complete row, column, or diagonal.

In order to code this algorithm and store it in the microcontroller MLAB editor was used [15]. There are several ways of programming the PIC microcontroller - using BASIC, C, or Assembly Language. The BASIC language is used because it is the easiest way to program the PIC [16]. Using MLAB editor, first, the program was wrote and saved as a source file (*.bas). After that, the .bas file was compiled to convert it in to assembly language (*.asm). Then by using PIC Shell it was converted to Hexadecimal (*.Hex) which is the actual machine language understood by the PIC microcontroller. Finally the *.Hex file was stored into the Microcontroller memory.

## V. Prototyping and Testing

The system prototype was developed as shown in figure (6). The components used are off-the shelf and therefore are cheap and available. By testing the prototype it was found that the system is working efficiently. The overall prototype design efficiency is assessed in terms of the employed touch screen accuracy and the system response time when is assessed in real game-play conditions. The playability achieved through the integration of the employed auditory user interface and the correctness and optimality of the counter play move done by the game.

The touch screen accuracy was perfect and there is no single error reported throughout the test. The response time of the machine was fast about 0.25 ms in average.

The written computer program based on Zero-Sum assumption algorithm works perfectly and the counter move selected by the machine is the best move. The zero-sum assumption algorithm works without any problems although it enumerates the 765 essentially different positions (the state space complexity), or the 26,830 possible games up to rotations and reflections (the game tree complexity) on this space [13].

## Conclusion

In this paper the implementation of portable player Tic-Tac game machine is presented. In order to make a real sensation for player who plays the game a GLCD is used. The GLCD serves as input and output for the game. The machine is designed to be portable inexpensive and fast. To make a cheap machine, the presented Tic-Tac game uses off-the shelf cheap components such as PIC microcontroller and GLCD. The complete system prototype was implemented and tested. The microcontroller is programmed using zero-sum assumption algorithm that implements the Tic-Tac game rules and responds to the player move. The prototype test shows that the system fulfills the game requirement in term of playability and fast response time which is about (0.25ms) on average.

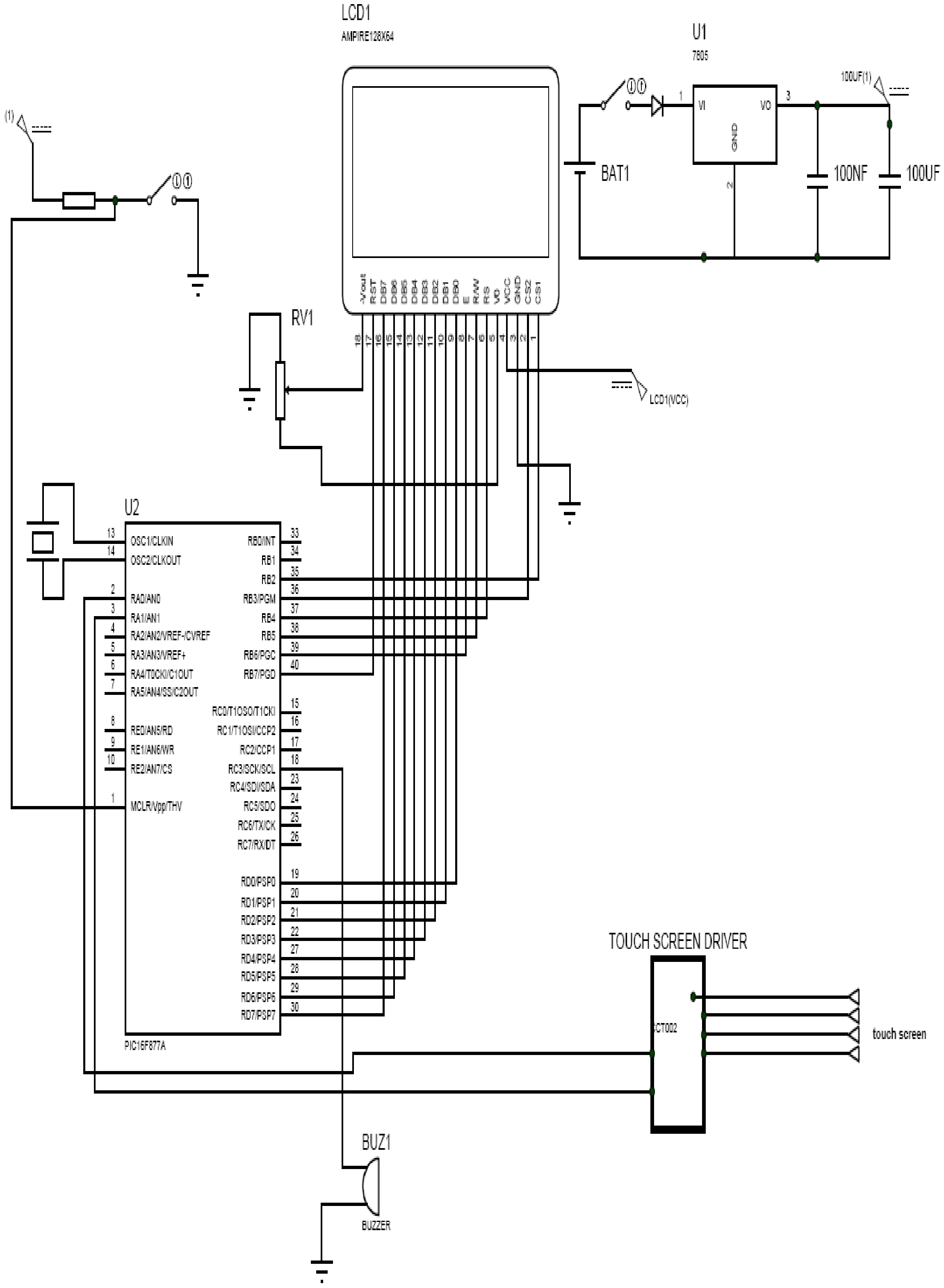

Figure (4) The complete Tic-Tac Circuit

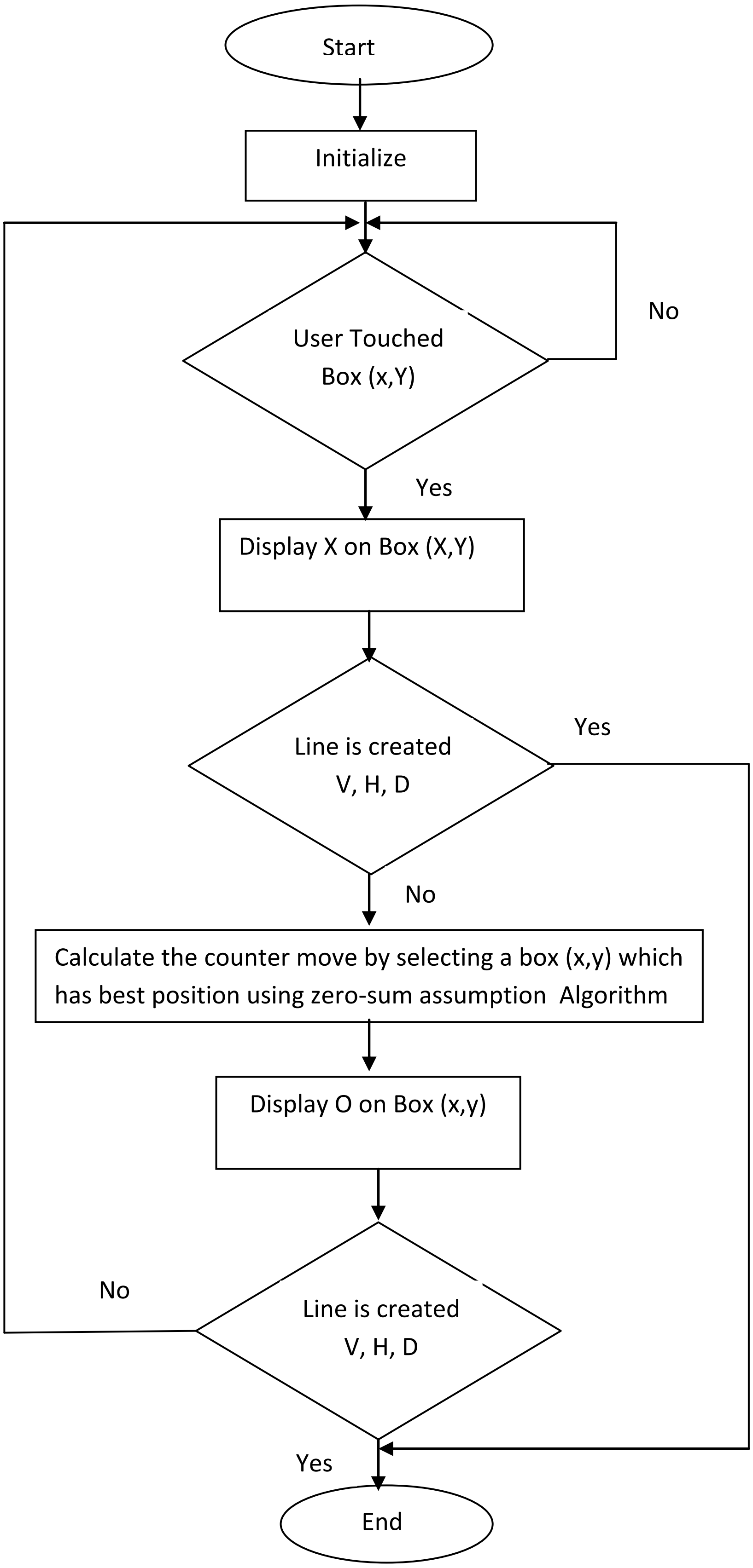
Start
Initialize
User Touched
Box (x,Y)
No
Yes
Display X on Box (X,Y)
Line is created
V, H, D
Yes
No
Calculate the counter move by selecting a box (x,y) which has best position using zero-sum assumption  Algorithm
Display O on Box (x,y)
No
Line is created
V, H, D
Yes
End

Figure (5) The flowchart of the microcontroller program

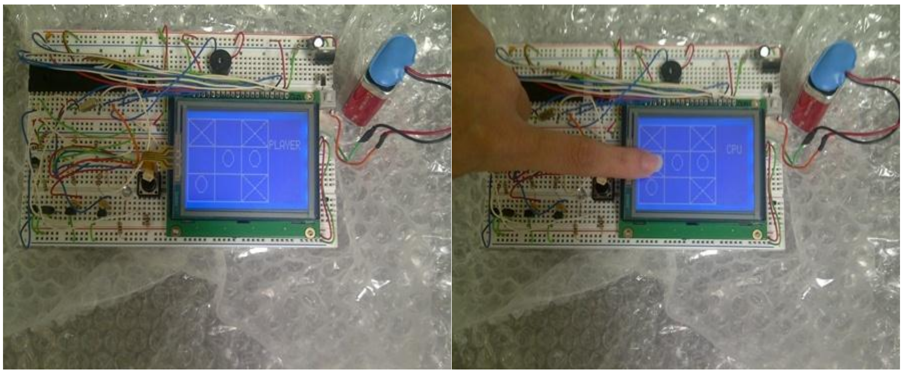

Figure (6) prototype implementation of Tic Tac toy

## REFERENCES

[1] iPhone Simple Tic-Tac-Toe Implementation, Retrieved 11/9/2012 from:

http://www.vworker.com/RentACoder/misc/BidRequests/ShowBidRequest.asp?lngBidRequestId=1622905

[2]Tic Tac for Android, Retrieved 1/9/2012 from:

http://www.androidapps-home.com/tic-tac-toe-free-android-57.html

[3] Benjamin Harris and Philip Bernard, "A TFT LCD with resistive touch screen powered by an ATmega644 with touch-focused minigames." Cornell University ECE 4760 - Final Project, Retrieved 21/8/2012 from: http://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/f2009/nic4_sck76/nic4_sck76/index.html

[4] Ashutosh Kumar Sahu, Parthasarathi Palita*, Anupam Mohanty, " TIC TAC TOE GAME BETWEEN COMPUTERS: A COMPUTATIONAL INTELIGENCE APPROACH", Retrieved 20/7/2012 from: http://ficta.in/attachments/article/55/21%20TICTACTOE_.pdf

[5] Andreas Floros,Nicolas-Alexander, Tatlas Stylianos Potirakis, "Sonic perceptual crossings: a tic-tac-toe audio game", Proceedings of the 6th Audio Mostly Conference: A Conference on Interaction with Sound ACM New York, NY, pp. 88-94, USA 2011.

[6] Thomas Hermann, Andy Hunt, John G. Neuhof, "The Sonification handbook", Logos Publishing House, Berlin, Germany, 2011.

[7] Stephen Mann and Matthew Netsch, "A parallel Embedded Neural Network for an Intelligent Turn-Based Engine", Retrieved 11/8/2012 from: http://www.samduffysinger.pwp.blueyonder.co.uk/Project%20Outline%20TicTacToe.pdf

[8] Shahzeb Siddiqui , Francis Mutuc and Nicholas Schmidt, " Designing a 5x5x5 Tic-Tac-Toe Game using a Neural Network with Backpropagation with a Twist", Retrieved 11/9/2008 from:

http://www.personal.psu.edu/fcm5007/eportfolio/AINeuralNetworks.pdf

[9] Fudenberg, D. and Tirole, J. (1993) Game Theory, MIT Press.

[10] Albert, Michael H.; Nowakowski, Richard J.; Wolfe, David (2007). Lessons in Play: In Introduction to Combinatorial Game Theory. A K Peters Ltd. ISBN 978-1-56881-277-9.

[11] Cameron Browne, Frédéric Maire: Evolutionary Game Design. IEEE Trans. Comput. Intellig. and AI in Games 2(1): 1-16 (2010)

[12] Beck, József (2008). Combinatorial games: tic-tac-toe theory. Cambridge University Press. ISBN 978-0-521-46100-9.

[13] Prajit K. Dutta (1999): Strategies andGames: Theory and Practice, MIT Press.

[14] Programming PIC MCUs in BASIC. Retrieved 11/9/2011 from:

http://www.mikroe.com/en/books/picbasicbook/01.htm

[15] PIC programming tools. Retrieved 1/6/2011 from:

http://www.geocities.com/nozomsite/pic1.htm

[16] PIC microcontroller tutorial. Retrieved 8/7/2012 from: http://www.voti.nl/picfaq/index.html