Multi-entertainment Tic Tac Toe Game

By
Jiacheng Zhu
Minkang Yang
Nuochen Lyu

Final Project For ECE 445, Senior Design, Fall 2017
TA: Xinrui Zhu

13 Dec 2017
Project No.30

Catalog

1. Introduction

1.1 Background

Game is always a significant activity throughout the history of human being. Although it seems like it is sometimes purely for entertainment, games offer elements that are crucial for human development, both physically and mentally. The use of games includes self-improvement, encouraging positive lifestyle changes, and increasing motivation to complete work objectives. Therefore, we want to make a tic tac toe game that is fun to play together. For instance, when customers in the restaurant are waiting for dining, rather than checking their phones, a small game box can help customers get in the mood.

We are three Computer Engineering Students, and we hope through this project, we could learn how to make a portable game device. We would like to make the tic tac toe game box a high completeness, good-looking, and captivating product for public use.

1.2 Overview

In this semester we made a game box that is able to support tic-tac-toe game. User should able to press LEDs to place pieces and the LEDs can display various patterns. This project includes circuit design, hardware, mechanic structures and software. The final function and performance are similar to the requirement set in our proposal, and we successfully implement all the features in the final dema. There are some change and update in our PCB design and components models such as LM385 comparator and 3 to 4 encoder. Overall this project is completed and beyond our expectation. The final paper will introduce the hardware and software modules design in this game box. The detailed requirements for the functionality will be introduced.

1.3 Objective

1. Human could play tic tac toe with another human.
2. Human could play tic tac toe with AI.
3. LED blocks must support animation patterns “X”/“O”

2. Design

2.1 Overview

The project is majorly composed of four sections: input, display, control, and power.The power section will deliver the corresponding voltage to every electronic element.

Display section ‘s light will be determined by the control section, which will continuously receive the input from the input section that receiving pressure input.

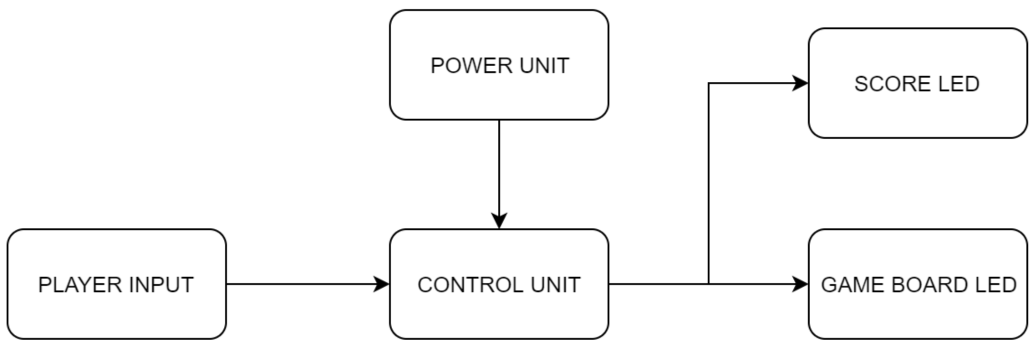


Fig. 1. Block Diagram

2.2 physical diagram

The final project would be similar to the below picture. The 3*3 LED tic-tac-toe board, 1*4 LED display on the top and two buttons on the right. The cube box would be made of wood or plastic, and the hardware design is concealed inside. Underneath the 3*3 LED board lies nine pressure sensors. The power supply is a battery that is plugged into the bottom of the cube box

The Fig 2 is our draft design. Fig. 3 is the real game box we have finally. The gamebox is made by Mr. Adrich from the machine shop according to our demand. We add some space and gap between each buttons so that wires and cables are able to settle.

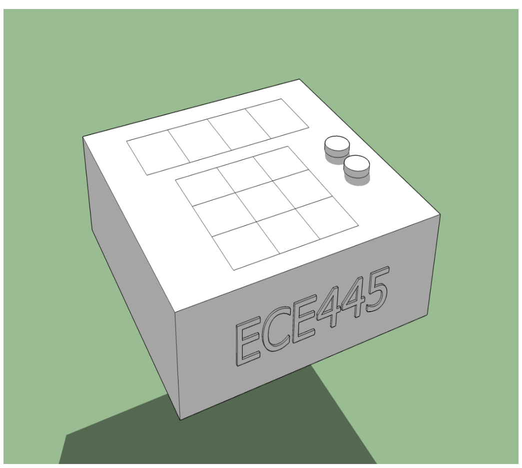


Fig. 2. Physical design sketch

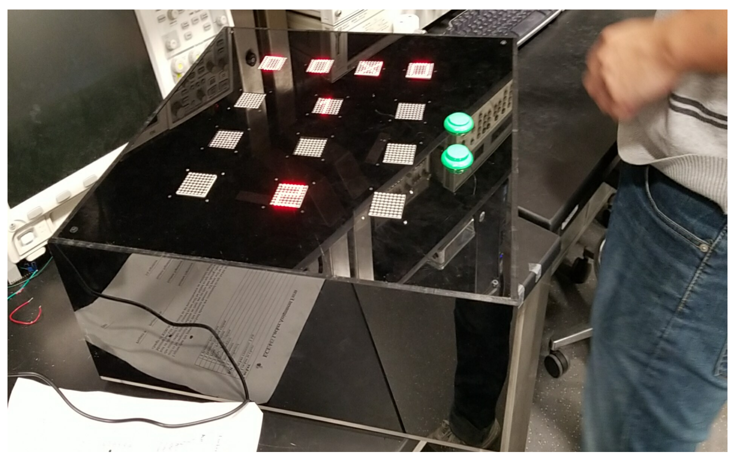


Fig. 3. Physical design real design

2.2 General Design Procedure

2.2.1 Control Unit

The control unit should able to receive correct input from play input, and give output to LED in a short period of time. We decide to use a microcontroller to be the control unit. Using gate - level RTL control unit is also possible, however since the control unit is required to deliver different patterns and make AI decisions. An RTL control unit will be hard to debug and along with very repetitive design. Using microcontroller is debugging friendly and easy to change.

Performance requirement: make decisions fast enough to response to human player.

2.2.2 Power Unit

We should use a very stable power source, both Battery - based, and plug-in based power supply are feasible to use. Due to safety and environment concern, we decide to use a plug-in power supply. Since Battery-based supply may cause severe fire danger in low - air pressure area.

Performance requirement: deliver stable 5v with no security concern.

2.2.3 Player Input

Our design is a board game; it is crucial for the game board to able to interact with player. Some options are: Making real chess and sense the change of chess position using computer vision, also making real chess but putting magnetic sensor underneath, or making press sensor underneath the game board, and sense player's pressure. The final decision is to use pressure sensor for it can bring stronger interaction between the game board and player.
Performance requirement: have tolerance range for human force touch(0-25lb). Respond need to be fast(< 0.1 s) and precise.

2.2.4 LED Display

Since our LED block is required to display pattern, we could use high resolution or single pixel based LED. The final decision is in between; we use a single pixel -based led matrix with serial input functionality. Meaning that it can display various pattern and maintain relatively clear output. Performance requirement: the current amount should be smaller than 0.1 amp. Luminous flux should around 550.

2.2.5 Design Tools

We use proteus 8.5 Design suite as our design Tools because of its versatility on simulation, full element library. and expertise in PCB design.
Performance requirement: the design tool proteus should able to generate admissive PCB layout.

2.3 Detail design procedure

2.3.1 Pressure sensor

SEN-09375 is selected for several reasons: 1. the acceptable force range from 10-10000g and the corresponding force range from 100K OHM to 0.5KOHM , the following graph shows the relationship:

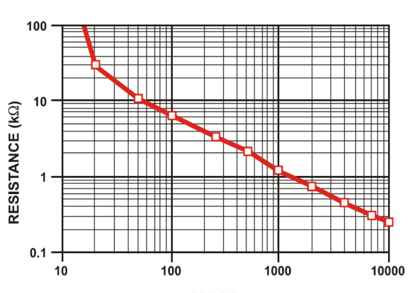


Fig. 4 FSR force resistance relation[3]

Fig. 5 FSR[4]

2. Since the whole resistor is covered in plastic, there is no short -circuit concern if it is embedded into the physical design.3. it has a large sensor area which perfectly suits our needs.

2.3.2 Pressure sensor circuit integration

Input: VCC, GND

output: change with threshold by the pressure sensor

By using an op-amp, we can convert an analog signal generated from the pressure to a digital signal that set up a threshold that can output digital 0 or 1 depending on the force applied. The circuit is Fig 6.

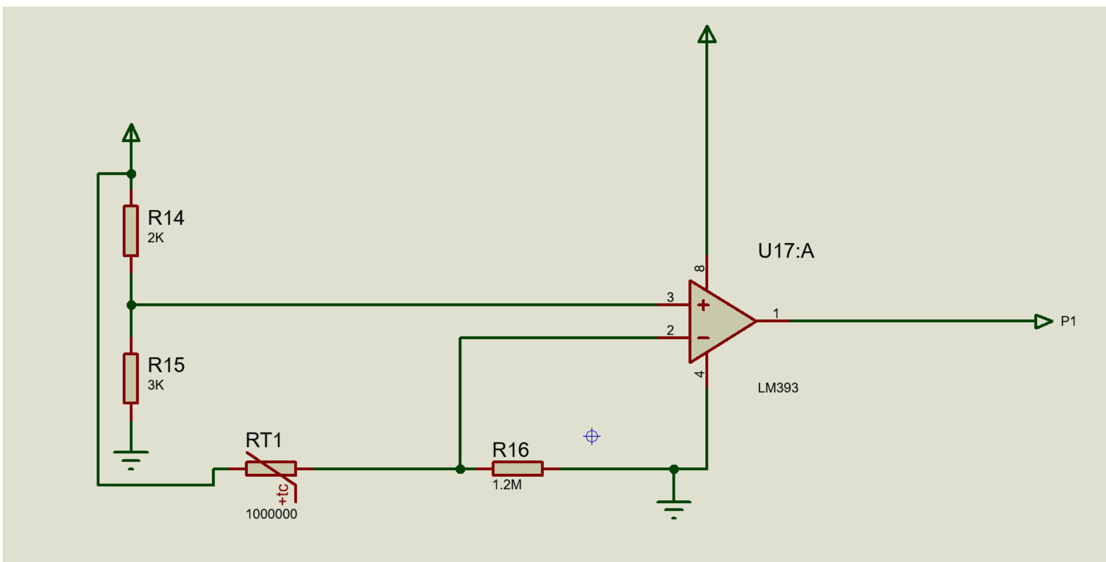


Fig. 6 A/D Circuit Schematic

RFSR = resistance of the force sensing Resistor

RP = resistance of the paring Resistor that is in serial with FSR(75K ohm).

VOUT = (V+) * [RFSR / (RFSR + RM)].

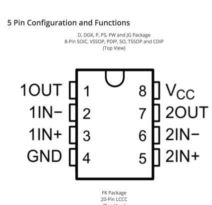


Fig. 7 Op-amp pin

P1 will output 0 or 1 , depend on the value of RT1, Op-amp specification is below.

Component: R16:1.2M, R14:2K, R15:3K

Force Sensing Resistors (FSR), is a polymer thick film (PTF) device which exhibits a decrease in resistance with an increase in the force applied to the active surface. Shown in Fig 5 on the left. The standstill default resistance for FSR is 1M Ω. The force sensing resistor we use is made by Pololu with Part Number 1696, UNSPSC Code 32121600.
The pressure sensor circuit is shown in Fig 5. A 5V DC power would be provided to all Vcc in fig.5. The FSR(RT1) is connected with a 1.2M resistor in seriy. With such 1.2M resistor the voltage change across RT1(1M Ω) would be able to differentiated. The voltage change should

be varied between 0.1 to 4.9 V. A op amp is connected to convert the changing analog signal from RT1 to digital signal. The op amp we use is made by Texas Instrument, with model number LM393. The voltage across RT1 will feed into the negative input of the op amp(pin 2). We use R14(2KΩ) and R15(3KΩ) to construct a voltage divider circuit and feed 3 V to the op amp’s positive input pin.
With the circuit we described above, if there is no pressure on the force densing resistor, the voltage feed to pin2 would be bigger than 3V, so the op amp will output 0V to our encoder, and 0V will be taken as digit 0. If there is enough pressure(can be easily done with one finger) on the force sensing resistor, the voltage feed to pin2 will be smaller than 3V, so op amp will output 5V(the Vcc to op amp is 5V) to our encoder, and 5V will be taken as digit 1.

2.3.3 LED matrix

Input : CLK, VCC, CLK,GND,CS
Output: none

We choose an 8x8 led matrix controlled by MAX7219 IC, which looks like Fig 8 and Fig 9 because it only requires four pins to control the whole 8*8 led matrix. Moreover, it can be toggled on or off without interrupt the pattern. Most importantly, all input ports meets Hold Time and Setup Time requirement.

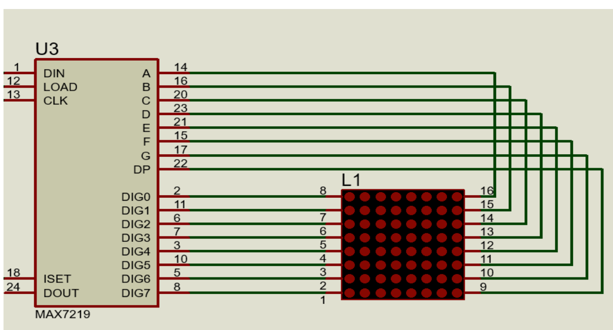


Fig. 8 LED Matrix Module

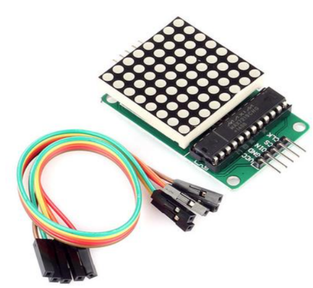


Fig. 9 LED Physical Picture[5]

The material is FR4 + electronic components; a single module can drive one 8*8 common cathode dot matrix, the working voltage of the The whole module will only receive three input instead of 16D will remain its pattern if LOW is received in LOAD input.

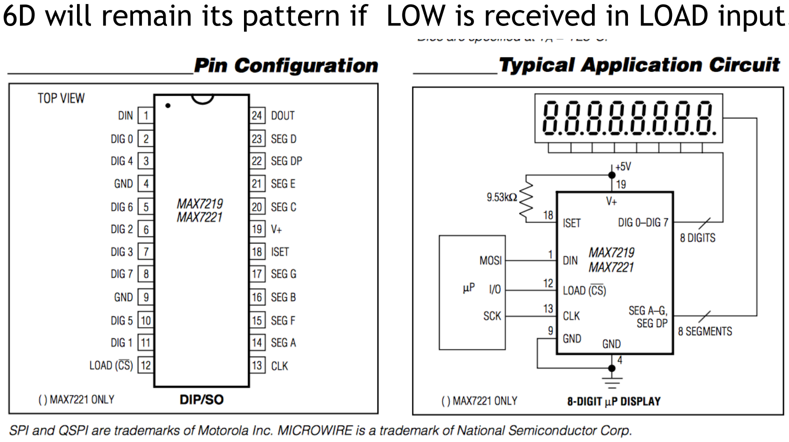
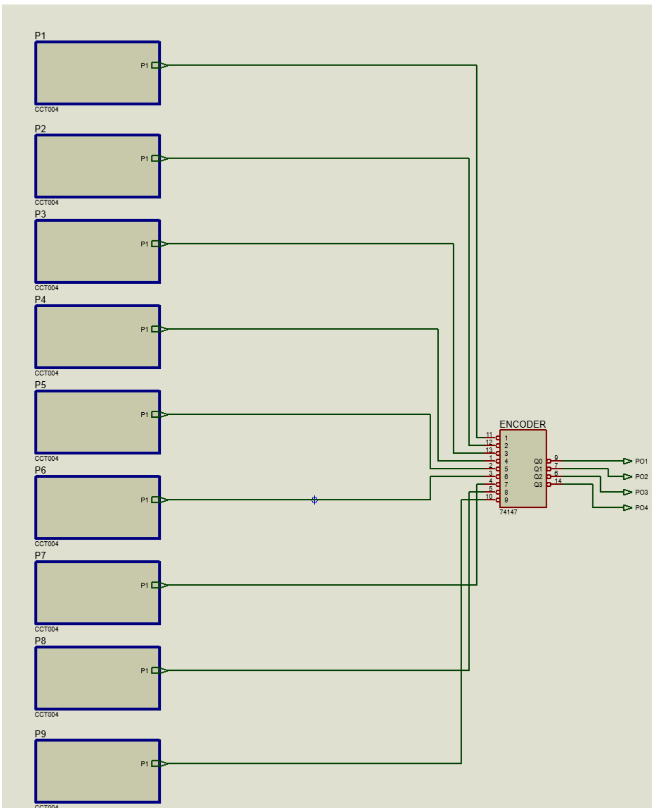


Fig. 10 MAX7219 Datasheet

The control chip is shown in Fig 10 and as long as the CS(chip select) is low, and the V+ is connected to the power, and CLK is connected to our Arduino CLK, our chip is ready to take input, DIN is for Serial-Data Input, Data is loaded into the internal 16-bit shift register on CLK's rising edge.

Fig. 11 A/D Converter Module



2.3.4 Integrate Multiple element into circuit - pressure sensor

There is nine input from OP-AMP in total, to increase port efficiency, use Encoder to encode the input op-amp signal from nine to four. The Encoder is 74HC147. We chose this encoder because we want only one pin to be active at a time. Specifically, we applied HC series since LS series doesn't fulfill the input/output requirement for ATMega328p. The encoder's input logic is in Appendix B, implementation Circuit is list in Fig 11. Each "P" module is an op-amp pressure sensor circuit since it is op-amp out0put is 0 active(when pressure senses a force), input from the Encoder is low-active, too.

2.3.5 Integrate multiple element in circuit - LED matrix

There are fourteen LEDs in total. Each contains three inputs Since there are two section: display section and LED board section. Also, many signals are repetitive or universal. It is preferred to use two separate encoder - a two to four decoder and a four to ten decoder, working logic is listed in the appendix., both will account for the chip select signal of the 8*8 LED, the first is 4 bit - 9-bit decoder, and will decide which board should receive the signal and display the tic tac toe. However in some conditions, we need to light up several sections of the board, for instance during the start the end games' graphics. In this case, since the led will remain its current lighting condition if the Vcc is not cut off, we will light each section one by one using the controller command. For the second location dispatcher, there is 2-bit input, and the decoder will select the according to LED screen to display text

Two to four decoder with four-letter display:

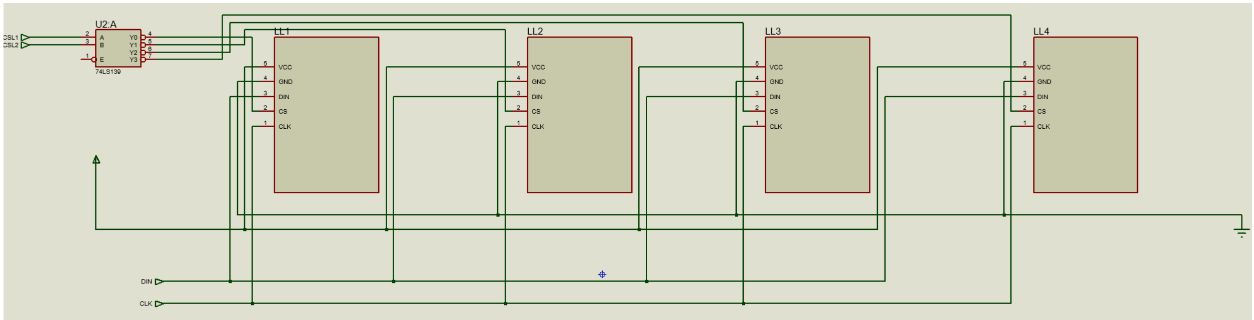


Fig. 12 LED Score Display Module

CLK,DATA pin are universal , decoder output control every CS signal.

Four to ten display with 9 game board display.

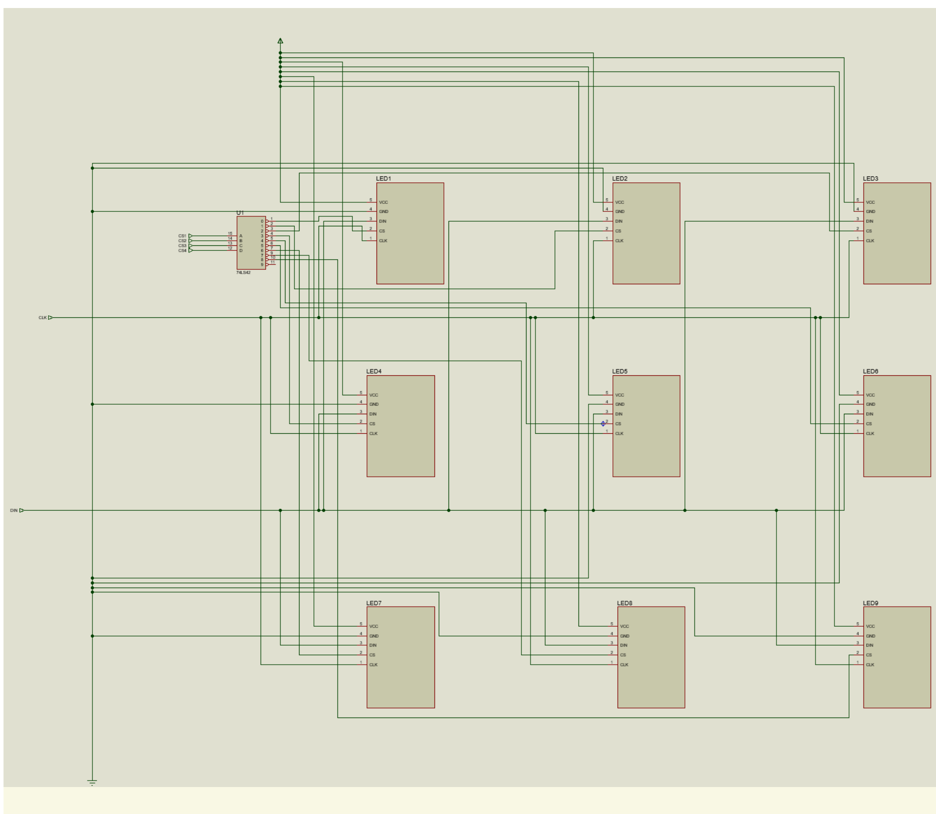


Fig. 13 LED matrix module

CLK,DATA pin are universal , decoder output control every CS signal.

2.3.6 Controller

Input :

[3:0] Encoder(pressure sensor),[1:0] Crystal, [0] Reset

Output:

[3:0]: game board display CS signal, [0] CLK, [0] DATA, [1:0] Letter display CS signal

An ATmega328P is selected for several reasons: 1. cheap, easy to access and purchase, 2. Via using Arduino development board with the controller, it is easier to debug.3. Storage space is big enough to store a tic-tac-toe decision tree. The configuration is listed in Fig 14.

The microcontroller will be the decision maker for the 9*8*8 LEDs on the board since the 8*8 LED need clock and input, the microcontroller will generate a 4 -bit chip select signal, an overall input for all the LED light, and a CLK to ensure the data-transfer. Also, it will receive the input from the 4-bit mux, which is from the pressure sensor. Following on the left are the control flow on the controller

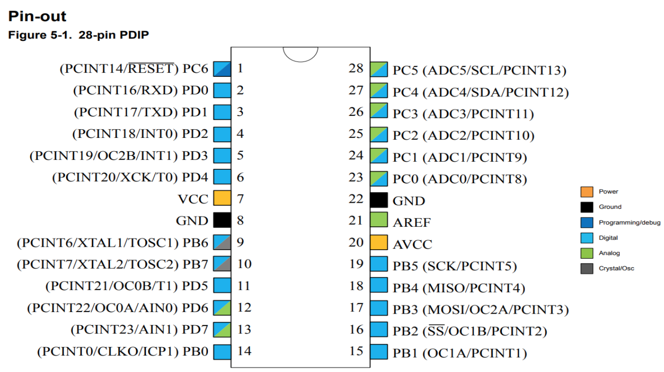


Fig. 14 Microprocessor Datasheet [2]

2.3.7 Switch button Concave plunger design

This is a 35mm arcade-game-like concave push button. With a simple screw-in design, it is perfect for mashing. This button has a great tactile, ‘clicky’ feel.

2.3.8 General Circuit diagram

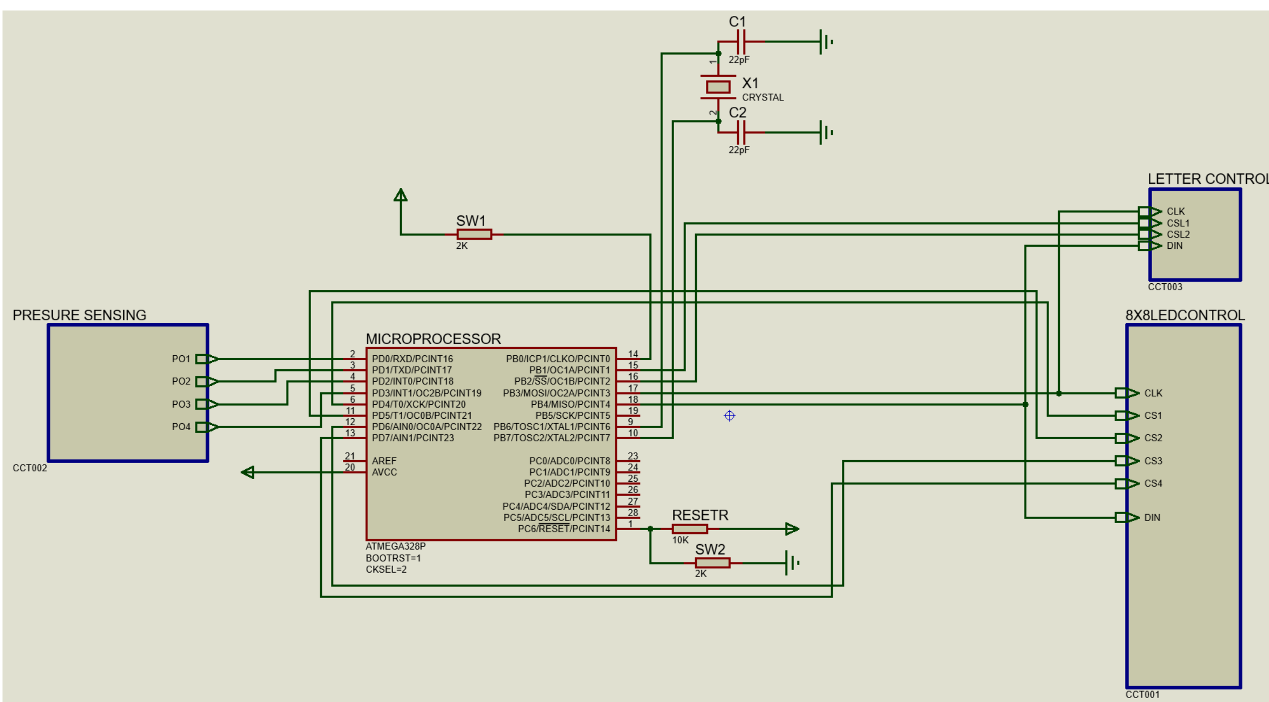


Fig. 15 Project Overview Diagram

In Fig.15, all components are connected to together. The pressure sensor module has four outputs connected to our MCU so that it knows which pressure sensor is being pressured. A crystal is appended to the MCU and the button is connected to the reset pin. 13 digital pins are connected to two LED modules not he right, one for score board and one for game board.

2.3.9 PCB Design

PCB design is composed of 2 Boards, Pressure sensor board , and LED display board.

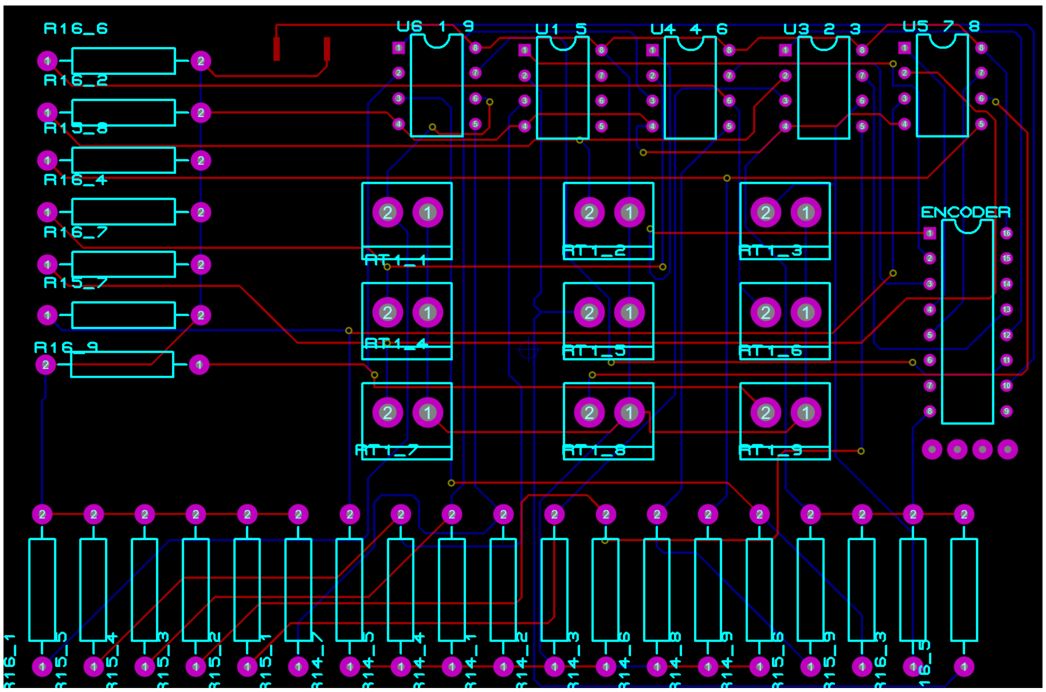


Fig. 16 Pressure Sensor PCB

In the middle there are nine pressure sensor, in alignment with 9 LED matrix in physical design.4 pins in the right are connect to the LED display matrix. The PCB is composed of 5 op-amp chip, a encoder, 9 pressure sensor, and some resistor.

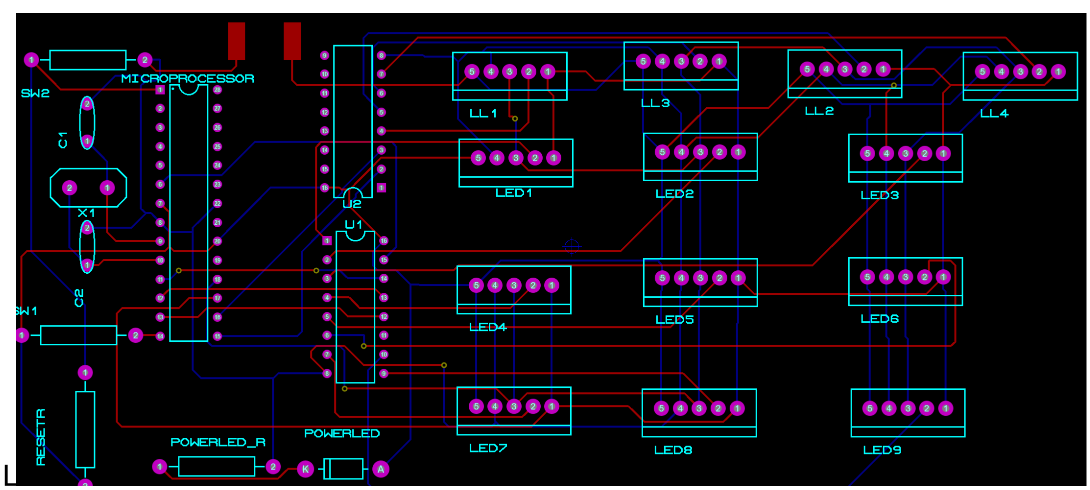


Fig. 17 LED Matrix PCB

4 LED I/O in the left top are align with 4 letter display, 9 LED I/O in the middle are align with 9 LED matrix, the PCB contains 2 decoder, 1 microprocessor, 13 LED I/O, SW, crystal, and resistors.

2.4 Software Overview

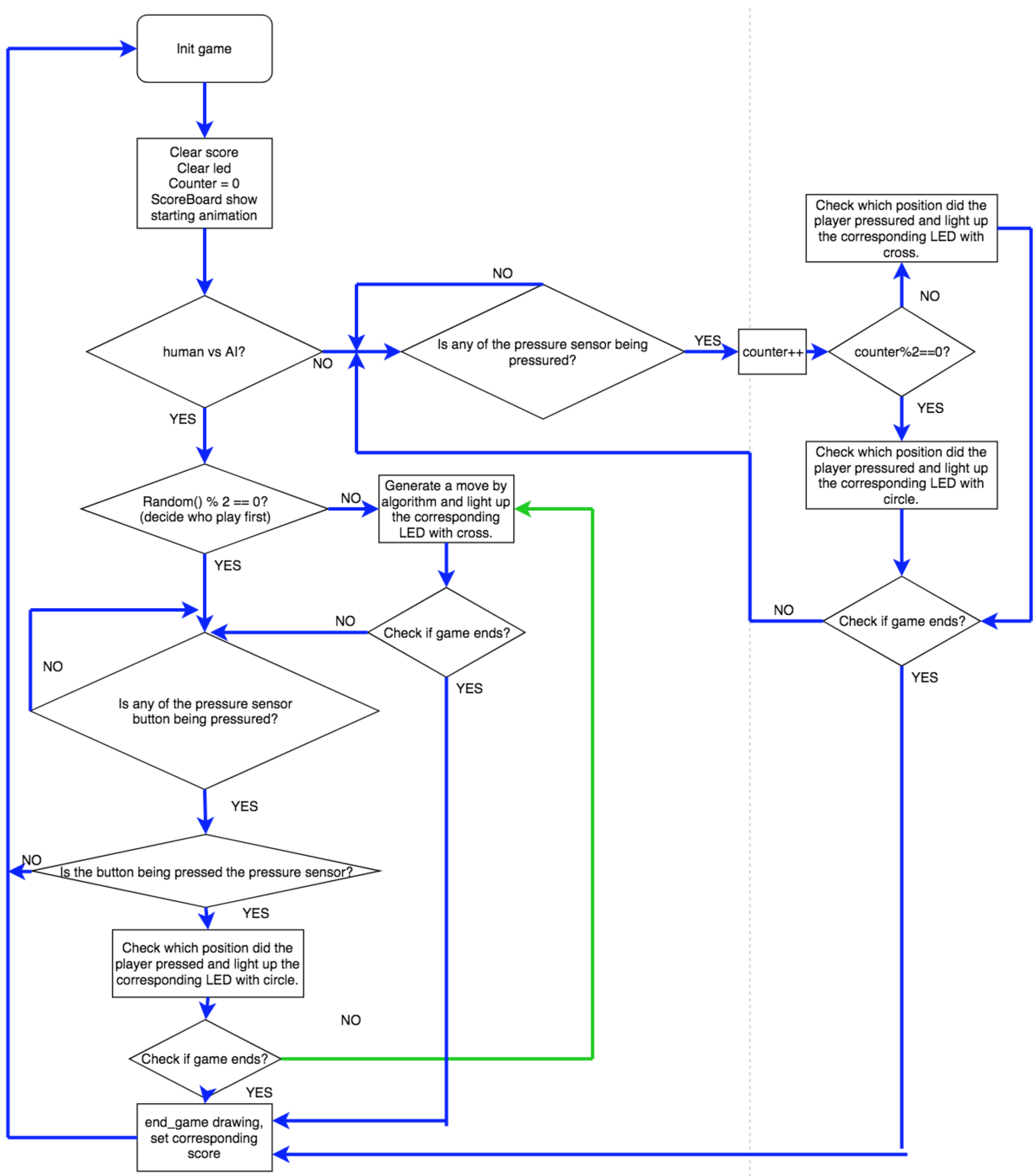


Fig. 18 Flowchart for software [11]

3 Requirements and Verification

3.1 Power Supply

Table 1. Power Supply RV

Requirements	Verification	Score
Power Supply (1) Need to output 5.0 voltage. The voltage varies within a 5% range. (2) Outputs current >= 10 mA	(1) Use multimeter to check the power source voltage. Should be 5.0 (+-5%) V (2)Use multimeter to check the current.	5

In the Lab we use a multimeter to test the Vcc of multiple components:
Power source: 5.00 V Encoder: 4.99 V LED: 4.98-5.00V Decoder 1: 4.99 V Decoder 2: 4.99 V Comparator: 4.97-5.00 V

Table 2. A/D Converter Module RV

Requirements	Verification	Score
A/D converter module (1) The A/D converter must be able to output Digital signal. Output voltage should < 1.0mV as digit 1; while the standby output voltage should be > 2.65 V as digit 0. (2) Encoder(multiplexer) should be able to read all nine inputs from A/D converters and give correct four bit output message. (3) Our A/D converter module circuit should take the little deviation of the resistance in 9dd resistors into account and correct the error.	(1) (a) Use multimeter to check the voltage for output1 pin in our comparator. Should be >= 2.65V when outputting digital 0. Should be <= 1mV when outputting digital 1. (b) Use Arduino digitalread() to check output from A/D converter. Should read 1 when we press the pressure sensor. Should read 0 when we release the pressure sensor. (2) Use Arduino to test the output from Encoder. The 4 bit output should match which pressure sensor that we pressed. Only one sensor can be pressed and recognized by the microprocessor at one time. (3) There may be some little error and deviation for our pressure sensor. We should able to compensate this error and let the output correct.	10

We built the A/D module on the breadboard. The pressure sensors did give us correct voltage level when we press/ not press the sensor. The voltage is above 3.71V when sensors are standstill. The voltage is under 0.01V when sensors are pressed. The Arduino reads four pins from the encoder, and the result is accorded to the logic table. Testing data from an encoder is shown in Table 3.

Table 3. Pressure sensor testing number reads from encoder

	Y3	Y2	Y1	Y0
Pressure sensor 1	1023	1023	1023	86
Pressure sensor 2	1023	1023	72	1023
Pressure sensor 3	963	958	0	0
Pressure sensor 4	1023	83	1023	1023
Pressure sensor 5	1023	74	1023	850
Pressure sensor 6	981	0	0	981
Pressure sensor 7	0	74	79	89
Pressure sensor 8	0	969	966	957
Pressure sensor 9	89	1023	1023	83

3.2 LED Module

Table 4. LED Module RV

Requirements	**Verification**	**Score**
LED module (1) LED matrix should be able to light up all its pixels. (2) LED matrix should be able to show cross pattern and circle pattern. (3) nine LED matrix should be able to show corresponding patterns collaboratively.	(1) When LED is powered. We should correctly configure the LED component (led intensity, display mode) and give enough power supply. All LED pixel should lighted up if correctly been initialized. (2) We would use Arduino to give data to LED’s Vin. We would let it display either “X” or “O”. “X” or “O” should be displayed on the LED. (3) We should able to let which one of the nine LED module to display and what picture pattern is been display on this LED. We would use Arduino to give signal to our LED module. The LED module should able to understand the signal from Arduino and behave correctly.	10

LED score display board (1) LED matrix should be able to light up all its pixels. (2) LED matrix should be able to show cross numbers. (3) 4 LED matrix should be able to show corresponding patterns collaboratively.	(1) When LED is powered. We should correctly configure the LED component (led intensity, display mode) and give enough power supply. All LED pixel should lighted up if correctly been initialized. (2) We would use Arduino to give data to LED's Vin. We would let it display number zero to nine. (3)We should able to decide which one of the four LED module to display and what picture pattern is been display on this LED. We use Arduino to give signal to LED module. The LED module should able to understand the signal and behave correctly.	5

This verification process is done by several testing procedure shown in Table 5.

Table 5. Power Supply RV

Test	Checkbox
Light up all LED	√
Display "A-Z", "0-9" in each LED blocks	√
Sequential let each LED display "X"	√
While one LED is switching patterns, the rest of LED does not change	√
Turn of all 13 LED	√

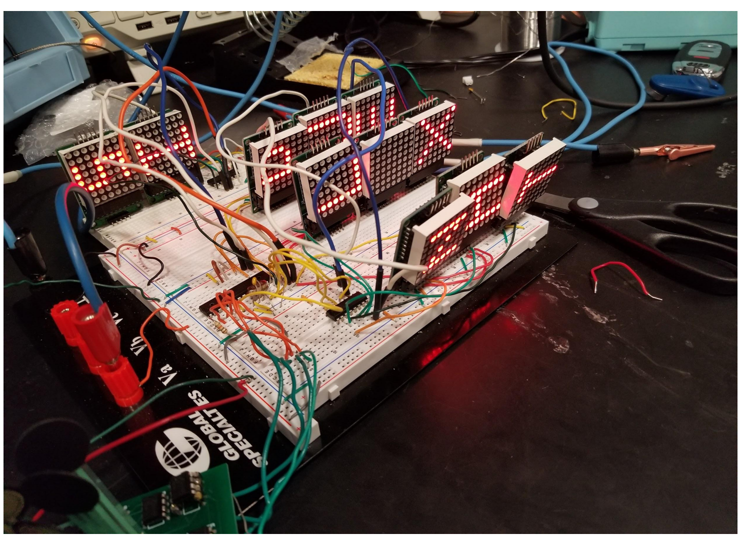


Fig. 19 LED Test Full

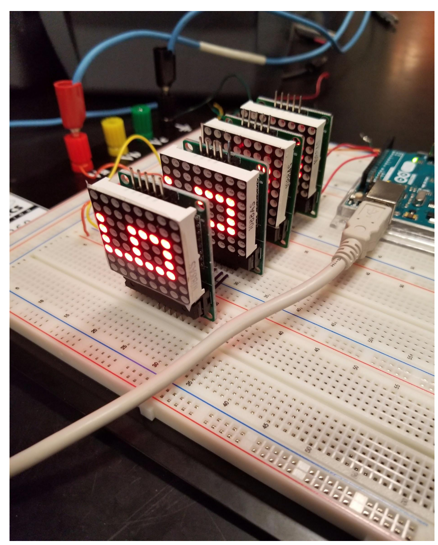


Fig. 20 LED Test Part

3.3 Microprocessor

Table 6. Microprocessor RV

Requirements	Verification	Score
Microprocessor (1) Microprocessor need to give voltage > 3.0 V as digitalwrite 1 , and voltage < 0.8 V as digitalwrite 0 (2) Microprocessor have 4 input ports; 9 output ports (3) Microprocessor should communicate with LED modules and A/D Converter module within 2s. (4) Microprocessor should handle hazard condition.	(1) Use multimeter to test the voltage and current of the PIN for microprocessor. The correct number should be > 3.0V and < 0.8V. (2) Count the input ports and output ports on our microprocessor, if the number matches our requirements, it is verified. (3) During the game there should be no obvious delay. (4) 13 LED matrix could get correct pattern. If there is a hazard problem, Din is disturbed and LED matrices would shown broken image.	10

The microprocessor functionality was shown during our final demo as the game was able to run. Most importantly is how we handle the hazard condition. The fourth requirement was not taken into concern until the later stage when we were trying to integrate our modules together. What the hazard condition means is that different signal is arriving at different times, which causes a spike in the digital signal. This little spike could be incorrectly read by our encoder and mess up the LED modules. The only way to solve this problem is to use PORT write which is able to write several ports in parallel. Our ATMega-328 can do PORT write.

3.4 Software

Table 7. Software RV

Requirements	Verification	Score
Software (1) Give the most admissive tic-tac-toe strategy (2) AI VS human and human VS human	(1) Our AI should be smart in the tic-tac-toe game. AI should not play randomly. Al should either win or draw. (2) We can play in two modes. At the beginning of the game we are able to select which mode to play. The verification should be self explaining during the demo.	10

Our software is able to support multiplayer and singleplayer. For single player, we make a program for AI. There are 40 chess combination strategies stored and hardcoded in our program. During the game, the AI placed the pieces according to the min-max tree which would offer the AI a path with the highest reward. Our AI would block uses move and find chances to let three in a row.

4 Costs

Our costs is composed by two parts: human labor fee and fixed manufacturing prototype costs.

4.1 Parts

Table 8. Parts Costs

Part	Cost(prototype)	Cost(bulk)
FORCE SENSING RESISTOR,0.5 INCH ,CIRCLE, 1oz-22LB,FLEXIBLE * 9	$ 134.55	$58.41
SainSmart MAX7219 Red LED Dot Matrix * 13	$107.77	$107.77
Atmega328p-pu Chip	$4.37	$4.37
PCBs	$3.10	$1.95
Lithium-ion batteries ???		
Total	$249.79	$172.5

4.2 Labor

Regard our human labor; we estimated our salary to be $38/hour. Our project started at the fourth of this semester. So entirely there are 16 weeks. In Saturday and Sunday, the work is considered overtime; salary is 1.5x regular salary which is 38*1.5. Each week we actually work 3 days on workday and 1 day in the weekend, 8 hours per day. The calculation of our labor cost is:

$$3\ people\ *16\ weeks\ *\left(\frac{\$38}{hr}\ *8*3\ h\ +\frac{\$38}{hr}*1.5\ *8\ h\right)\ =\ \$\ 65{,}664$$

So the total cost of our project is 36902.29 dollars. We do not consider the other cost such as electricity, air conditioner, and additional expense. The real cost would be a little bit higher than what we calculate, but the difference should be small.

5. Conclusion

5.1 Accomplishments

We successfully build a Tic Tac Toe Game Box that has two play modes: Human VS AI mode and Human VS Human mode. In Human VS AI mode, our AI can always make the best move against the human. In both modes, the scoreboard keeps tracks of the scores of both sides. Whenever the player presses the reset button, the game will restart, and the box will reset the score to 0. Through this project, we learned how to design PCB, how to select IC chips, and how to program in the micro-controller.

5.2 Uncertainties

The uncertainties are largely due to the physical design of the structure that holds the LED matrix and the pressure sensor. In our design, the pressure sensor is stuck to the bottom metal plate, and the LED matrix is screwed with the metal plate above the pressure sensor with springs between the two plates to hold enough distance so that pressure sensor is not touched due to the weight of the metal plate with LED matrix. However, The screws that holds the metal plates can be unstable to cause false positive errors. Additionally, the spring can get stuck into the screw thread and hinder the screw.

5.3 Ethical considerations

One risk we need to take into concern is the use of Children. In 2012, the U.S. Consumer Product Safety Commission (CPSC) reported 11 toy-related deaths and an estimated 265,000 toy-related injuries treated in emergency rooms[6]. Our tic-tac-toe game box is an electronic device that could be dangerous for small children. We can have prominent warning labels so that kids will not access the dangerous circuit hardware of our game box. The cords and strings are hidden inside the game box so that it would not pose a strangulation hazard for infants. We would set the age limit for our tic-tac-toe game for people older than eight years old according to the U.S. Consumer Product Safety Act[7]. The panting and material for the cub box are degradable and environment-friendly. The pigment in the box painting is non-volatile and non-toxic[8].
According to Student Rights and Responsibilities part 4 Academic Integrity Infractions 1-402 Academic Integrity Infractions[9], we didn't use or attempt to use any code, data, circuit design or idea without authorization or citation. We highly appreciate all professors, TAs, and any other people who gave us helps during the project. According to IEEE code of ethics # 7[10], all sources of help and reference would be acknowledged and included in the reference section.
The input voltage of our game box would be only 5 V, which is very safe for play even if the coat of the wire is worn out, and the player touches the broken part. We helped each other during the whole process and made sure that everyone in the team is on the same page.

5.4 Future work

We are going to continue developing our game box for its potential development. We will transfer the game box for more complex abstract strategy games. For example, chess and even Go(weiqi).

Appendix A: Requirement s and Verification

Requirements	Verification	Score
Power Supply (1) Need to continuously output an 5.0 voltage. The voltage varies within a 5% range. (2) Outputs current should be at least 10 mA	(1) Use multimeter to check the power source voltage. Should be 5.0 (+-5%) V (2)Use multimeter to check the current. Should be under 10mA.	5/_5_
A/D convertor module (1) The A/D converter must be able to give Digital signal as its output. When player press on the FSR, the A/D converter would give output voltage > 2.65V as digit 1; while the standby output voltage of the A/D converter would be < 1.0 mV as digit 0. (2) Encoder(multiplexer) should be able to read all nine inputs from A/D converters and give correct 4 bit output message. (3) Our A/D convertor module circuit should take the little deviation of the resistance in 9 resistors into account and correct the error.	(1) (a) Use multimeter to check the voltage for output1 pin in our comparator. Should be bigger than 2.65V when outputting digital1. Should be smaller than 1mV when outputting digital 0. (b) Use Arduino digitalread() to check output from A/D converter. Should read 1 when we press the pressure sensor. Should read 0 when we release the pressure sensor. (2) Use Arduino to test the output from Encoder. The 4 bit output should match which pressure sensor that we pressed. Only one sensor can be pressed and recognized by the microprocessor at one time. (3) There may be some little error and deviation for our pressure sensor. We should measure the resistance of each individual resistor one by one and adjust our circuit module according to the condition of the pressure sensor. If we successfully handle this case, we can read correct output from our Arduino so the verification is quite straightforward.	10/_10_

LED module (1) LED matrix should be able to light up all its pixels. (2) LED matrix should be able to show cross pattern and circle pattern. (3) 9 LED matrix should be able to show corresponding patterns collaboratively.	(1) When LED is powered. We should correctly configure the LED component (led intensity, display mode) and give enough power supply. All LED pixel should lighted up if correctly been initialized. (2) We would use Arduino to give data to LED's Vin. We would let it display either "X" or "O". "X" or "O" should be displayed on the LED. (3) After we connected all the 9 LED module. We should able to decided which one of the 9 LED module to display and what picture pattern is been display on this LED. We would use Arduino to give signal to our LED module. The LED module should able to understand the signal from Arduino and behave correctly. For example, our test case can be: (a) let the leftmost LED to display "X" (b) let the middle LED display "O" (c) let middle LED change from "O" to "X"	10/_10_

Microprocessor (1) Microprocessor need to give voltage > 3.0 V as digitalwrite 1 , and voltage < 0.8 V as digitalwrite 0 when it is giving data the LED matrix. (2) Microprocessor of choice(AT Mega-328) should have 4 ports for input; 5 ports output for LED matrix on the tic tac toe panel; another 3 ports output for LED score board. (3) Microprocessor should be able to communicate with LED modules and A/D Converter module within 2 seconds.	(1) Use multimeter to test the voltage and current of the PIN for microprocessor. The correct number should be > 3.0V and < 0.8V. (2) Count the input ports and output ports on our microprocessor, if the number matches our requirements, it is verified. (3) Connect the microprocessor to a pc and log out the state of the microprocessor. Press the forcing sensing resistor and check if the microprocessor can take that input and light up the corresponding LED within 2 seconds.	10/_10_

LED score display board (1) LED matrix should be able to light up all its pixels. (2) LED matrix should be able to show cross numbers. (3) 4 LED matrix should be able to show corresponding patterns collaboratively.	(1) When LED is powered. We should correctly configure the LED component (led intensity, display mode) and give enough power supply. All LED pixel should lighted up if correctly been initialized. (2) We would use Arduino to give data to LED's Vin. We would let it display number zero to nine. (3) After we connected all the 4 LED module. We should able to decided which one of the 4 LED module to display and what picture pattern is been display on this LED. We would use Arduino to give signal to our LED module. The LED module should able to understand the signal from Arduino and behave correctly. For example, our test case can be: (a) let the leftmost LED to display "9" (b) let the middle LED display "0" (c) let middle LED change from "0" to "2"	5/_5_
Software (1) Give the most admissive tic-tac-toe strategy (2) AI VS human and human VS human	(1) Our AI should be smart in the tic-tac-toe game. AI should not play randomly. Al should either win or draw. (2) We can play in two modes. At the beginning of the game we are able to select which mode to play. The verification should be quite self explaining during the demo.	10/_10_

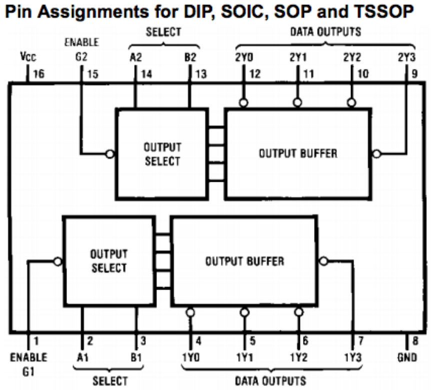
Appendix B Chip Configuration

Order Number	Package Number	Package Description
MM74HC139M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150" Narrow
MM74HC139SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
MM74HC139MTC	MTC16	16-Lead Thin Shrink Small Outline Package (TSSOP) JEDEC MO-153, 4.4mm Wide
MM74HC139N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram

Pin Assignments for DIP, SOIC, SOP and TSSOP



Truth Table

Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Level
L = LOW Level
X = Don't Care

Fig.21 Connection Diagram

Nexperia **74HC42**

BCD to decimal decoder (1-of-10)

6. Functional description

Table 3. Function table[1]

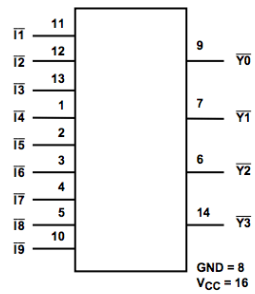
Inputs				Outputs									
3A	2A	1A	0A	$\overline{0Y}$	$\overline{1Y}$	$\overline{2Y}$	$\overline{3Y}$	$\overline{4Y}$	$\overline{5Y}$	$\overline{6Y}$	$\overline{7Y}$	$\overline{8Y}$	$\overline{9Y}$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	H	L	H	H	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H	H	H
H	H	L	L	H	H	H	H	H	H	H	H	H	H
H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	H	H	L	H	H	H	H	H	H	H	H	H	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H

[1] H = HIGH voltage level;
L = LOW voltage level

Fig. 22 74HC42(decoder)

CD54HC147, CD74HC147, CD74HCT147

Functional Diagram



TRUTH TABLE

INPUTS									OUTPUTS			
$\overline{I1}$	$\overline{I2}$	$\overline{I3}$	$\overline{I4}$	$\overline{I5}$	$\overline{I6}$	$\overline{I7}$	$\overline{I8}$	$\overline{I9}$	$\overline{Y3}$	$\overline{Y2}$	$\overline{Y1}$	$\overline{Y0}$
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

H = High Logic Level, L = Low Logic Level, X = Don't Care

Fig. 23 CD74HC147 [1]

References

[1] High-Speed CMOS Logic 10-to-4 Line Priority Encoder, Texas Instrument, November 2003, http://www.ti.com/lit/ds/symlink/cd54hc147.pdf

[2] 8-bit AVR Microprocessor, Atmel Corporation, 11/2016, http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

[3] Figure 2 force vs resistance, FSR force sensing resistor, Interlink Electronics, https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf

[4] Fig.1 FSR construction. FSR Force Sensing resistor Inregration Guide and Evaluation Parts Catalog, https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf

[5] Dilip Raja, 8*8 LED Matrix using Arduino, Circuit Digest. Nov 20, 2016. https://circuitdigest.com/microcontroller-projects/arduino-8x8-led-matrix.

[6] BU-410: Charging at High and Low Temperatures:http://batteryuniversity.com/learn/article/charging_at_high_and_low_temperatures

[7] United States Consumer Product Safety Commission, “Regulations, Mandatory Standards & Bans”, 2014,
https://www.cpsc.gov/Regulations-Laws--Standards/Regulations-Mandatory-Standards-Bans

[8] Natura Veda Innovations from nature, “The Dangers of FD&C Color Pigments”, 2017
https://www.naturaveda.com/pages/the-dangers-of-fd-c-color-pigments

[9] University of Illinois at Urbana-Champaign, article 1 - student rights and responsibilities, 2008, http://studentcode.illinois.edu/article1_part4_1-402.html

[10] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 29- Feb- 2016].

[11] Minkang Yang, University of illinois at Urbana-Champaign, ECE 445 Block Diagram, 2017,
https://www.draw.io/#G0Bwtp5vDqfIdWbnEyWUtGOG5RTGc