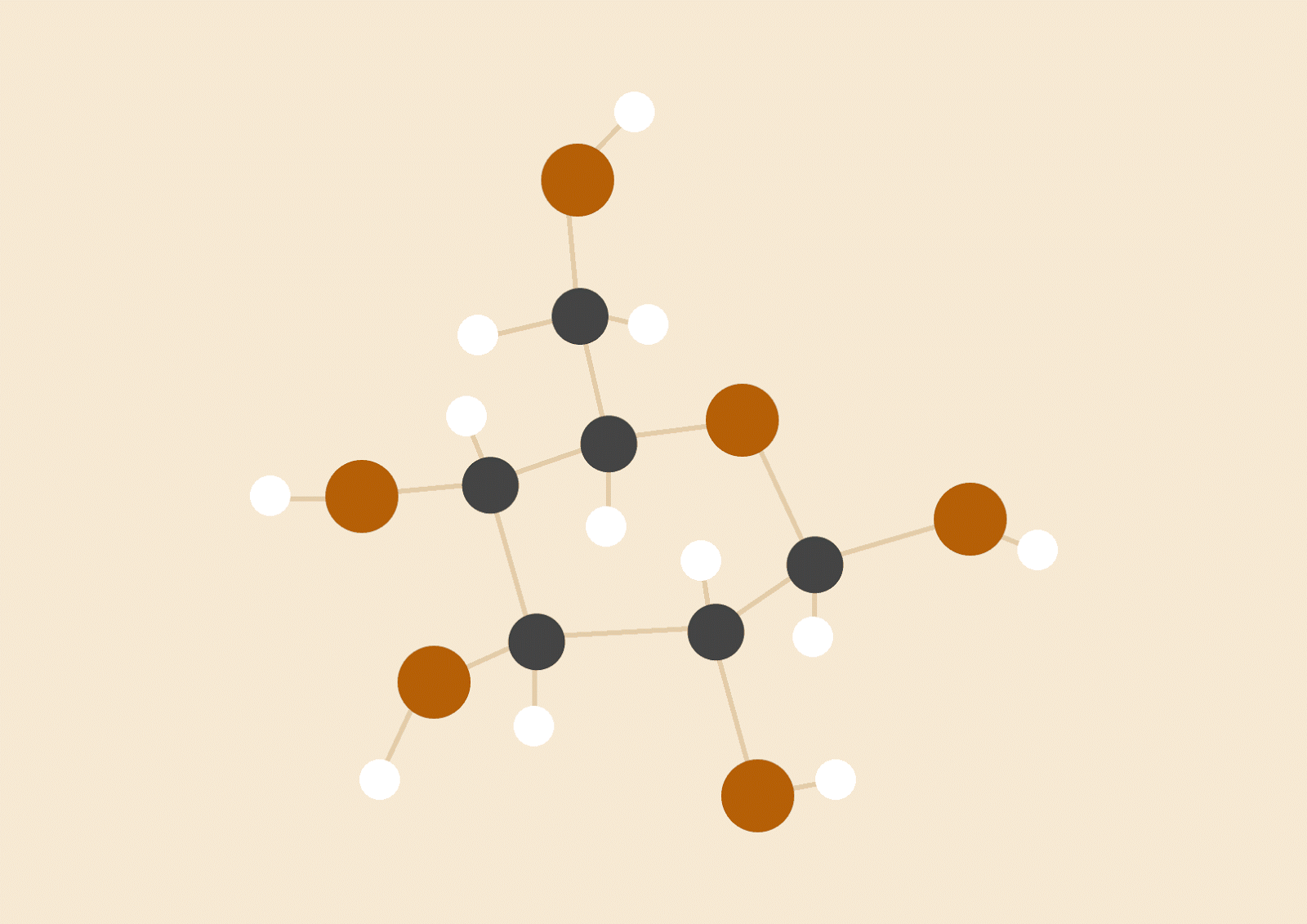
EC535

FINAL PROJECT REPORT



**VIRTUAL PIANO**

TIANYI ZHANG

CHAO MA

# **INTRODUCTION**

A. PROJECT TOPIC

Our project topic is Virtual Piano, which means that what we make is a piano-like musical instrument without real keyboard on it.

B. PROJECT MOTIVATION

The idea of this project is from the music game apps in the smartphone. Those music games imitate different musical instruments on the touch screen of smartphone, which allow players to play music without real musical instrument. Without big and cumbersome instruments, playing music becomes easy and free. Players do not need to carry the musical instruments with them traveling through the heavy traffic. Also, because the apps simplify their products to make them easier to control by green hands. Users do not need to have excellent skills to enjoy the joy of playing music. Therefore, these apps are very popular among people of different ages.

However, these apps have two main drawbacks. One is that the space is restrained due to the size of smartphone, which limited the development of them from games into real simulators for musical instruments practicing. The second is that the apps must be related to smartphones which significantly increase the threshold cost of them. Many people in the world would like to learn to play piano but cannot afford a real piano or a smartphone, but they have the equality to enjoy the pleasure of playing piano. So making a virtual musical device with low prize is the motivation of our project.

The goal of this project is to make a simple portable device with the function of a virtual piano as well as an acceptable prize. It is like playing the piano in the air in front of the device to make different sounds, and the piano on the screen will show which key you are pressing on. Also, the device contains a game mode which shows the bricks to teach you how to play a certain song to allow green hands to enjoy the virtual piano as well.

C. STRUCTURE OVERVIEW

The project mainly contains four parts: input part, control part, sound output part and screen output part. The input part is five IR sensors which detect the inferred signal reflected from the fingers. The control part is the Gumstix board given by the professor. The sound output part is a buzzer which make different sound by changing the frequency through the square waves sent by the Gumstix board. And the screen output part is a LCD Screen to show the animation of play mode and game mode. (The diagram of the structure is shown in Figure 1)

D. RESULTS

We successfully got the signal values of IR sensors for detecting fingers;

We accomplished to make sounds of five different tones (Do, Re, Mi, Fa, So) from the buzzer;

We managed to show the keyboard on the LCD screen and improve it with the additional game mode.

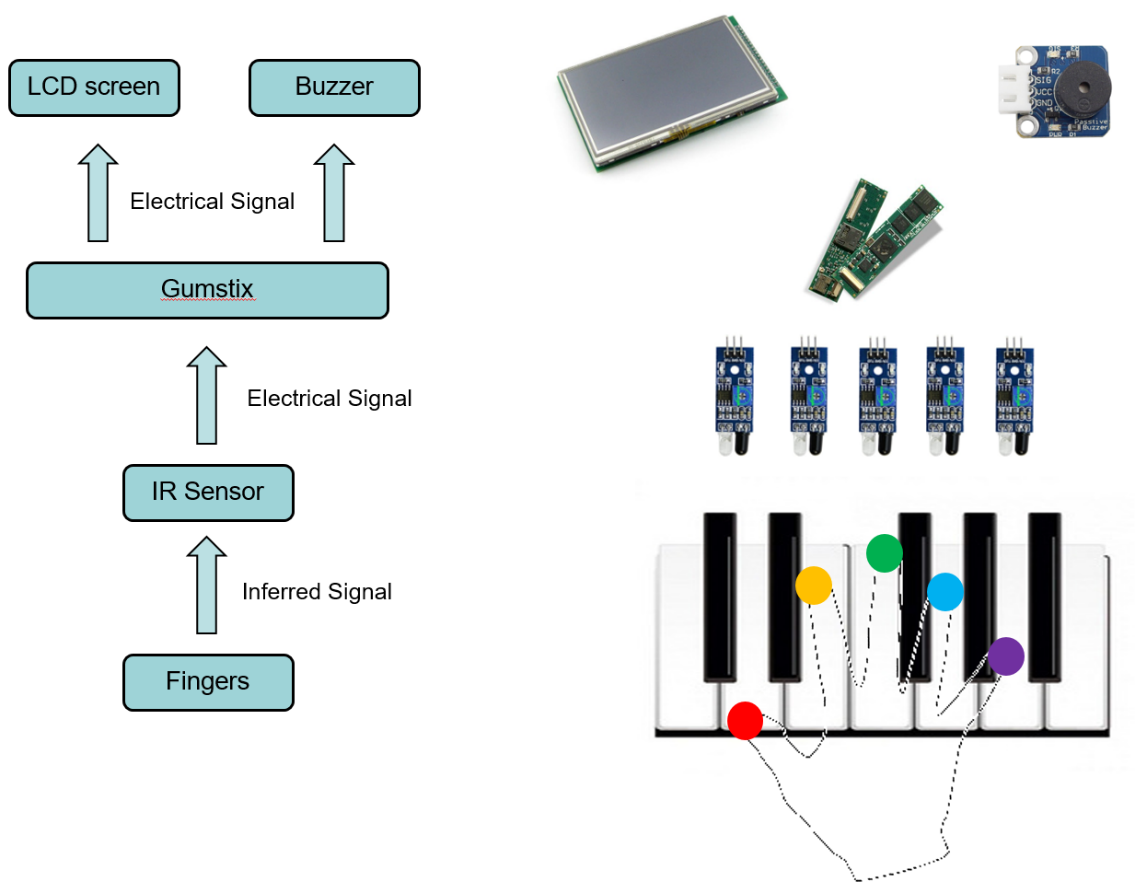


Figure 1. System Structure

# **DESIGN OVERVIEW**

The tasks of this project includes mainly eight parts: 1. Selecting suitable sensor and testing; 2. Machine Learning; 3. Connecting between sensors, buzzer and LCD screen using Gumstix board; 4. Qt coding; 5. Kernel Module Coding; 6. Sending correct square waves to the buzzer; 7. Device packaging. 8. Solving the conflict between the buzzer and the LCD screen;

Since we only have two members in the group. Most tasks were actually done by both of us together in the lab. The first part is the sensor selecting and testing. 1. We selected suitable sensor among three choices: ultrasonic sensor, passive IR sensor and IR sensor. Tianyi tested the ultrasonic sensor and found out that the distance is not accurate in short distance. Chao tested the IR sensor and found out that it is accurate to detect object but can be easily affected by the near objects. 2. The second part is using machine learning to reduce the number of sensors. However, since the unreliability, we gave up on using the machine learning method in the end. 3. The connection is totally handled by the Gumstix board using gpio to get and set value. 4. Qt coding part is mainly done by Tianyi. 5. We coded the Kernel Module together in the lab. Tianyi wrote the function of square wave and Chao wrote the function of getting signal and the main function. 6. Tianyi was in charge of adjusting the frequency of square waves to generate tones. 7. The shell and electric components were assembled and packaged by Chao. 8. The last task was solving the conflict between buzzer and LCD screen. We tried several methods together and solved this problem by allocating different time periods to each of them.

|  |  |
| --- | --- |
| Main Tasks | Principal |
| Sensor Selecting and Testing | Chao Ma |
| Machine Learning | Chao Ma |
| Signal Connection | Tianyi Zhang |
| Qt Coding | Tianyi Zhang |
| Kernel Module Coding | Both |
| Square Wave Sending | Tianyi Zhang |
| Device Packaging | Chao Ma |
| Conflict Solving | Both |

Table 1. Tasks Assignment

# **PROJECT DETAILS**

1. Infrared sensor module

In this project, infrared sensors are used to detect the position of fingers. The working principle of this sensors are shown in figure 2. The infrared generator will give out infrared. When there is an obstacle in front of the sensor, the infrared would be reflected and received by the sensor. Then, the output of this sensor will be low-level voltages. If there doesn’t have any object in front of the sensor, the sensor won’t receive anything and then it will give out a high-level output. One advantage of infrared sensor is that it gives out a non-stop output so when a change is happened, the output will present it rapidly. This guarantees the whole product will make a quick response which provides a good user experience.

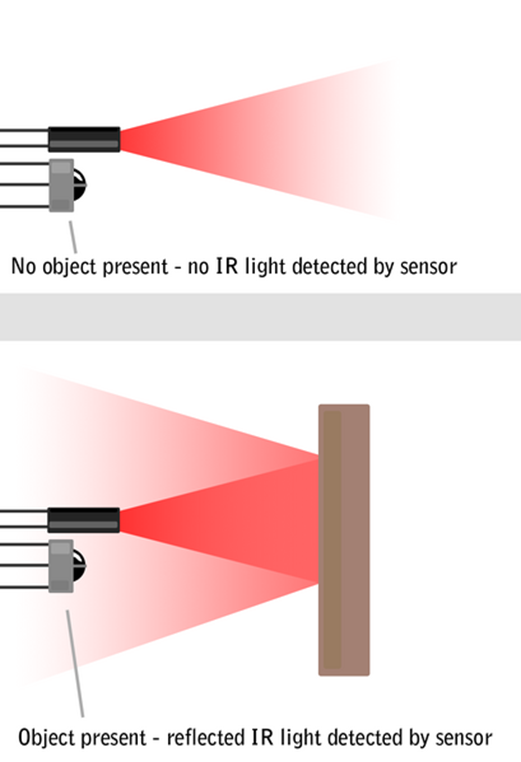
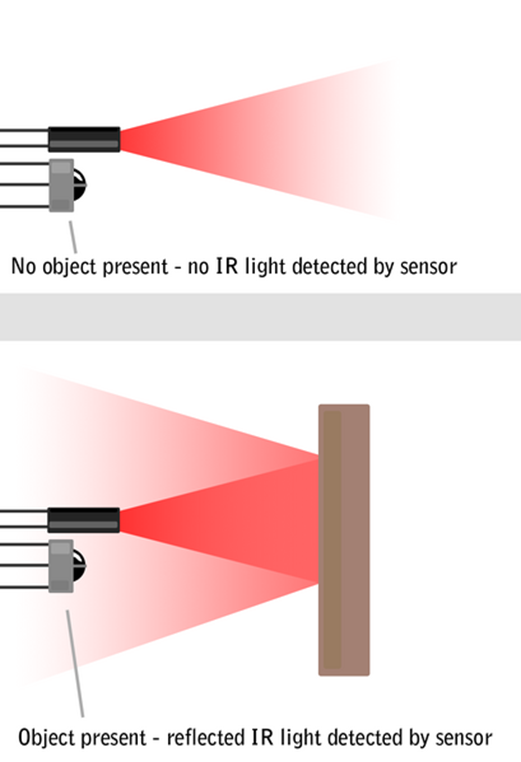


Figure 2. Principle of IR sensor

The detect range of this sensor is 2 ~ 35cm and the angle 35°. This is also the reason why infrared sensor is chosen, as the range we need is shorter than 5cm and the small detect angle makes it’s hardly to detect wrong fingers.

1. Device Driver for Buzzer

A buzzer is applied in the project to give out music. By giving different frequencies of square wave, the buzzer could generate different sounds. Table 2 list the relationship between the tones the project used and the frequencies.

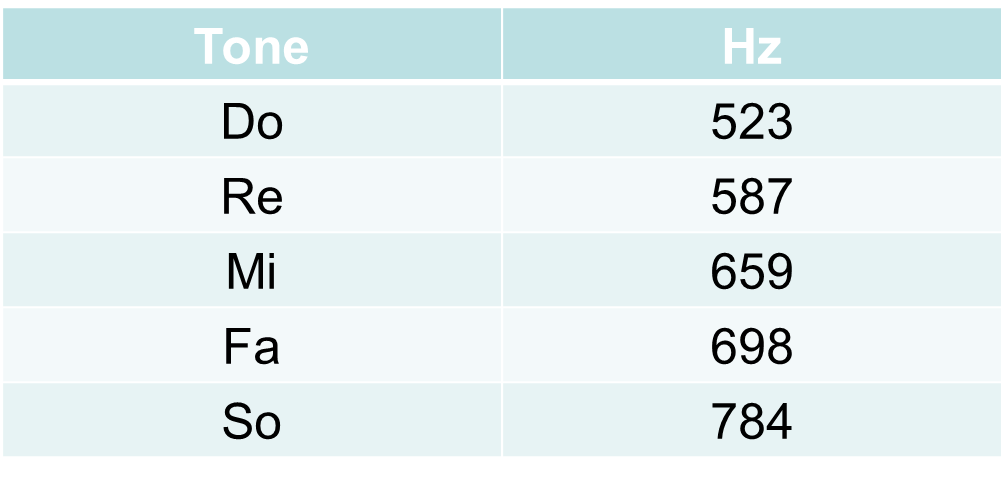


Table 2. Relationship between tones and frequency

Here is an example of how the product give out “Mi” sound. The frequency of this sound is 659hz, so one cycle of the square wave will last 1/659hz=0.001516s. The lasting time of high-level and low-level part will be equal so both of them should be 0.001516s/2=0.000758s. In function music(), we set this cycle to run 65 times. Hence, each time this function is called, it will last 0.001516\*65≈0.1s.

To generate the square waves, we

s=0;

while(s<65){

pxa\_gpio\_set\_value(28,0);

udelay(758);

pxa\_gpio\_set\_value(28,1);

udelay(758);

s++;

}

1. LCD screen (See widget.cpp)

With the help of LCD screen, the product provides two kinds of service. The first one is the play mode. In this mode, users can play whatever they like freely, just like playing real piano, and the LCD will show the which keyboard is pressed. When users pressed “Do”, “Mi”, “Fa”, “So” together, it will go into the game mode. Game mode is designed to help users learn to play music. There will be blocks fall down from the top of the screen and the user need to press the corresponding virtual keyboard when the blocks hit the pattern of the keyboard. It’s a very direct and interesting way for users to learn to play songs. In the prototype, the song is “song of joy”. To help users learn quicker, the LCD screen different colors on the keyboard. If users press the right keyboard in the right time, it will be blue, otherwise it will be yellow. We used green and red color first, however, due to some problems on the gumstix, the LCD screen loses the red part for all the color so we tried different colors and chose two colors that looks like yellow and blue on the LCD screen. The game mode will calculate and show the accuracy during the game and when the game is finished, a comment will be given based on the accuracy. (The according comments are shown in Table 3)

|  |  |
| --- | --- |
| Accuracy | Comment |
| [0%, 60%) | You need more practice. |
| [60%, 80%) | Good. |
| [90%, 98%) | Great! |
| [98%, 100%] | Perfect!! |

Table 3. The relationship between accuracy and comment

1. Circuitry

Figure 3 shows the circuitry of the whole project. In this project, GPIO 118, 117, 16, 29 ,101 are used to read the outputs of the infrared sensors, and Gumstix board connects to the buzzer via GPIO 28. For the power supply, a bread board is used to connect all the VCC and GND pins of infrared sensors to the VCC and GND pins of the Gumstix board. The VCC and GND pins of the buzzer are connected to another pair of VCC and GND pins of the board. This is because if all the VCC and GND pins are connected together, the buzzer will affect the sensor so the sensor cannot work normally.

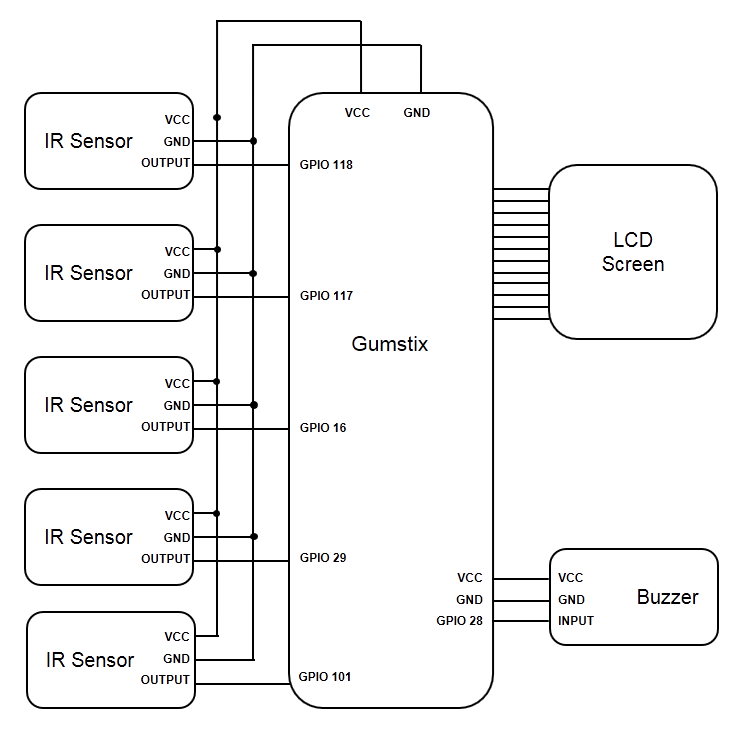


Figure 3. The circuitry of the project

|  |  |
| --- | --- |
| Number | Meaning |
| 9\_ | Play mode |
| 0\_ | Game mode, no block will start to fall |
| 1\_ | Game mode, the block on the top of “Do” will start to fall |
| 2\_ | Game mode, the block on the top of “Re” will start to fall |
| 3\_ | Game mode, the block on the top of “Mi” will start to fall |
| 4\_ | Game mode, the block on the top of “Fa” will start to fall |
| 5\_ | Game mode, the block on the top of “So” will start to fall |
| \_0 | None of the keyboard is pressed |
| \_1 | Show “Do” is pressed |
| \_2 | Show “Re” is pressed |
| \_3 | Show “Mi” is pressed |
| \_4 | Show “Fa” is pressed |
| \_5 | Show “So” is pressed |
| 89 | Game mode end, show the comment of the game |

Table 4. the number used in communication and its meaning

1. Kernel (See mygpio.c)

In this project, kernel is the most important part. It reads the output of the sensors and gives the signals to the LCD screen and the buzzer. Among these communication, the most challenge part is designing the communication between the kernel and the Qt program. Here is how it works:

First, every 0.1 second, by using a timer, the kernel will check the output of the five sensors and record the results. Then, every 0.1 second, the Qt program will read /dev/mygpio. This will run mygpio\_read(). In the read function, it will call another function, music(), to let the buzzer plays the tone and copy a string contains two numbers to /dev/mygpio based on the record of the outputs of the sensors. All the two numbers this program used are listed in table 4.

From these two numbers, the Qt program will know which keyboard is needed to be plotted as pressed and which block will start to fall down. Also, as the fall down of one block will takes 5s, the Qt program can prepare the second number with the first number it got 5s before to find out whether the user presses the right keyboard or not.

1. Changes from the original idea

This part will introduce why the prototype is a little different from the original idea and why the changes are made.

The first change is the cancel of neural network. Applying neural network was a good idea as it will achieve using three sensors to detect the positions of five fingers. We first collected the outputs of three sensors and the positions of fingers together. Then, by using TensorFlow, a machine learning software, we tried to find the relationship between these two things so that when getting the outputs of three sensors, we are able to predict the positions of five fingers. However, the result of the training was far from satisfying. First, the insufficient number of samples only brings a high error rate. Second, the sensors themselves have error rates, which makes the performance of this methods went worse when we tested it on our prototype. Because this is an instrument, an error rate over 10% is unacceptable, as a song has at least 50 notes and it will be total disaster if 5 of 50 notes go wrong. Therefore, realizing that there is no hope to reduce the error to be lower than 10%, we abandon the idea of using machine learning.

The second change is the cancel of the projection. The primitive idea of our product was using a flashlight to project a pattern of piano keyboard. However, as we cancelled using machine learning, we paid more attentions on the LCD screen to add a game mode on it, so it plays a more and more important role in the whole project. Therefore, because the LCD screen already could show the pattern of piano keyboard and is more interesting, we stopped using the flashlight to let the user keep their eyes on the screen. Also, in the prototype, adding the flashlight will make the height of the whole product becomes twice times than before. As the product is designed to be portable, it’s not wise to add the flashlight.

# **SUMMARY**

The project is a virtual piano. It is a simple portable device with the function of a virtual piano as well as an acceptable prize. The electronic devices we use are IR sensor, Gumstix board, buzzer and LCD board.

Accomplishments:

1. We successfully got the signal values of IR sensors for detecting fingers;
2. We accomplished to make sounds of five different tones (Do, Re, Mi, Fa, So) from the buzzer;
3. We managed to show the keyboard on the LCD screen and improve it with the additional game mode.

The remaining challenges are as following:

1. Replacing the buzzer with a speaker to improve the sound performance;
2. Adding projection from the screen to the front to make sure players can press on the right position;
3. Improving the device packing to avoid the effect of daylight.

# **REFERENCES**

[1] Kristen G. *Communication via Eye Blinks Detection and Duration Analysis in Real Time* [J]. IEEE, 2001.

[2] Michael C and Margrit B. *Real Time Eye Tracking and Blink Detection with USB Cameras* [M]*. Boston University Computer Science Technical Report*, 2005.

[3] [Sanyam G.](https://www.blogger.com/profile/10106518971387961191) (2016, January 18). *Codes and Musings of Sanyam Garg*. Retrieved from <http://sanyamgarg.blogspot.com/>

# [4] *Face Detection using Haar* Cascades. Retrieved from <http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html>

[5] Qifan H. (2017, April 20). *Median Filter.* Retrieved from

<https://github.com/QifanHe/ROS_bebop_faceavoid/tree/master/face_detect/nodes>