

LOG BOOK

FOR

50.002

1D Project

Li Xingxuan

1002189

Index:

MHP Meeting 1:	3
MHP Meeting 2:	4
Lab Checkoff Fail Attempt	5
1D Meeting 1 (for CheckOff 1):	6
1D Meeting 2 (for CheckOff 1):	7
1D Meeting 3 (for CheckOff 2):	8
1D Meeting 4 (for CheckOff 2):	9
1D Meeting 5 (for CheckOff 2):	11
1D Meeting 6 (for CheckOff 3):	13
1D Meeting 7 (for Poster Submission):	16
1D Meeting 8 (for CheckOff 3):	17
1D Meeting 9 (for CheckOff 3):	19
1D Meeting 10 (for CheckOff 3):	21
1D Meeting 11 (for final exhibition):	23

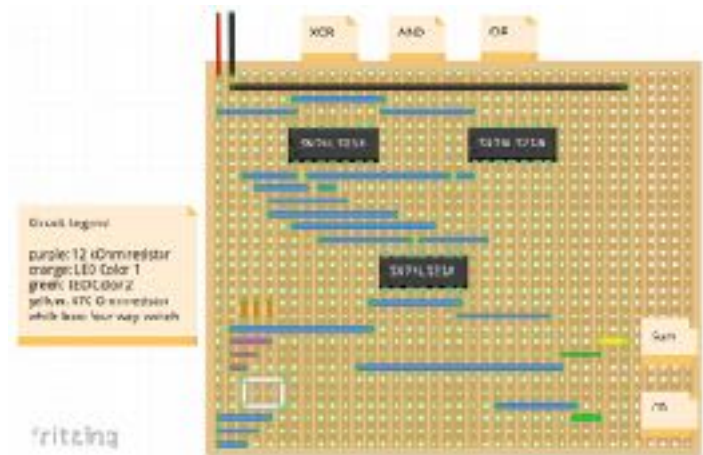
MHP Meeting 1:

- Time:

1. 20:00 ~ 24:00pm, 20th September
-

- Before meeting:

1. Before meeting, I draw the truth table and circuit of 1-bit full adder. Guanlun put it in freezing.
2. Also I downloaded Mojo IDE and all the stuff. I tried to follow the tutorial to build my first mojo project. However, it didn't work. So I leave the problem to the meeting.



- Goal of the meeting:

1. This is the first meeting of our team and the first meeting of MHP project. The goal of the meeting is to distribute the job and getting everyone in the same pages.
-

- What we do during the meeting:

1. During the meeting, I explained the circuit of 1-bit full adder to my teammates to make sure that everyone understand the basic principle of the adder and know how it works.
 2. After that, Guanlun and I are assigned to do the mojo part, which is to use finite state machine to do a automatic testing. The other two teammates are assigned to solder the adder.
 3. After assigning the job, Guanlun and I started working on the mojo. As I mentioned before, there were few problems to set up mojo IDE. It took us about half an hour to fix all the problems and finally get it working.
 4. Since I never used Lucid before, I started reading the tutorial and tried several sample projects.
-

- What I learned:

1. I learned the basic syntax of Lucid.
-

- To-do list after meeting:

1. For Guanlun and I, we need to finish Lucid tutorial as soon as possible to get started with our MHP project.

MHP Meeting 2:

- Time:

1. 20:00 ~ 24:00pm, 25th September
-

- Before meeting:

1. Before this second meeting, due to the tight schedule, I haven't finished the Lucid tutorial. I've read through IO Shield tutorial and still need to read the finite state machine tutorial.
-

- Goal of the meeting:

1. Manage to finish the soldering of the adder.
 2. Manage to realise manual checking on mojo.
-

- What we do during the meeting:

1. During the meeting, the first version of soldering seems to have some problems. We spent about half an hour checking the circuit and finally our two teammates who do the soldering part decide to redo the whole adder.
 2. After that, Guanlun and I start to focus on the mojo part. We firstly finish the Lucid tutorial and start to work on our MHP.
 3. We divided the work into two parts. The first part is led part, including the seven-segments and the 24 small less. My job is to light up the corresponding led when we are doing the testing case. This is a relatively easier part compared to the second part, which is taken by Guanlun.
 4. The second part is to write a finite state machine to self generate testing cases to check our adder.
-

- What I learned:

1. During the meeting, I learnt how to control the led and it can be useful in the future 1D project.
 2. After finishing my led part, I also started helping Guanlun with his finite state machine part. So I learned the basic structure of writing a finite state machine.
-

- To-do list after meeting:

1. For our finite state machine, we have 'IDLE, TEST, PASS, FAIL' four state. In the 'TEST' state, we have few testing cases. One major problem is that we cannot find a way to create a delay between theses cases.
2. One way we tried to solve the problem is to write an iteration between the cases. So that there will be calculation delay. However, after consulting Prof Oka, we realised that a for loop in Lucid is calculated in parallel so that there will be no calculation delay.
3. Prof Oka suggested us to write a counter.

Lab Checkoff Fail Attempt

- Time:

1. 4th October
-

- Problems:

1. When $C_{in} = 1$, $A = 1$, $B = 1$. S and Cout are supposed to be 1. Two led will light up. However they didn't light up.
-

- How to fix:

1. The problem lies n USB cable or our battery holder.
2. We tried with DSL NI ELViS board 5v power supply and it starts to work.

1D Meeting 1 (for CheckOff 1):

- Time:

1. 21:00 ~ 22:00pm, 15th October
-

- Before meeting:

1. Finish lab alu code and list the functions that our mojo alu need to be done.
-

- Goal of meeting:

1. Get started with mojo all.
2. Distribute work.
3. My part is multiply and divide.
4. Guanlun and I also need to put all the code together.
5. Two more functions that we plan to do is not in the table, we also plan to do multiply and divide.

Operation	ALUFN(5,0)	hex
ADD	000000	0x00
SUB	000001	0x01
AND	011000	0x18
OR	011110	0x1E
XOR	010110	0x16
"A" (LDR)	011010	0x1A
SHL	100000	0x20
SHR	100001	0x21
SRA	100011	0x23
CMFEQ	110011	0x33
CMPLT	110101	0x35
CMPLT	110111	0x37

- What we do during the meeting:

1. During the meeting, the first thing that we did is to distribute the job.
 2. After that, we started to do our individual part.
-

- What I learned:

1. Using Lucid to write alu functions is much more easier than Jsim.
 2. However it will be a little bit painful to write the automatic testing.
-

- To-do list after meeting

1. We finish individual part in two days and meet again on 17th.
2. We plan to put the individual part together and write the automatic testing case in second meeting.

1D Meeting 2 (for CheckOff 1):

- Time:

1. 19:30 ~ 24:00pm, 17th October
-

- Before meeting:

1. Before meeting, I finished my part, which is multiply and divide.
 2. I also wrote an 'alu.luc' file to contain all the alu functions. So that we don't need to write our testing cases in 'mojo_top.luc'. The code will look nicer and cleaner. Otherwise, it will be in a mess.
-

- Goal of meeting:

1. To put all the alu functions together and write our automatic testing cases.
 2. Test on mojo and finish our project.
-

- What we do during the meeting:

1. Since we wrote our functions independently, the variable names are in a mess and we spent the first hour to modify our code so that it makes me easier to compile the code.
 2. Guanlun and I spent the rest of time working on the automatic testing. It is easy to check manually. However the seven segments led doesn't work in our automatic testing.
 3. We need to find another time to finish our testing.
-

- What I learned:

1. The most important thing I learned is that, when several people are working on the same project, it is useful to regulate the function name and structure of individual part before started. It will make it much easier to compile the code.
 2. In MHP, we wrote all the led and seven segment led in 'mojo_top.luc'. It is also possible to write them in other files. Some concepts in java can be applied here. It makes my life easier once I realise this.
-

- To-do list after meeting

1. We have finished manually testing part.
2. We still need to finish the automatic testing case.

1D Meeting 3 (for CheckOff 2):

- Time:

1. 20:30 ~ 22:00, 7th November
-

- Before meeting:

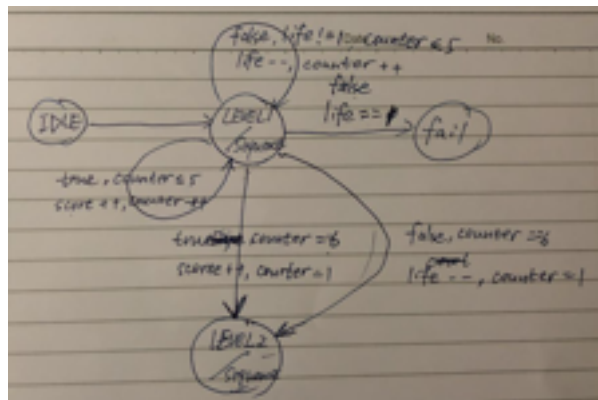
1. Before the meeting, we already had gave a brief thought to what game we are going to do.
 2. This most doable game is whack-a-mole.
-

- Goal of meeting:

1. The goal of the meeting is just to briefly talk about our ideas of what game we want to do.
 2. Because that we need to come out with several ideas and plans, we will split work during the meeting.
 3. Also, take whack-a-mole as a template, we come out with all the state diagram and game logic, as well as how we gonna use alu in our game.
-

- What we do during the meeting:

1. Firstly, we design the state diagram for whack-a-mole.



2. secondly, we came out with the game logic.
 3. Finally we discussed how we gonna implement alu.
-

- What I learned:

1. To design a game, we need to design the game logic first.
 2. Then we should design the game logic.
-

- To-do list after meeting

1. To design the game logic and state diagram for other games before next meeting.

1D Meeting 4 (for CheckOff 2):

- Time:

1. 21:00 ~ 24:00, 12th November

- Before meeting:

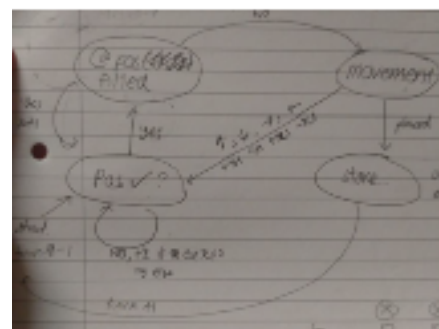
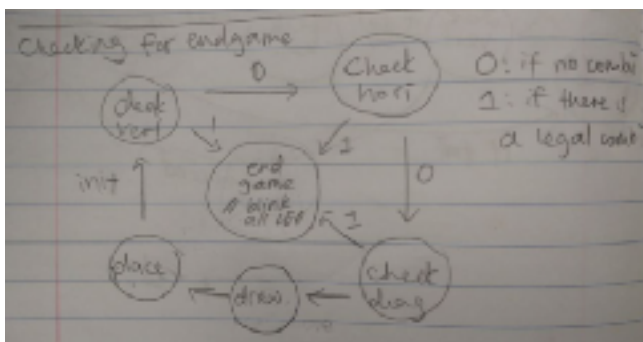
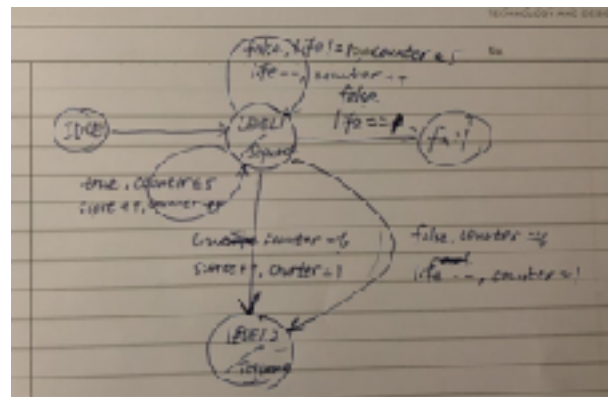
1. Before the meeting, we already had some ideas and thoughts about what game we are going to make. The most doable game is wrack a mole.
2. We did some research on other games and tried to come up with more plans as back up plans.

- Goal of meeting:

1. The goal of the meeting is to finish the report for 1D check off 2.
2. Come up with more plans for different games as back up plans.
3. Briefly consider about the schematic of the game and how do we implement the circuit and so on.

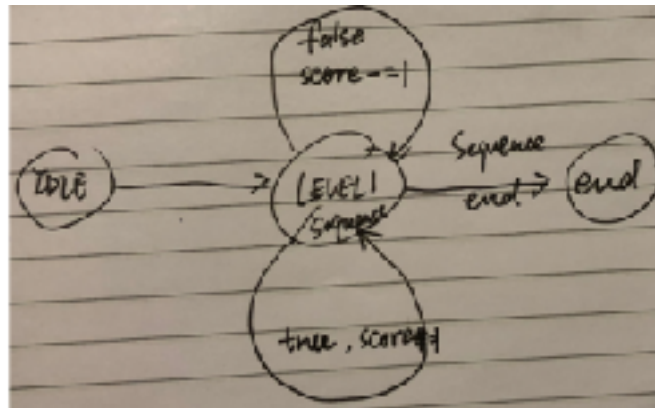
- What we do during the meeting:

1. During the meeting, we came up with 6 game ideas.
2. One is the most doable game: whack-a-mole. We came up with the schematic, the state diagram, as well as how we gonna implement the circuit and alu functions. We also design the game logic and the difficulty levels.
3. The second game is also doable but less interesting. Which is Five in a row. This game works just like tic tac toe but has a space that is at least 6*6, thus providing players with more opportunity to win later in the game. We came up with the detailed state diagram for placement of markers and for checking for 3 in a row. The material needed for this game is the fewest.



4. The third game we came out with is a music game. Choose the song you want to play and press the corresponding button when the lighted led arrives the last row. When the user press the correct button, speaker will play correct tone at the same time, score counter will +1. However, if wrong button is pressed, noise tone will be played.

Score will minus one if you press the wrong button. The state diagram for this game is quite simple. There are only three states which are: song selection, play , end. The material need is quite expensive by considering the LED matrix can be expensive. This game is much harder to implement than the two games before. Also, using speaker might require a lot of debugging.



5. The next game is one-sided pingpong. Or we can call it break the bricks. Given the budget and time constraints, resolution will be too low. As a result, it will be hard to represent angle shot of the ball.
6. The next game idea we came up with is Tetris. I think I don't need to introduce the game rules here. The game logic could be very hard to implement. And also consider of the budget. We may not be able to buy the screen with enough resolution that we need.
7. The last game is tank game. This game is actually the hardest game that we thought of. There are too many requirements to program because you have other opponent tanks randomly moving and generated. Also, the resolution is another issue.

- What I learned:

1. It is really hard to design a game. Considering the budget and limited time.
2. To design a game, it is important to start with the game logic and try to think of how to implement them.
3. Always take budget into consideration.

- To-do list after meeting

1. After this meeting, we will have a meeting with Prof Oka to discuss our game.
2. Need to decide which game we are going to do finally.
3. Once we decide which game we are going to do. There remains a lot of stuff like redesign the state diagram and reconsider about the game logic and need to start buying all the stuff we need.

1D Meeting 5 (for CheckOff 2):

- Time:

1. 19:00 ~ 23:00, 16th November

- Before meeting:

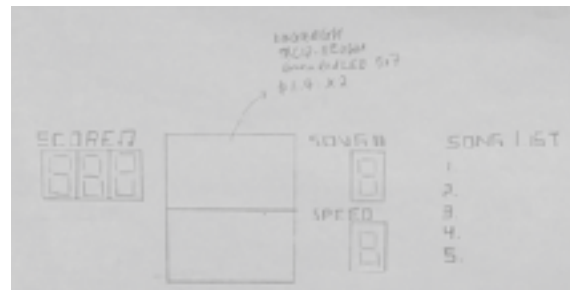
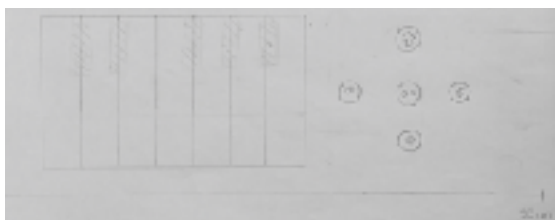
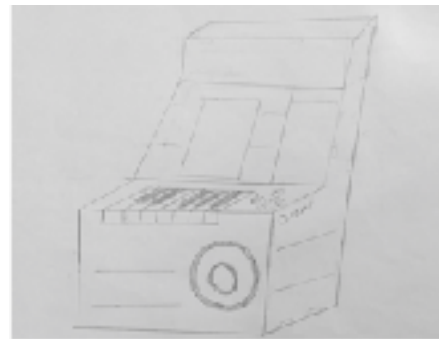
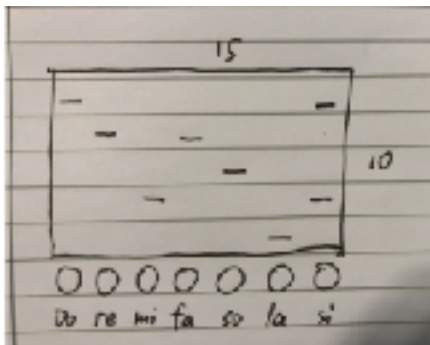
1. We have already came up with 6 game ideas and consulted Prof Oka.
2. We have designed game logic and state diagrams for each game.

- Goal of meeting:

1. The goal of the meeting is to finalise the game that we are going to make.
2. Also we need to redesign the game logic and the state diagram.
3. Finally, we finish our report and wrap up everything for check off 2.
4. Also we need to calculate the budget and decide what time to buy all the stuff.
5. Then we need to come out with a schedule for the following weeks.

- What we do during the meeting:

1. Finally we decide to do the piano game.
2. The following pictures are for the LED screen interface, console interface, top view of keyboard interface and top view of screen interface.

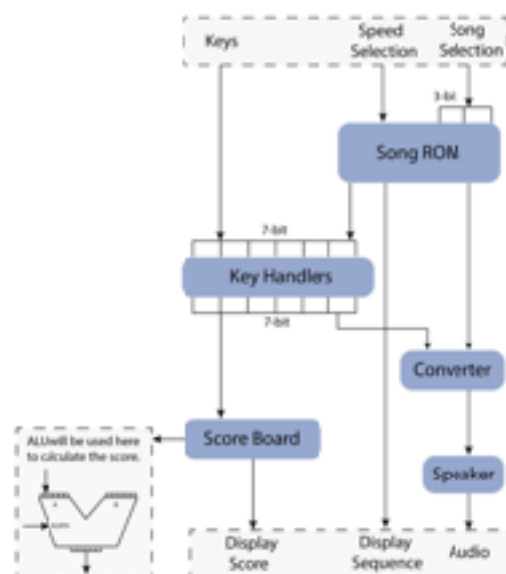


3. Then we calculate the hardware cost and it is within budget.
4. Then we designed the state diagram. There are 4 states: idle, song selection, play, end. When player press ok button, jump to song selection state. In song selection state, player selects song from number 1 to 5, song id will plus 1 or minus 1 depending on right and left button respectively. Players can also select speed from number 1 to 3. During play state, if key pressed == location of led, score will plus 1.

No.	Thing to do	Dateline
1	Design the hardware circuit	22nd Nov
2	Order parts / Steal parts CAD Housing Coding on Mojo	17th Nov
3	Work on code Laser cut CAD	24th Nov
4	Test hardware circuit in DS lab + debug code Order of extra parts if required	24th Nov
5	Debug hardware	30th Nov
6	Mount parts	30th Nov
7	Test & beautify Poster making	30th Nov
8	Report	8th Dec
9	Panggang lo (Exhibition)	12th Dec

Else if key pressed !=location of led, score will minus 1, speaker will play the wrong note. During the end state, maybe all LEDs will light up and final score will be displayed in 7-segment display on top left corner.

- Then we redesign the game logic and alu operations. We will use alu for add and subtract functions.



- What I learned:

1. Still, designing game logic and state diagram is the most important thing to start making a game.
2. Not only for hardware games, but also for software games.

- To-do list after meeting

1. In the next page.

1D Meeting 6 (for CheckOff 3):

- Time:

1. 15:00 ~ 15:30, 22nd November

- Before meeting:

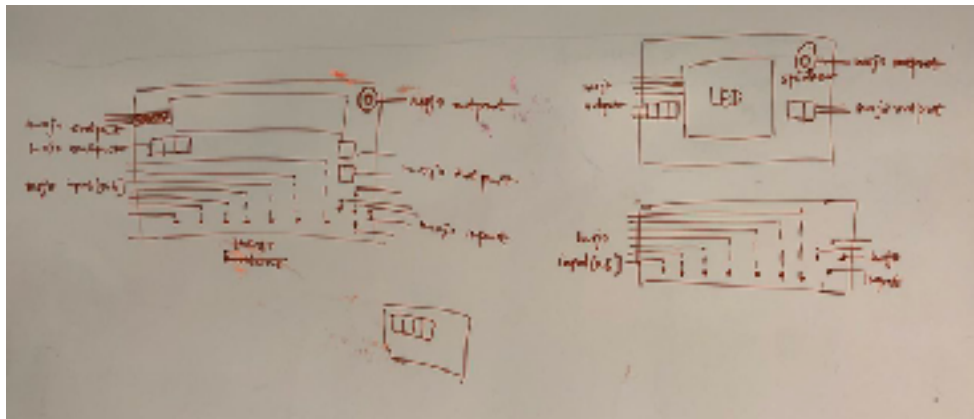
1. Before the meeting, we have decided that we are gonna do the piano game.
2. We have designed the game logic and state diagram.
3. We have decided how to implement alu.

- Goal of meeting:

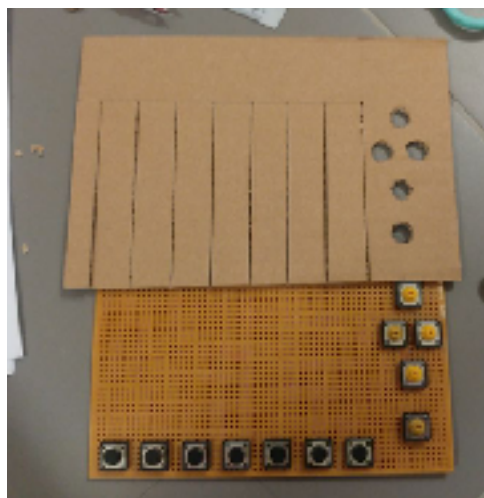
1. The goal of the meeting is to settling the circuitry of our game.
2. We also need to design the user interface.
3. Design the housing of our game.

- What we do during the meeting:

1. During the meeting, we designed the circuit for each strip board.



2. We also design the interface part by part.



- What I learned:

1. Circuit designing is both important for hardware part and software part.
2. We can design the circuit base on each strip board.
3. Or based on our circuit, we can separate them into several strip boards to make it easier to implement.
4. The designing of the circuit should consider the interface of the game.

- To-do list after meeting

1. For me and guanlun, we are supposed to order all the stuff.
2. Order speaker, order LED matrix.
3. Start to design the software implementation.

4. Start to think about the structure of the game and try to kick off the code writing as soon as possible.

1D Meeting 7 (for Poster Submission):

- Time:

1. 22:10 ~ 24:00, 29th November
-

- Before meeting:

1. Before the meeting, we have already roughly designed the circuit, the hardware team has started their design.
 2. The interface is partially done.
-

- Goal of meeting:

1. The goal of the meeting is to finish the poster for submission.
 2. Update the current progress.
 3. Start software implementation and split work.
-

- What we do during the meeting:

1. During the meeting, we finalised our poster.
 2. Guanlun and I started working on the LED matrix.
 3. We also did some research on speaker part.
-

- What I learned:

1. LED matrix can be controlled by spi board.
 2. We tried to implement LED matrix on spi board because it needs less pins. So that we can reduce the hardware connection issues.
-

- To-do list after meeting

1. During the meeting, Guanlun and I split the work of software implementation.
2. I will focus on the LED matrix.
3. Guanlun will focus on the speaker part.
4. After we are done individually, we started to write the entire state machine and the game logic.

1D Meeting 8 (for CheckOff 3):

- Time:

1. 18:00 ~ 6:00, 5th November
-

- Before meeting:

1. I did some research on the the spi board on controlling the matrix, however it doesn't work.
 2. I tried to do the game logic but got some issues on delay implementation.
-

- Goal of meeting:

1. The goal of this meeting with Guanlun is to help with our individual coding part.
 2. We tried to finish the LED matrix part and speaker part.
 3. After that we tried to star with our game logic design.
-

- What we do during the meeting:

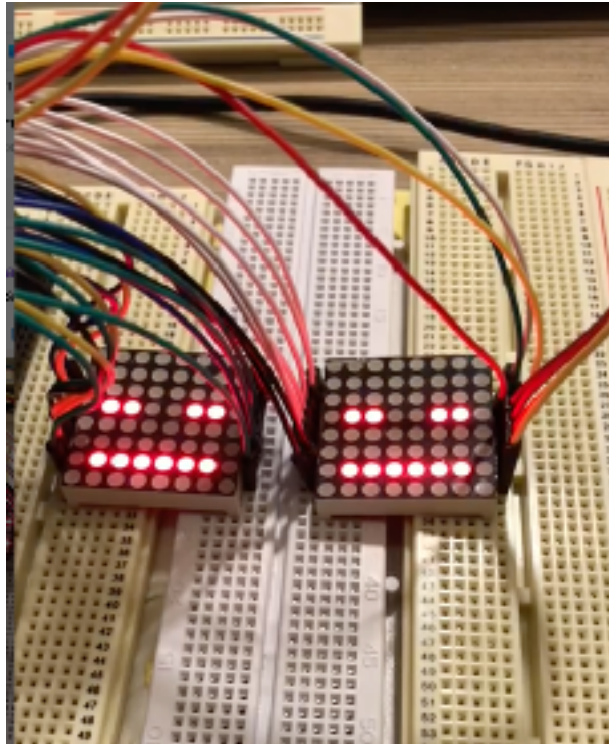
1. During the meeting, we start from the LED matrix.
2. We used 8*8 led matrices to display our notes. Each led matrix has 16 pins which is connected to mojo top. We harked the song notes into a file. In order to accomplish the goals of moving down the notes one by one, we take in the first 8 indexes of the hardcoded matrix at first and move it forward by one after each delay. The delay is accomplished by creating a global d-flip-flop.

```
if(gcounter.q ==0) {  
    mycounter.d =mycounter.q +1;  
}
```

3. To display each frame, we create a function and it take in a list of eight 8-bit binary numbers. And it outputs the row and column values of the led matrix. Each of the row and column output is an 8-bit binary number. The row output is initiated as b00000000, and the initial value of column output is b11111111. For each frame, we display the eight 8-bit numbers row by row. Since the interval is very small, we can see an integrated frame.

```
r =b00000000;  
c =b11111111;  
  
whichrow =counter_r.value;  
r[whichrow] =1;  
for(i =0; i <=7; i ++){  
  
    c[i] =~cr[i][7 -whichrow];  
  
}
```

4. We also meet the design issue that hardcoded file is read from the button to top. It took me quite a long time to find that out.
5. Finally the LED matrix works and we are able to accomplish the LED part.



- What I learned:

1. Always read the lucid official tutorial.
2. You can find everything in the tutorial, the only issue is how to implement it.
3. SPI board implementation of the led metrics can only be used by arduino.
4. Used bread board to test all the circuit, once you solder the parts on the strip board, it is hard to modify.

- To-do list after meeting

1. Although we have finished our LED matrix implementation, it still remains a lot.
2. The biggest issue is the speaker.
3. Then we need to make the speaker and the led matrix work synchronously.
4. We need to design the entire game logic.
5. We need to hardcode a few more songs for both two led metrics and the speaker.
6. We need to implement the buttons.
7. We need to design different delays for our game.

```

//main play state

if(songandspeed.yellowled[2:0] ==b001) {
    temp1 =mysong.song11;
    temp2 =mysong.song12;
} else if(songandspeed.yellowled[2:0] ==b010) {
    temp1 =mysong.song21;
    temp2 =mysong.song22;
} else {
    temp1 =mysong.song11;
    temp2 =mysong.song12;
}

for(j =0; j <8; j ++) {

    cr[j] =temp1[j+mycounter.q];
    cr2[j] =temp2[j +mycounter.q];
}

display.cr =cr;
r =display.r;
c =display.c;

display2.cr =cr2;
r2 =display2.r;
c2 =display2.c;

yellow[2:8] =b888;
green[1:8] =b88;

```

1D Meeting 9 (for CheckOff 3):

- Time:

1. 18:00 ~ 4:00, 6th November

- Before meeting:

1. Before the meeting, we are now able to control the led matrix.
2. Before the meeting, we have hardcoded the first song: Happy Birthday.
3. Before the meeting, we have discussed about the game logic and had a rough idea of how to implement it.

- Goal of meeting:

1. The goal of the meeting is to finalise the circuit and the software implementation.
2. Another goal is to hardcode the other song: Mario.
3. We also need to make the speaker and the led matrix work at the same time.
4. We need to implement the game logic with the buttons, led matrices and the speaker work properly.

5. We also need to make sure everything works so that tomorrow can wrap up and put it in the case.

- What we do during the meeting:

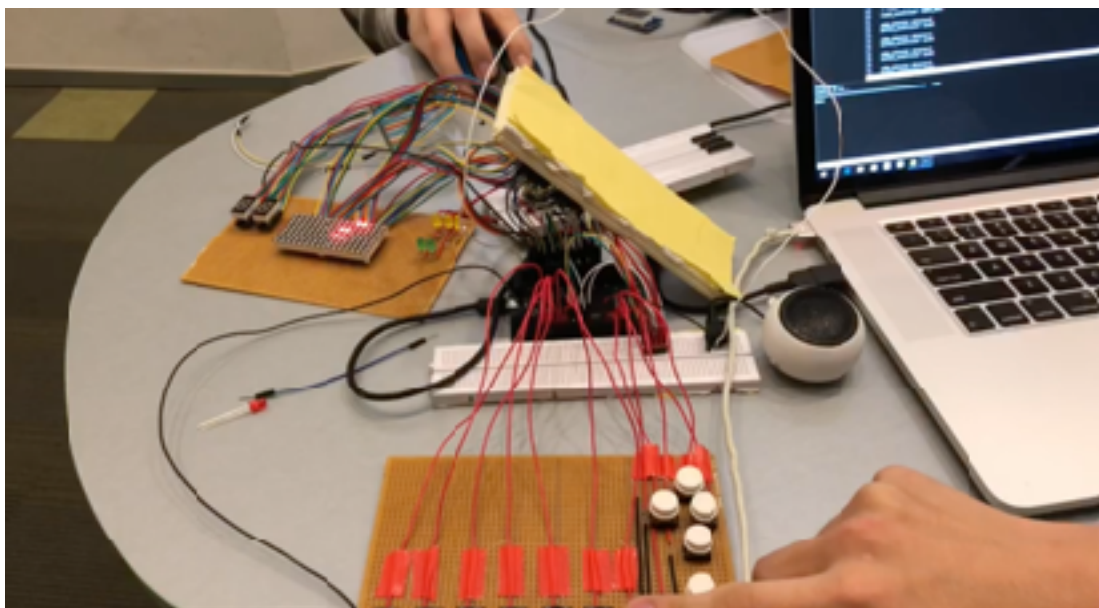
1. Before the meeting, we still have some issues of implementing the speakers. So we solve the speaker issue first.
2. After that, we implement the led matrixes and the speaker to work synchronously. This is a little bit difficult because the speaker that Guanlun implement originally outputs a sequence of notes. In order to work with led matrix, we need to rewrite the function into which only gives one note. So that we can add it into the main function. We did that also.
3. The next thing is to add song selection and speed selection functions into our game. This is not the hardest part and we finish that quickly.
4. Finally the game is almost done, good night guys.

- What I learned:

1. I learnt a lot of the coding part. I took some screenshots of the code below.
2. GitHub link of the code: https://github.com/SanPersie/Piano_Bit_by_Bit.git

- To-do list after meeting

1. We haven't implement the seven segment part.
2. We need to put them in the case.
3. Finalise our game.



```

module multi_seven_seg #(
    DIGITS = 2 : DIGITS > 0,
    DIV = 16 : DIV >= 0
)()
    input clk,                // clock
    input rst,                // reset
    input values [DIGITS][2], // values to show
    output seg [7],           // LED segments
    output sel [DIGITS]       // Digit select
) {

    // number of bits required to store DIGITS-1
    const DIGIT_BITS = $clog2(DIGITS);

    .clk(clk), .rst(rst) {
        counter ctr (#DIV(DIV), #SIZE(DIGIT_BITS), #TOP(DIGITS-1));
    }

    seven_seg seg_dec; // segment decoder
    decoder digit_dec (#WIDTH(DIGIT_BITS)); // digit decoder

    always {
        seg_dec.char = values[ctr.value]; // select the value for the active digit
        seg = seg_dec.segs; // output the decoded value

        digit_dec.in = ctr.value; // decode active digit to one-hot
        sel = digit_dec.out; // output the active digit
    }
}

```

1D Meeting 10

(for CheckOff 3):

- Time:

1. 18:00 ~ 3:00, 7th

November

- Before meeting:

1. Before the meeting, we most of the software
2. We haven't implement

have already implemented part.
the keys.

- Goal of meeting:

1. The first goal of the

meeting is to finish the



software implementation.

2. One is to finish the seven segment part.
3. We also need to connect the keys to the game.
4. Then we finalise the game.

- What we do during the meeting:

1. During the meeting, we firstly finish the seven segment part.
2. Then when we tested the circuit, we found that our buttons have some connection issues. Others are working perfectly.

- What I learned:

1. When hardware got some problems, it is much easier to redesign in the software site. Otherwise, you may screw the whole circuit.
2. Next page I will put the seven segment implementation and the final game box.

- To-do list after meeting

1. To fix the buttons issues.
2. To make a nicer case.

1D Meeting 11 (for final exhibition):

- Time:

1. 18:00 ~ 22:00, 11st November

- Before meeting:

1. Before the meeting, we haven't finish all the software implementation.
2. The buttons still got problems.
3. Some holes of mojo is not stable.

- Goal of meeting:

1. The goal of the meeting is to fix the button issues.
2. Make a nicer case.
3. Test again the holes of mojo top, make sure voltage are going through properly.
4. Finalise the game.

- What we do during the meeting:

1. We tried to disolder the buttons but it still not work.
2. So we changed our software design and change the first two keys as song selection.

```
setSong(songandspeed1.clk1clk1, .rst1rst1), .btn_down(btn_down), .btn_ok(btn_ok), .keys1(key1_store.q), .keys2(key2_store.q));
```

3. END state is added now to display the scores.

```
global_state.END:
    score.d = score.q;
    for(j =0; j <8; j ++ ) {

        cr[j] =0;
        cr2[j] =0;

    }
    display.cr =cr;
    r =display.r;
    c =display.c;

    display2.cr =cr2;
    r2 =display2.r;
    c2 =display2.c;
    yellow[2:0] =b000;
    green[1:0] =b00;
    if (detector6.out){
        global_state.d = global_state.SONG_SELECTION;
    }
```

4. Put the game in a nicer case.
5. We changed several constraints to make sure everything is working properly.

- What I learned:

1. It is better to change software design rather than change the whole hardware.
2. It will screw the whole thing.

- To-do list after meeting

1. Charge the speaker.
2. And good night :)