6.115 Final Project Lab Notebook

3D Tic Tac Toe

Andrew Xia

1 Overview

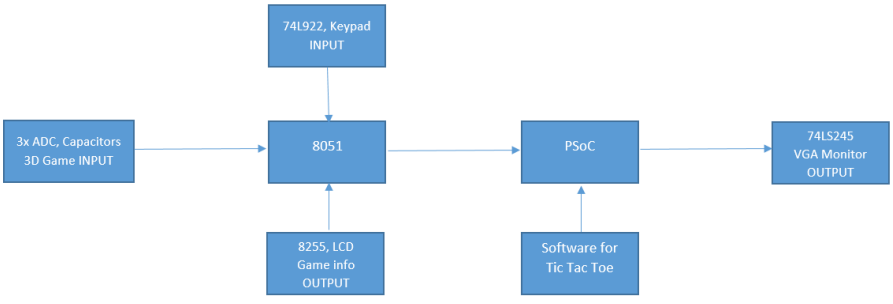
April 20, 2016 Proposal: For my final project, I hope to build a 3D touchless tracking interface and connect it with my 8051 and PSoC to allow users to play 3D tic tac toe with either each other or a computer AI.

What interested you in the idea? I am interested in this idea because I want to work on a project that has both an interesting and flashy hardware and software component. Writing the AI software for my 3D tic tac toe board will be a very rewarding and educational experience for me, and building the 3D tracing interface will also be very fun.

Why is this project interesting? This project is interesting because while 2D tic tac toe is a game familiar to all, 3D tic tac toe represents a game that while it is familiar, it can also be novel and entertaining. This project also uses a 3D touchless interface for user input, which is another novel way of interacting with software, as we are used to touching buttons or screens to communicate with devices in our daily lives.

1.a Hardware Overview

The key hardware component of my project will be building a 3D touchless interface, which is shown in the image to the side. The three aluminum foils (or a similar material) on the three sides are the capacitors that the human finger complements, and thie distance from each of the three capacitors will definite the capacitance of the capacitance, which we can measure using an ADC, and effectively pinpoint the 3D location of the human hand.
I will use a 74LS245 chip to connect my PSoC to a VGA cable which can then display information on a screen. I will use what was covered in lecture to complete this task.
Below is a preliminary hardware schematic of my final project:



1.b Software Overview

As for my software, I will code my project in C, using the PSoC as my main control module, as it would would be more efficient to do so over writing code in assembly for the 8051. The Tic Tac Toe AI is not too software heavy, so I hope to develop the basic framework for my code and spend the rest of my extra time focusing on hardware improvements.
Here is the high level software workflow for my final project:

1. Initiate & Reset Tic Tac Toe Board
2. Mode 1: two player Tic Tac Toe
   (a) User can view & rotate, using 74L922 or capacitive-based sensors, current 3D tic tac toe board on VGA display
   (b) User adds piece to the board by leaving hand in position of 3D touchless tracking interface for extended amound of time
   (c) After entering piece on board, the other player can play.
3. Mode 2: one player AI Tic Tac Toe
   (a) User can view & rotate, using 74L922 or capacitive-based sensors, current 3D tic tac toe board on VGA display
   (b) User adds piece to the board by leaving hand in position of 3D touchless tracking interface for extended amound of time
   (c) After entering piece on board, computer plays against user and submits his move through AI algorithm.
4. Reset the game and enter either mode through the 74C922 Keypad

1.c Format of my Lab Notebook

I plan on formatting my lab notebook in this latex file, in which each day I work on the lab will be a section, and I can also reorganize my lab notebook to be thematically organized by the module that I will work on.
Each day, or section, will contain an *introduction* section in which I discuss the goals that I have for the day, and I will detail the tasks that I will work on in the subsections for each day.

2 April 21, 2016

Goals I have for today / **done** if task completed

- Set up github for version control of my lab notebook and code **done**
- Build my initial capacitance based sensor
- Set up my computer so I can code in C **done**

2.a Setting up github

I am following the tutorial here to create a new github repository. It will be located in my 6.115 dropbox folder, and on my github account under the repository 6.115-final-project

2.b Building my initial capacitance based sensor

References include this Makezine link.
Notes on building a capacitance based sensor:

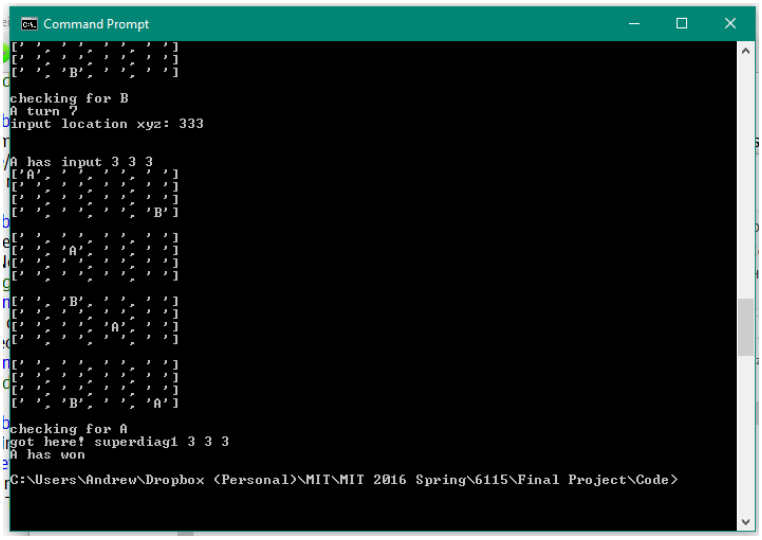
1. Beacuse humans have very little capacitance $10pF$, the time constant to which the plate would charge up based on the distance of the finger to the plate should be very small. Thus, I may need to think of more intelligent methods to build an effective capacitance based sensor.
2. I can design different circuits

2.c Software development

To install **C**, I am following this website to install a C compiler for my computer. It is in fact better to code in Linux rather than C, so I will be using linux/ubuntu when needing to test my Tic Tac Toe code on my computer.

To install the Cypress PSoC creator IDE, I am following the instructions here. Having PSoC Creator on my computer should help increase my productivity and workflow.

I completed a two player Tic Tac Toe game coded using python today. The logic of the game is documented in my code. I plan on investigating more into AI algorithms, potentially using minimax or alpha beta pruning. This website has pretty good information on evaluation methods and algorithms for 3D Tic Tac Toe.



Above is an image of my python 3D tic tac toe script running

3 April 24 2016

Goals I have for today / **done** if task completed

- Learn how to use Cypress EDS Laser Cutter **done**
- Look into VGA connection to PSoC, how to connect my capacitance based sensor

3.a Using the Cypress EDS Laser Cutter

I printed out a circular acryllic piece today with ny name on it. Some things to remember: I should use CorelDRAW or solid works to design my cuts. The *red* line must be set to hairline width for cuts. The *blue* line must be set to hairline width for scones. The *black* areas must can be set to anything for rasters.

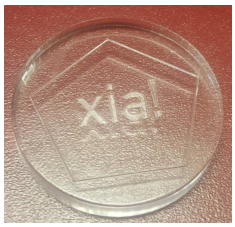
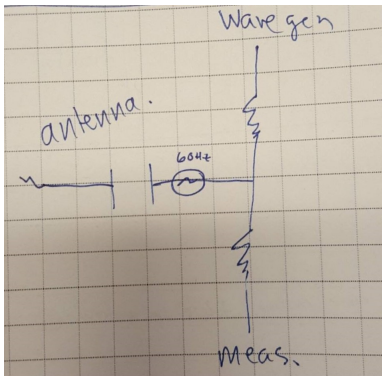


Figure 1: test.

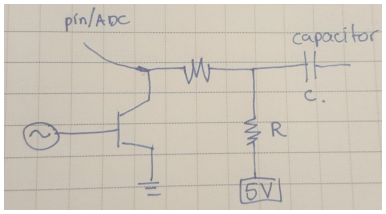
3.b Working on my capacitance based sensor

The initial circuit that I tested did not work. It was picking up a lot of 60Hz noise (which is coming from the AC power that exists all around the lab). The top resistor has value $220k\Omega$ and the bottom resistor has value $10k\Omega$.



This circuit does not work.

I am now trying a new circuit, designed as follows according to the Makezine article



I am able to sense when I touch the cardboard aluminum foil capacitor plate, but I am unable to get a different capacitor RC time constant reading when I vary my hand distance from the aluminum foil.
I can consider designing a twin t notch filter to filter out the 60Hz noise.

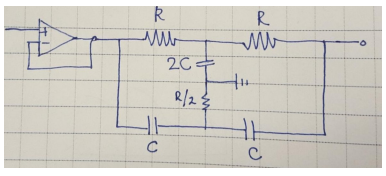
4 April 25 2016

Goals I have for today / **done** if task completed

- Finalize design for my user input
- Look into connecting PSoC to 8051 and VGA display
- Continue Building my 3D Tic Tac Toe AI [**done**]

4.a Designing the Capacitance Based Sensor

This is the twin t notch filter that I have in mind:



The formula for the cutoff frequency is $f_c = \frac{1}{2\pi RC}$, and I am using $C = 100pF$ and $R = 27k\Omega$. However, I am unable to filter out a default 60Hz.

4.b Connecting PSoC to 8051

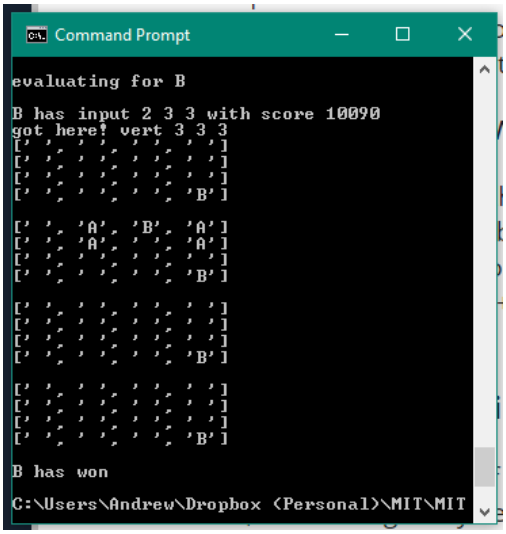
Check out this link to see how to connect my 8051 to my PSoC.
Today, I soldered the Sparkfun connector piece, and I got the LA checkoff showing that I could successfully power my PSoC using R31JP power. I finished the pdf linked above and got the following to display:
(insert image)

4.c Using the 74LS245

The purpose of the 74LS245 chip is to connect my PSoC to a VGA display, so I can effectively visualize the 3D tic tac toe game.

4.d Building my 3D Tic Tac Toe AI

I completed building my AI for my 3D Tic Tac Toe Game in python today. I thought about running a brute force tree search to exhaust all combinations of the 3D Tic Tac Toe Game, but that would be on the order of 64! moves and would be a bit complicated. Instead, I wrote an evaluation metric to evaluate the values of certain locations on the boards, and the AI would place its next move based on the highest scoring location. My scoring works as follows, for a certain row/diagonal of 4 that does not have both an AI and player move:



1. If the AI has a 3 in a row, connect it
2. If the opponent has a 3 in a row, block it
3. If the opponent has a 2 in a row, block it
4. If the AI has a 2 in a row, connect it
5. If the AI has a 1 in a row, connect it
6. If the row has nothing, add a move
7. If the opponent has a 1 in a row, block it

The cumulative sum of the scores after checking all potential connect 4 paths will determine the next place to play. You can take a look at the code here.

As for further work to be done, I should (a) port the game to C to put on the PSoC, (b) consider difficulty settings for the game, (c) WORK ON THE HARDWARE ASPECT!

5 April 26 2016

Goals I have for today / **done** if task completed

- Connect PSoC to VGA display
- Port 3D Tic Tac Toe to C **done**
- Buy capacitance based sensing chips

5.a Connect PSoC to VGA display

I got a PSoC module from Prof. Leeb today that will allow me to connect my PSoC to the module using a Serial Port, and the module will convert the serial port data to something displayable on the monitor display through a VGA display port.
I acquired a null modem cable and a male-to-female serial cable, which will allow me to connect my R31JP to the PSoC-based VGA board. See this document for more help.

5.b Port 3D Tic Tac Toe to C

I also spent time at home and booted into my Ubuntu and ported my python Tic Tac Toe code to C. I now have a working two player version that can run on terminal, and I plan to write the AI maybe the next day and get it running on my PSoC soon.

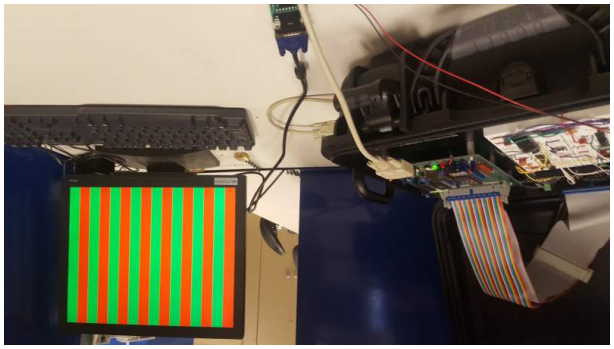
6 April 27 2016

Goals I have for today / **done** if task completed

- Connect 8051 to VGA display **done**
- Connect PSoC to VGA display **done**
- Port 3D Tic Tac Toe code to PSoC

6.a Connect 8051 to VGA display

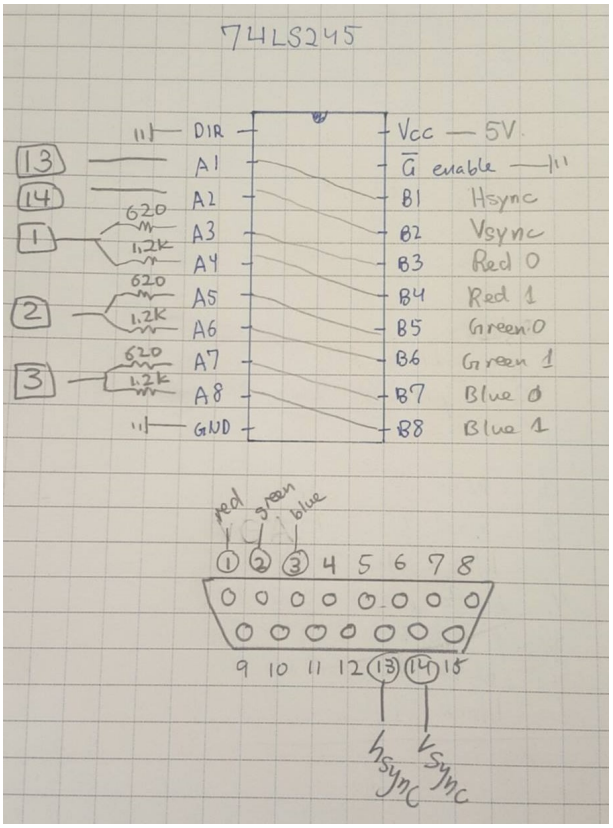
Following the instructions from this document, I wrote a simple `vga.asm` file to display red and green stripes on a VGA display. Consult the image below to see my display.



6.b Connect PSoC to VGA display

Using the PSoC on a board, I was also able to connect my PSoC 5 to the VGA display. I am using the UART_PutChar() function to write to serial. I am also powering the 5V PSoC on a board using a phone charger that I took apart.

Eventually, I will construct a simple circuit to connect my PSoC to my VGA display board using a 74LS245 chip. The 74LS245 datasheet can be found here or here. Below is a schematic of the desired circuit:



I will first try to get my game running using the PSoC on a board before creating my own wiring schematic.

7 April 28 2016

Goals I have for today / **done** if task completed

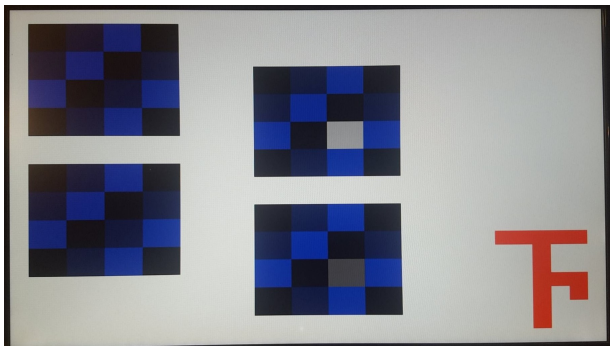
- Display Tic Tac Toe Board on PSoC **done**
- Port Tic Tac Toe Code to PSoC **done**

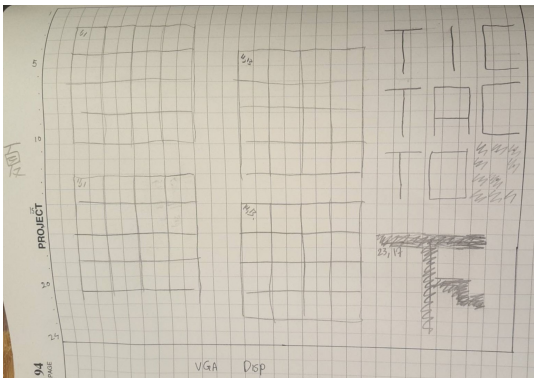
7.a Display Tic Tac Toe Board on PSoC

I am running some problems while trying to program my PSoC. I have been getting `Build error: The command 'arm-none-eabi-gcc.exe' failed with exit code '1'.` multiple times, even when I remodify the code to the original state the message does not dissappear. I will need to debug this.
Apparently the solution to this is to not `#include<disp.c>` or whatever file that I need to include; it seems as if the files are included implicitly. However, I need to declare structures in the `main.c` file; see my April 29th `tic tac toe board displays on VGA` commit to see where my code is. I'll ask a TA and see what's up.

7.b Port Tic Tac Toe Code to PSoC

Having written my c files, ion which I can play 3D tic tac toe on my linux terminal, I will now try to run my code on the PSoC. I managed to port the two player code over, although I seem to be having some trouble with the `#include` dependencies of my `tic_tac_toe_ai.c` file that requires `tic_tac_toe.c`. I was, however, able to get the Tic Tac Toe board to display on my VGA. The late night was worth it. See the images below for what I got to display after each player moved once (left) and my visual layout design (right). (The Chinese character Xia is a homonym for my last name.)





Here are some further palces for improvement:

- Learn more about C memory allocation/freeing so I do not have memory leakage
- Think about the visual VGA display; how can I improve the aesthetics
- Get AI working on PSoC
- Think about my user input interface and how I want to link up the 8051

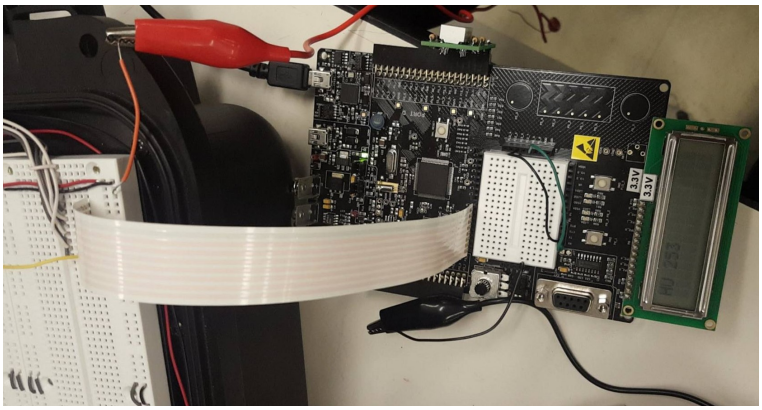
8 April 29 2016

Goals I have for today / **done** if task completed

- Connect 8051 to PSoC via Databus **done**
- Get AI working on PSoC

8.a Connect 8051 to PSoC via Databus

Today, I want to connect my 8051 to PSoC via databus. First, I configured my PSoC jumpers to take 5V instead of 3.3V.
I will use port a of my 8255 on the microcontroller to send databits to Port 3 of the PSoC. I configured my port 3 to be input and *high impedance digital* with a low drive. You can see my code at `main.c` at my current commit `connected 8051 to PSoC via 8255 databus` to see the code I wrote.
Keep in mind that the PSoC has to be at the same ground as the microcontroller, otherwise I may get wacky 16v readings. Also, for some reason the PSoC can't read an input of 0, but I will try to fix that. Below is an image of my PSoC LCD displaying the digital value fed from the 8255 port A. (Which I can generate by writing `WFE33 = 80; WFE30 = FD;`).



8.b Writing software to connect 8051 to PSoC via Databus

Now that I have the basic connection completed, I should also develop a protocol of how I want to write my code on the 8051 and PSoC side to determine when the PSoC should be reading a signal from the 8051 and when it should not be reading. (For example, if the 8255 is set at `80` all the time, I should not continuously print `80`.)
I wrote `keypad_to_8255.asm` so now when I press a keypad, I can send the corresponding ascii value to the 8255 (and to the PSoC). However, my 8255 seems broken in the sense that PA0 is always high. I might get a new 8255.

9 May 4 2016

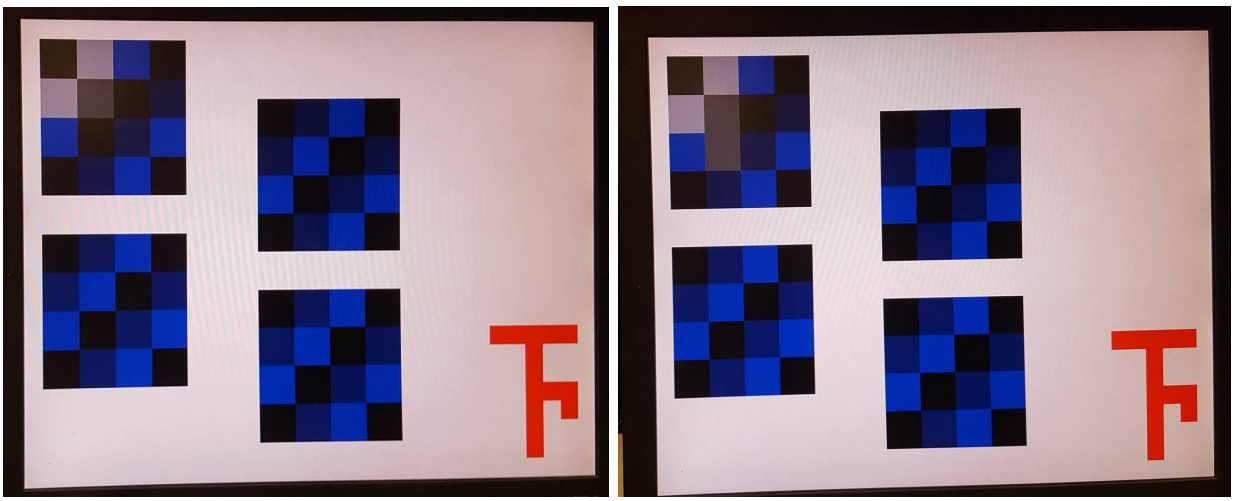
Goals I have for today / **done** if task completed

- Program 8051 to be able to send Tic Tac Toe game signals to PSoC **done**
- Get AI working on PSoC **done**

9.a Program 8051 to be able to send Tic Tac Toe game signals to PSoC

For some reason, my 8255 is not completely reliable (or maybe my PSoC port 3 isn't completely reliable). Sometimes, PA0 is unresponsively high (which means that I am only able to generate even numbers) or sometimes PA6 is unresponsively high (which pushes all ASCII values up by 64). It could also be a software

With that said, here is a general overview of my program flow. In my 8051 code, when I press a keypad button, the 8255 sets PA the corresponding ascii symbol for $80 \times 256 \times 2$ machine cycles (this value can change; see `keypad_to_8255_1.asm`, and then resets the value to 0.
The PSoC reads the values from the 8255 continuously; displaying the current read (which most of the time is 0) and the most recent nonzero read. When we get an integer input; we think of it as that the user has inputted a value, and we 'play' a move on the 3D tic tac toe board. See the github commit `can play 2d tic tac toe on vga via 8051 keypad`. Below is a picture of the game so far:



The light gray colors were fed in from the keyboard as player 1; the dark gray as player 2. From the left image to the right image, the player has placed a value at 1,2,0

9.b Get AI working on PSoC

I'm still not exactly understanding how the implicit include in PSoC creator works, but one solution to solve this problem is to keep my AI and non-AI code in the same c file. See my `got ai to work in tic_tac_all.c without keypad` commit to see this stage of the project.
The solution is to include the header `disp.h` file in `main.c` in which `disp.h` defines all of the methods (and includes the structures) but does not include the c file. In fact, never include the c file. See my `added .h files so now psoc include works` commit to see this stage of the project.

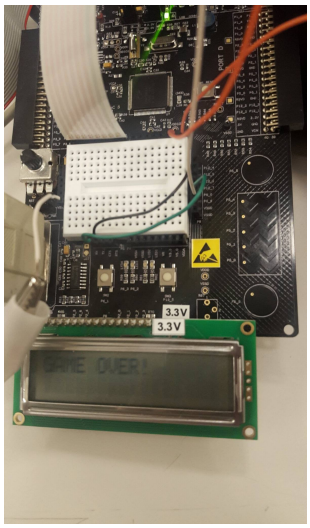
The next thing I need to do is get a random seed for my random number generator (otherwise I will be generating the same random numbers every time and thus the program will no longer be random).

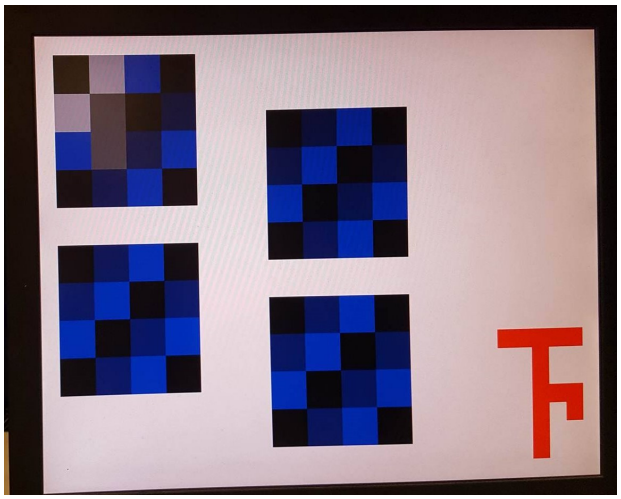
I'll let P6_6 be my input which will be very noisy. EDIT: not exactly working because I think as I've grounded my PSoC to the 8051 and the input noise is too predictable.

I'll now try to use the PSoC clock module to generate random seed bits. I will use the 8254 on my 8051 to generate random signals.

9.c Two Player Game Interface Running

I also implemented the full game tonight, that is playable using the keypad and two high/low wires into P0_0 and P0_1 (which are the orange wires below). Now I can input across planes.





I've committed the code in `random using 8254 timer, got 2 player implemented with p0 low/high cables`.
Next, I need to figure out a better user input interface.

10 DATE 2016

Goals I have for today / **done** if task completed

- Task
- Task