FYP Interim Report - Bryan

by EEE TAN ZHEN YE, BRYAN

School of Electrical and Electronic Engineering

FYP Interim Report

Submitted by: Bryan Tan

Matriculation Number: U1722930H

1. Introduction

The purpose of this project is to create an interactive display that can be used to showcase what software and hardware can do together. The interactive display should be something interesting and unique. Thinking along these lines whilst getting inspiration from the game “Pong” and Pinball, my idea was to build a physical 2-player pinball-like game, with 1 “human” player and the other, the computer.

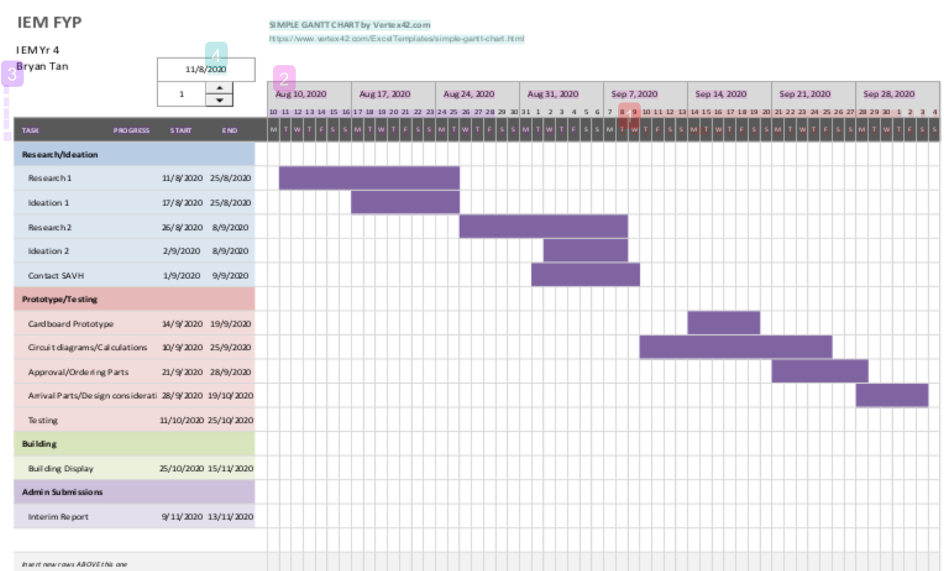
1.1 Scope

The project consists of a physical game board that the player controls from one side. The score can be tracked on a scoreboard mounted beside the board itself. The opponent is the computer who is able to know when to hit the ball based on various sensors placed at different locations on the gameboard.

1.2 Objective

The objective is to create a physical 2 player game in which one side is controlled by a human and the other by the computer.

1.3 Timeline



TimeSchedule.xlsx

Table 1: Double click icon to scroll through in Excel

1.4 Interim Report Purpose

The purpose is to provide the reader with information on what the project is and how it is progressing. It also covers the work conducted, including the issues faced and the future work to be completed before the end of the project.

2. Work Conducted

2.1 Work Completed

The work completed can be broken down into 4 main components. Testing of the flipper mechanism, building the gameplay board, miscellaneous testing and object tracking. All the components were tested on an Arduino board. An Arduino board was chosen as most modules and sensors are easily interfaceable with Arduino. Besides the modules, there are also many variants of Arduino boards, which means that should the project have a need to be used with Wi-Fi, the current board can easily swapped out for one that is Wi-Fi capable.

2.1.1 Testing of flipper mechanism

Typical flipper mechanism makes use of a specialized solenoid with 2 coils around the same core [1]. One coil is for the flipper to be strong enough to hit the ball while the other is to hold the ball in place should the player press and hold the flipper button. It is necessary to have 2 coils as powering the stronger coil for a long period of time will cause the coil to become burnt out. The design of the flipper is very specialized hence it is not possible to use the same mechanism in this project. This reduced the choice of mechanisms to be either a DC motor with an encoder or a servo motor. A DC motor with an encoder will allow the DC motor to have precise positions set, which allows the flippers to have a set rotation [2]. However, with the flipper having such a small range of motion, it was not be necessary to use a DC motor. Using a servo motor as a flipper thus became the final choice.

The SG90 micro servo was chosen. It has an operating voltage of 5V with a running current of about 220mA and stall current of 650mA [3]. To ensure it does not reach the stall current, two aspects were carefully considered. Firstly, the ball chosen had to be light enough and yet, not much affected by air currents. It should also be solid enough to move a decent distance when hit by the flipper. Secondly, the movement of the flippers, when hitting the ball, should not hit its maximum of +-90 degrees. Therefore, the total current needed for the 4 flippers, should all of them be moving simultaneously, is 220mA*4 giving a total of 880mA.

Figure 1: 2 Flippers on each side

A typical configuration would be to power the servos separately and have each servo connected to one pin on the Arduino. This would use up 4 pins on the board. More pins will be needed if I used more servos. To reduce the number of pins needed for all the servos, a PCA9685 I2C interface driver was chosen.

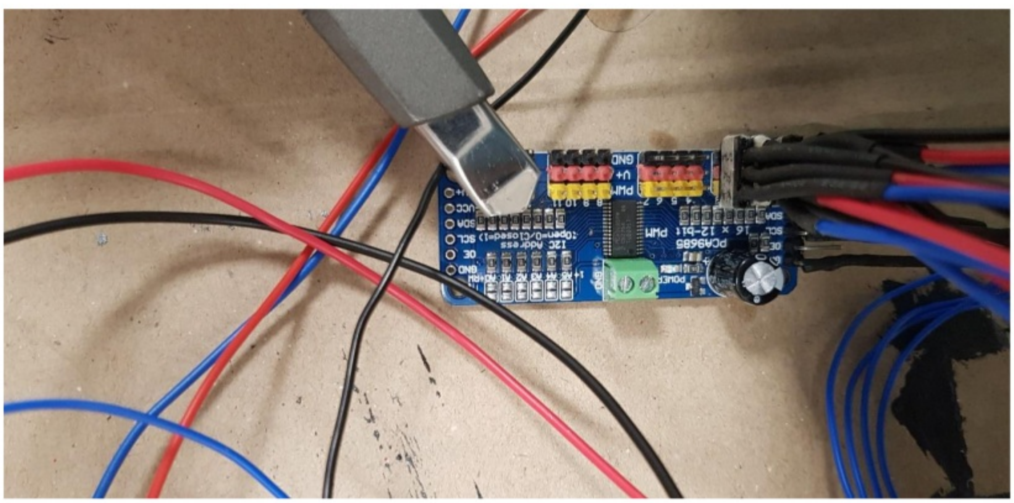


Figure 2: PCA9685 board

The PCA9685 driver can drive up to 16 servos with only 2 pins, through the usage of the I2C communication protocol. The 2 pins being the serial data (SDA) and the serial clock (SCL) on the Arduino board. Another reason for using the PCA9685 is to allow other modules using I2C protocol to chain with this and reduce the need for another 2 communication pins [4].

2.1.2 Building the gameplay board

The gameplay board is constructed to be easily portable thus the main frame of it is made of extruded foam. This makes the entire structure lightweight yet strong enough to hold the weight of the various components mounted on it. The selection of balls was another item taken into consideration. 4 balls of varying size and materials were taken into consideration. Out of the 4, the medium sized plastic ball was chosen as it did not stick to the board due to static and was not too oversized.

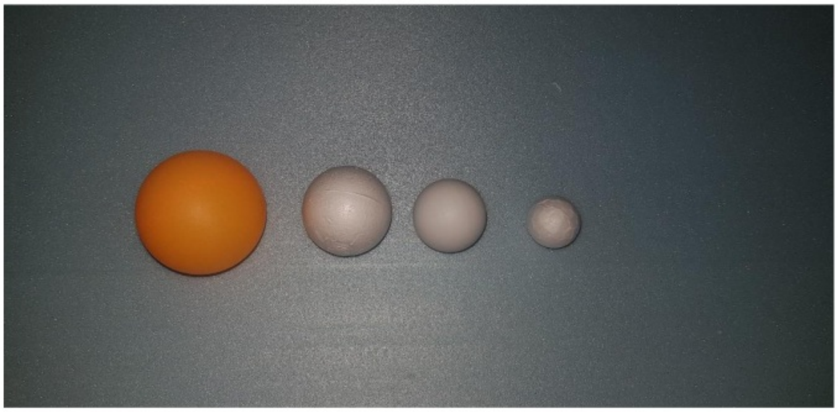


Figure 3: Type of balls considered, 3rd from the left is the chosen ball

Testing was also done to determine the angle at which the board should be at to allow the ball to slide down at a reasonable speed and not get stuck. This was determined to be an angle of 2.15 degrees. To achieve this, blocks of different heights were used as legs.

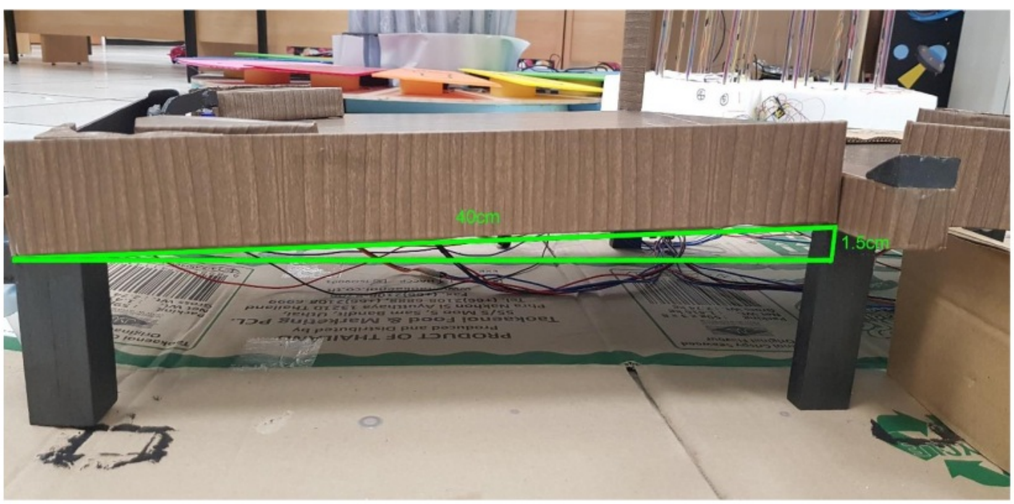


Figure 4:Gameplay board

To achieve an angle of 2.15 degrees, the following calculation was done, $\sin^{-1}(\frac{1.5}{40})$,meaning that the height difference of the front and back of the board had to be 1.5cm while the board was to be 40cm long. Having this angle, the speed of the medium plastic ball works out to be on an average of 0.2m/s.

2.1.3 Miscellaneous testing

Other than the main parts of the flippers, there were 2 other components that were tested and used. These are, infrared sensors and a 4-digit 7 segment LED. The infrared sensors are used to count the scores when the ball passes the flippers. The infrared sensors are a repurposed line tracer module which work digitally and activate when an object is in front of it.

Figure 5: Infrared sensor

The 4-digit 7 segment LED is used to show the scores for each side of the board. To reduce the number of pins needed, the 4-digit 7 segment LED is controlled by a TM1637 driver. This driver allows the control of all 4 digits using only 2 pins with the two-wire bus interface. For all intents and purposes, this is the same as the I2C protocol mentioned earlier except it does not have a slave address so it cannot be chained with the I2C modules [5].

2.1.4 Object Tracking

Much research was done on the best way to achieve object tracking. The solutions ranged from using a Raspberry Pi with OpenCV to using multiple range sensors, either ultrasonic or infrared, to triangulate the position of the ball. However, after much testing, it was determined that the triangulation method was too inaccurate as the infrared light was interfering with each other and using an additional Raspberry Pi just to do object detection with OpenCV was deemed not cost effective. Therefore, to achieve object tracking, a chip with the TinyYOLOv2 model was used. TinyYOLOv2 takes an image of size (416 x 416) and outputs the class probabilities and bounding box coordinates [6]. The original YOLOv2 uses 24 convolutional layers while the TinyYOLOv2 uses 9 convolutional layers and 6 max-pooling layers.

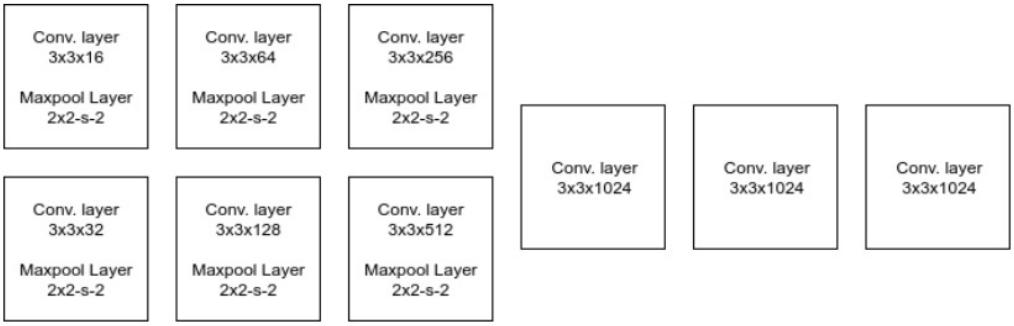


Figure 6: TinyYOLOv2 Architecture

This means that while the original is more accurate, the TinyYOLOv2 is faster and takes up less processing power. The model is put on a Kendryte K210 processor which allows for ease of use. The processor is then integrated with a camera and produced as an entire module called "HuskyLens". The HuskyLens is able to communicate via either UART or I2C with the Arduino. As mentioned previously, to reduce the number of ports needed, the HuskyLens will be chained with the PCA9685 using the I2C ports. This will be used to detect the ball and determine when to use the flippers for the side controlled by the computer.

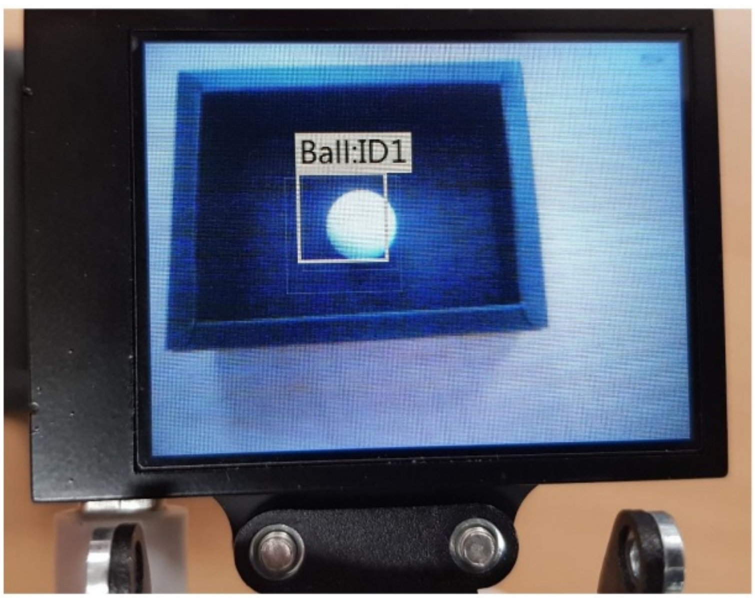


Figure 7: Tracking of the ball

3. Problems Encountered

Being unfamiliar with all the various protocols used for communications with the Arduino and component, I had to do a lot of research on the benefits of each protocol. These protocols ranged from UART, I2C, Serial over wire, etc. There was also a need to source for a driver for the servos as running 4 servos directly from the Arduino board will cause it to burn out due to the required current being over the limit of what it can take. There was also issues with regards to the object tracking model as I was completely new to object tracking and AI. Lastly, was the non-technical issues like the consideration of the materials, whether they were smooth enough for the ball to roll and whether it was strong enough to be moved without breaking among others.

4. Future Work

The main structure of the board, the flippers and the scoreboard have been completed. What is left is the obstacles on the board, programming the computer based on the object tracking and improving the aesthetics. Currently, no changes have been made to the original requirements of the project.

5. Conclusion

Based on the proposed timeline, the project is slightly behind time. This was due to a couple of factors, namely, the long shipping time of the parts and unfamiliarity of the parts used. This meant that I had to spend more time researching and testing out the parts before incorporating them into the project. However, with regards to the challenges faced, I think the delays were reasonable.

References

[1] M. Gibson, "Fun With Pinball," Fun With Pinball, n.d. n.d. Circa. 2020. [Online]. Available: https://www.funwithpinball.com/learn/flippers-coils-and-power. [Accessed 30 October 2020].

[2] M. Murray, "The Geek Pub," The Geek Pub, 15 December 2019. [Online]. Available: https://www.thegeekpub.com/245407/how-rotary-encoders-work-electronics-basics/. [Accessed 2020 October 2020].

[3] Opencircuit, "Opencircuit," n.d.. [Online]. Available: https://opencircuit.shop/Product/TowerPro-SG90-9G-micro-servo-motor-180. [Accessed 30 October 2020].

[4] SFUptownMaker, "I2C," [Online]. Available: https://learn.sparkfun.com/tutorials/i2c. [Accessed 2 November 2020].

[5] Titan Micro Electronics, "LED Drive Control Special Circuit," 2015.

[6] Y. J. Wai, Z. b. M. Yussof, S. I. b. Salim and L. K. Chuan, "Fixed Point Implementation of Tiny-Yolo-v2 using OpenCL on FPGA," *International Journal of Advanced Computer Science and Applications,* vol. 9, no. 10, pp. 506-512, 2018.

FYP Interim Report - Bryan

ORIGINALITY REPORT

7%	5%	0%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to asiapactech Student Paper	3%
2	Submitted to Queensland University of Technology Student Paper	2%
3	Submitted to Georgetown University Student Paper	1%
4	Submitted to Southern New Hampshire University - Continuing Education Student Paper	1%

Exclude quotes	Off	Exclude matches	< 15 words
Exclude bibliography	On		