FlipAble: An Assistive Technology Board Game

Final Report

Team Members

Beatrice-Cristina Bezdadea

Lucille Cazenave

Amira El Fekih

Linnea Evanson

Gionieva Fraser

Emmanuel Gkigkilinis

Charalambos Hadjipanayi

Maksim Lavrov

Marta Mambrilla Velazquez

Wesley Zuzarte

Supervisor: Dr. Ian Radcliffe

Revision	Date	Author	Description
A	17/06/2018	FlipAble Group	First release

18 June 2018

Word Count: 17739

ABSTRACT

FlipAble is an assistive technology board game which enables children with high level motor control disabilities to play independently alongside their able-bodied peers. The target users of this project are the students at John Chilton School.

The device developed is a four-player pinball-type game where players try to defend their goal using a pair of flippers. To control their flippers, players have the option of several different control technologies: Word Detection (WD) Control, a Pushbutton, and an Eyebrow control unit. The Word Detection Control uses the Snowboy word detection software, triggered by the hotword 'go', to activate the flippers. The Pushbutton is a familiar device to the users as it is commonly used at the school. The Eyebrow control unit is composed of a Mechanomyogram (MMG) sensor and an Inertial Measurement Unit (IMU). Together they detect when the player raises their eyebrow to activate their flippers. The flippers are activated by a central processing unit which allows current to flow through pinball-standard high voltage coils, powered by a central power supply. These function as solenoids and pull a plunger which induces the flipper bat rotation through a plastic coupling part.

The board game is supported by a metal chassis with electronics enclosed beneath a play field and with a ball return mechanism on each side. The playing field and the outer walls of the game are made of medium-density fibreboard (MDF). The ball return mechanism is necessary to automatically return the ball to the field once a goal is scored, allowing the students to play without an able-bodied assistant. Audio and visual feedback is necessary to include visually and hearing-impaired children. This is achieved through elements such as a score display LCD screen, LEDs which light up to signify the concession of a goal, and several sounds which signal the start of the game, the activation of the flippers, receiving a goal, and the end of the game.

The main limitation of the design is the low power of the flipper bat when it strikes a ball, which results in dead spots on the playing field as the balls do not have sufficient velocity to reach another side. One way to resolve this issue in future iterations of the game is to use an EOS switch in the flipper assembly circuitry which would supply more power to each solenoid, generating higher pulling forces. The game flow is also limited by the delay response of the controls and by their activation success rate. Word Detection control has the longest delay (730 ms) and the lowest activation success rate (72%). The prototype could be improved by adding a waiting period in the Arduino codes of the Eyebrow control unit and the Pushbutton, to compensate for the larger latency period of the Word Detection Control. The accuracy of the Word Detection Control unit should also be increased by training the hotword model on the Snowboy platform with more children from the school. This would make the game fair for all players.

This product is one of the first of its kind in the field of toys using assistive technology, as no similar prototypes were found on the market.

Contents

Introduction

One of the most important aspects of a child's development is social interaction with their peers. This is sometimes difficult for differently abled children, who may be excluded many times from group entertainment due to congenital challenges. Many board games on the market require the player to roll a dice or flip a card which can be quite difficult for some students to do without external help, thereby limiting their independence. This is an issue faced by the students of John Chilton School, which caters to children with high level motor and mental disabilities. To address the issue, an assistive-technology board game, given the name "FlipAble", was developed. The game allows the disabled children at John Chilton School to play independently alongside their able-bodied peers. The key to the development of an adaptive game is the availability of different control units, to provide alternative controls in case an individual is unable to use one. FlipAble is a four-player pinball-type game with a range of controls such as Word Detection, Eyebrow Control and a Pushbutton. Each control is plugged into one side of the board game and when the child activates a control, the flippers on the corresponding side of the board move to push a ball around the playfield. The object of the game is to act as 'goalkeeper', preventing a ball from entering your goal. The player with the least number of goals is declared the winner.

1.1 Aims

The aim of this project is to manufacture an autonomous and multiplayer board game that allows children with all ranges of motor disabilities to experience the joy of playing an interactive game with their friends and family. An important aspect of this game is that the children should be able to play without assistance from teachers or others at the school. Thus, the only inputs required to maintain game flow should be the control units. Specifically, the aim is to develop a four-player pinball-type game with different control types to suit different disabilities.

1.2. Objectives

The aim was achieved by dividing the project into specific objectives seen in Table 1 below. These objectives were then organised into a Gantt chart and the members of the group were arranged to tackle certain areas (see Project Management, Appendix A).

Table 1 - Objectives

Control objectives:	
1	To implement multiple control systems: Word detection, a Pushbutton, Eyebrow control
2	To implement time delays for certain controls to maintain a fair environment among different player abilities
Structural objectives:	
3	To develop a play surface that allows for constant ball movement and a dynamic gameplay
4	To make the dimensions of the game suitable for a person in a wheelchair
5	To build a chassis to provide mechanical support and stability to the game
Autonomic objectives:	
6	To implement a ball return mechanism to ensure the game is autonomous and at least one ball is constantly in play
7	To develop a flipper mechanism with enough force to propel the ball
Feedback objectives:	
8	To implement sound and visual feedback
9	To implement a goal counting mechanism and a score display for each player
Safety objectives:	
10	To ensure electrical and mechanical safety to prevent any harm to the users and minimize accidents

2. Requirement Definition

2.1 Background Research

Several fun and interactive children's board games already exist on the market and could have been adapted to suit the differently abled youth of John Chilton School, a school in London for children with severe learning disabilities. Games such as Hungry Hungry Hippos, Connect Four and Chess are all very popular games [1]. Selecting the right game took two main criteria into account: the requirement for inclusivity (it had to suit any age from 11 to 17 years old) and the requirement for autonomy (eliminating the need for assistance when operating the game). An ambitious choice that allowed for the fulfilment of both requirements involved the design and creation of a unique four player pin ball type game. Having multiple players adds a social aspect to the game, while being a dynamic game requires speed and focus, thus making it very interactive for all ages.

The idea was presented to the school at an early visit and confirmed the popularity of the choice with the children. A visit to John Chilton School was organized to obtain first-hand knowledge of the most common disabilities. There, the interaction with the children allowed for a better understanding of what they desired the most in a board game. After speaking to Ms. Jane Hales, the Head of Physical Education at John Chilton School, it was determined that the most common disabilities involved speech impairment, lack of fine motor skills, and lower limb paralysis. Although there are a host of assistive technologies for individuals with motor control challenges, the control systems already in use at the school, according to Ms Hales, include push buttons, eye gaze control, and touchscreens. The chosen devices for the game had to be user-friendly, inclusive for all kids and also suit the gameplay, which requires fast repetitive action and a clear vision of the board. It was important that the controls, for example a head switch, did not block vision and did not put a strain on the kids when activated repetitively. While doing some intensive research, some other common assistive devices were found to be a mouth stick, an eye gaze tracker, a head wand, a sip and puff switch and an oversized trackball mouse, among others. A few have been compared and the results can be seen in Table 2. The controls selected for this project were word detection control, a push button, and eyebrow control [2] since they achieved all the criteria mentioned and allowed for the most interactivity.

Table 2 - Controls comparison

Control System	Quick description	Advantages	Disadvantages
Word Detection	Controlled by the human word detection and responds to spoken commands	- Does not require any physical movement apart from jaw movement - Hands-free - Wide info spectrum - User friendly - Affordable - Microprocessor compatible modules available (see Table 3)	- Background noise [3] - Different pronunciation/voice [3] - Time delay during recognition [3] - Might need training [3]
Eye Gaze Tracker	Tracks the movement of point gaze	- User friendly - Quick response	- Accurate trackers are intrusive and expensive [4] - Generally requires calibration [4] - Requires user to look at board or electronic device all the time

MMG (mechano-myography)	Measure of muscle surface vibration (microphone) produced by the muscle contraction	- Can aid in rehabilitation - Considerably accurate - Small delay - Placement of sensors not required to be precise or specific [5] - Unaffected by changes in skin impedance [6] - Signals can be used to detect the degree of muscle contraction [7]	- Still not a well-known technology - Difficult implementation, needs an IMU (Inertial Measurement Unit)
Puff/Blow Sensors	Sensors in the mouth detecting the air pressure	- Easy implementation - Affordable	- Slow [8] - Users might become out of breath - Needs regular cleaning [8] - Limited adaptability and flexibility [8]
EMG (electro-myography)	Detects and measures the electrical impulses produced by muscle cells	- Non-invasive technique - Sensors readily available - Consistency [9] - Full range and high resolution [9] - Considerably accurate - Widely accepted	- Takes time to calibrate for each individual [8] - Requires low noise [5] - Requires stable signal component [5] - Electrodes are expensive - Error Prone [8] - Require highly specialized hardware and processing algorithms [8] - A gel needs to be applied on the electrodes every time
EEG (electro-encephalography)	Detects electrical impulses transmitted between cells in brain	- Enhances concentration - Affordable [10] - Excellent time resolution [10] - Successfully implemented before [11]	- Not applicable for games with multiple instructions - Poor spatial resolution [10]
Push button	Button activated when pressed	- Provided by the school - User friendly - Easy implementation - Versatile (can be pressed by hand, feet, head)	- Does not target some of the disabilities of the children in the school

Various word detection platforms were explored and the assets and drawbacks of each were compared. The following table summarizes the different approaches considered.

Table 3 – Comparison of different approaches for word detection control

Option	Advantage	Disadvantage
Raspberry Pi + "Snowboy" online Software	Simple to use, readily available documentation	Keyword model needs to be trained by a group of people
Arduino + EasyVR Shield	Readily available documentation	High price: £50
Raspberry Pi + BitVoicer software + Arduino Control	Readily available documentation, easy interface with rest of game if Arduino is used	Complicated design, BitVoicer operates on Windows OS

Due to its widely and readily available documentation and since the Snowboy [12] word detection platform has been utilized before by past students of the Imperial College London Bioengineering department [11], it was chosen as the most suitable software to implement the word detection control module.

It was also considered important that each control could be plugged in on any side of the game. Given that the controls chosen had various types of connections and given the advice of our Electronics Technician, Mr Paschal Egan, it was determined that the use of an Arduino Nano to allow USB connection to the game was the best choice given our resources.

Games requiring balls to return to the field usually have a system of returning them into a pocket where the player manually picks them up and places them back on the playfield. An automated version of this mechanism was not found in any board game. The aim of this project is to allow players with motor disabilities to be able to play independently, which is why a ball return mechanism had to be designed to release the ball back onto the playfield. To achieve this, multiple designs have been studied, like the conveyor belt [13] [14], the on and off ramp escalator [15], an elevator and an idea of a wheel with compartments. The latter was chosen for implementation after considering the space allowed for it in the mechanical structure of the game, the budget and the time to manufacture.

Ideas on implementing ball counting mechanisms were investigated and the two main approaches involved pressure plates and infrared sensors. The latter was adopted due to its wide and readily available documentation [16] [17], cost efficiency and its use in motion detection.

2.2 User Requirements

The target users for this project are the students at John Chilton School who all have special needs. Some of these children have high level motor and mental disabilities. Several members of the group visited the school and met with staff and students to gain an understanding of their physical disabilities as well as their favourite types of games. From this meeting the user requirements were defined as follows:

1. **Variety of controls:** multiple types of control systems that use different body parts to suit a range of disabilities needed to be implemented.
2. **Control usability:** each control system must be easy to use and operate by children. They must be comfortable to use for the duration of the game.
3. **Control homogeneity:** the choice of control system should not disadvantage a player.
4. **Game response:** fast response to a signal from a player.
5. **Dynamic game play:** smooth game flow so there are no frustrating stalls in the game.
6. **Autonomous game:** no input should be required from the users during play except for their use of controls.
7. **Short set-up time:** the controls should be easily connected to the game.

8. **Visibility of playing field:** visually impaired players should also be able to follow the game.
9. **Audio-visual feedback:** the game should be interactive and stimulating for both the visually impaired and the hearing-impaired students.
10. **Size and weight:** the game should be large enough to allow a student in a wheelchair to sit on any side. It should be light enough to be transported and stored easily.
11. **Safety and security:** the students should not be at risk of electrical or mechanical harm.

2.3 Technical Requirements

2.3.1 Functionality and Performance

1. The time delay between detection of an input signal and flipper action should be less than 1 second.

2. The number of goals scored should be updated on the screen within 5 seconds of the goal being scored.

3. Audio and visual feedback should be observed within 2 seconds of the corresponding action.

4. The playing field should be made of robust material that can withstand ball impacts, with reinforced areas such as walls where impacts are more frequent and powerful.

2.3.2 Size and Weight

1. Weight of the game should be less than 12 kg (Standard weight for safe manual lifting, according to the Bureau of Labor Standards of the U.S. Department of Labor)

2. The board should have sides of 70 cm (Average width of motorized wheel chairs [16])

3. The board should be no larger than a generic table top (Average size of table tops ~100 x 70 cm [18])

2.3.3 Usability, Interface and Ergonomics

1. The connection for the control units should be standardised and the same on each side of the board.

2. There should be high colour contrast between the ball and the board for the visually impaired students.

3. The outer wall of the game should be transparent in front of the goals to maximise visibility.

4. The response time of each control (between input signal detection and flipper action) should be equal for all controls.

2.3.4 Safety and Security

1. The game should conform to the standards for children's toy safety: Consumer Product Safety commission [19] or European Union Harmonized Standards [20], [21]

2. The board should not have sharp edges or abrasive surfaces.

3. The game should conform to the standards from the International Electrotechnical Commission (IEC).

4. The electrical components should be properly contained away from user access.

5. The electrical components should be properly grounded and appropriately sized fuses used.

6. The electrical components should be insulated to prevent discomfort to the user due to heat dissipation, and to reduce flammability.

2.3.5 Life Reliability and Maintenance

1. The game should not contain components which need frequent replacement (such as batteries).

2. The game should last at least 2 years under normal use and proper maintenance.

2.3.6 Legal and Regulatory

1. The audio feedback from the game must be no more than 60 dB at distance 30 cm from the sound source. Sixty decibels is the level of a typical conversation according to American Speech-Language-Hearing Association

2. The game should conform to the standards for children's toy safety, as stated in section 2.3.4

3. Final Design

3.1 General design

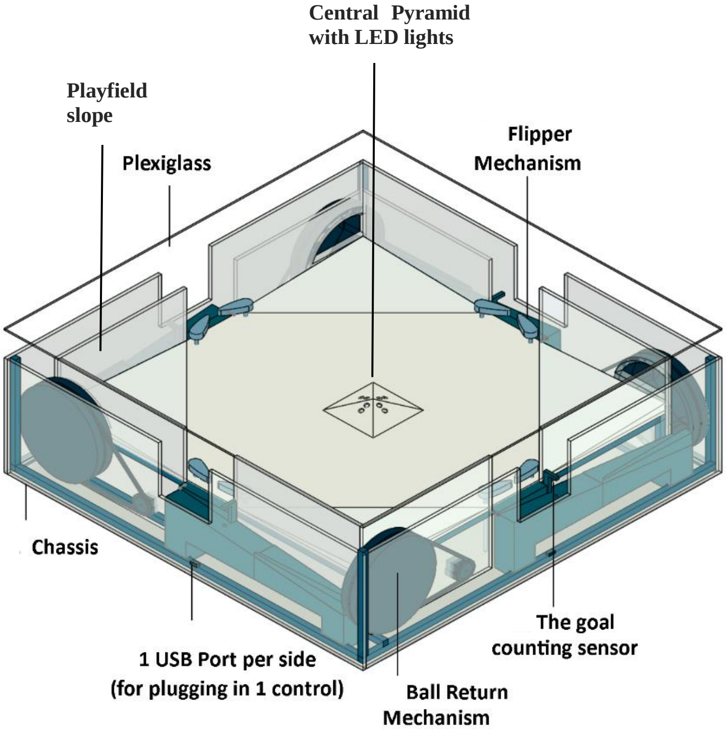


Figure 1 – Overview of final design (outside dimensions: 700x700x200mm with 100mm width goal)

The overview of the design is shown in Figure 1 above. The playing field is composed of a level surface, four 3D printed corner slopes which aid in the return of the ball and a central pyramid with the same purpose and the added function of visual feedback through embedded LEDs. The walls are cut at the goal to allow an unobstructed view of the field. On each side there is an LCD screen, a buzzer and a USB port, which can be seen in figure 2 below.

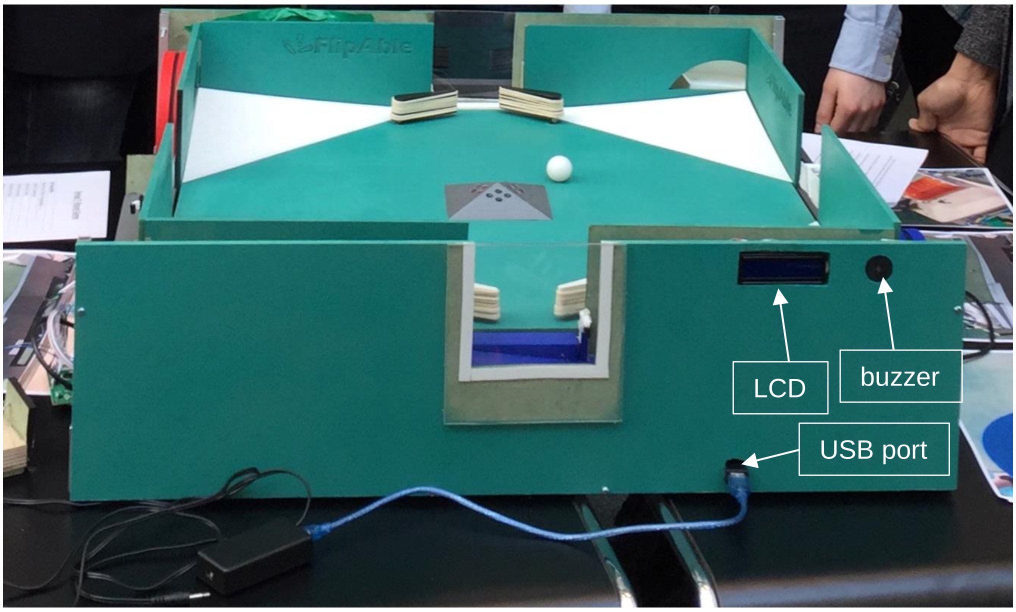


Figure 2 - Player view of first prototype

A chassis was made of bolted tubes of stainless steel and aluminium, satisfying our need for robustness and resistance to corrosion. Aluminium was used for the central cross (see Figure 3) supporting the playfield because it does not bend under its own weight and the load. Stainless steel was used for the rest of the structure because it can withstand drilling and bolting. The tubes and pillars were bolted together using mild steel bolts and corner brackets.

The outer walls and the playing field were made of Medium-Density Fibreboard (MDF). This material was chosen because it “ has no grain, it can be cut, drilled, machined and filed without damaging the surface” [22]; it is also moisture-resistant and low-cost which satisfy the technical requirements.

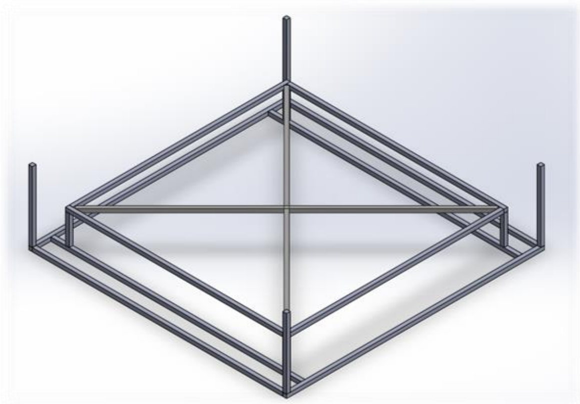


Figure 3 – Basic frame design on CAD, aluminium in central cross and stainless steel elsewhere

3.2. Button Control

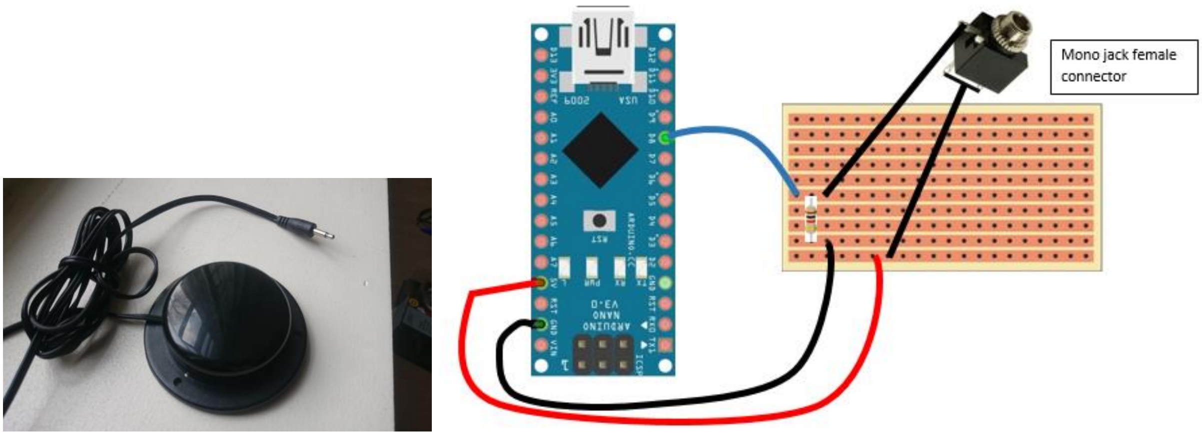


Figure 4 – Common assistive technology button, with jack plug (left). Adapter for button (right)

The button is a simple assistive technology device that can be used by the majority of students at John Chilton School. For the purposes of this game, any simple motor function in the player’s arms will suffice to operate it. The school already has many of these standard buttons in stock. These buttons with jack plugs were therefore adapted to our game by converting the press of a button to a digital pulse. The press of the button acts like a switch shorting the 5V (jack pin) with Arduino digital pin (jack ground pin) (See Fig.4). The code in the Arduino Nano converts the 5V input detected by the digital pin to a serial output. A long press only outputs once (see code in appendix F.5 and response in Figure 20.a)). A micro-USB to USB cable was then connected to the Arduino allowing serial communication with the CPU to control the flippers. The code uploaded on the Arduino can be read in Appendix F.5.

3.3 Word Detection Control

The word detection (WD) flipper control is implemented to allow children with motor disabilities to play the game. The word detection unit consists of a microphone, a sound card, a Raspberry Pi 0 (RPi0) and an Arduino Nano. The prototype is shown in figure 6. The microphone collects the analogue audio signals while the sound card discretizes, quantises and filters the signals. The RPi0 microprocessor compares the captured signal with a keyword audio model and activates the flipper bats, via the game CPU, upon detection of the keyword. This process is summarised by the flowchart in Figure 5. In order to comply with the pre-design game specification requirements, the short and easily pronounced keyword "GO" was chosen to allow accessibility of the unit by the majority of the children noting that those with better vocalisation will tend towards this control.

Keyword → Input to microphone → Sound Card → Raspberry Pi 0 → Raspberry Pi 3

Figure 5 – Flowchart describing the stages in word detection

The online hotword training service "Snowboy" provides the audio model and a keyword detector program free of charge. The service trains the keyword model with machine learning techniques based on a set of voice inputs from the perspective users. The intermediate RPi0 processing module was chosen to reduce the computational workload assigned to the Central Processing Unit (RPi3), inherent in the keyword detection application. Connections between Arduino, RPi0 and CPU are provided in Appendix E.6.

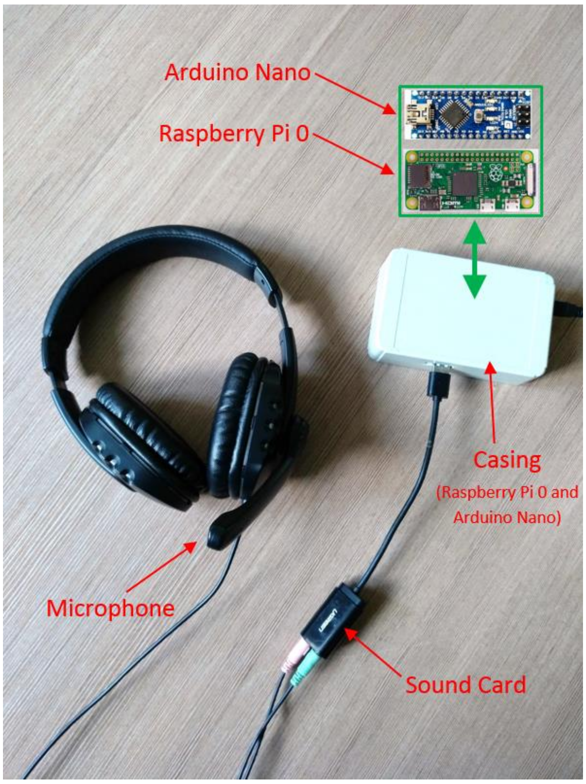


Figure 6 – Word Detection Control Unit

3.4 Eyebrow Control

There are children who might not be able to use their hands or speech as easily as others to control the game, but they might have control of the muscle in their forehead (i.e. frontalis muscle) and are able to raise their eyebrow. This action can be used to control the flippers by using a microphone to record the sound made by the contraction of the muscle. As some of the children have involuntary head movements which would be detected by the microphone, the gyroscope of an IMU (SparkFun 9DoF sensor stick) was used to detect the angular velocity of the head. This prevents false activation of the flippers. If a big movement is detected, the control is not triggered despite the output from the microphone. The combination between the microphone and the IMU creates a binary switch which controls the set of flippers. This control is mounted inside of a headband shown in Figure 7 (left), which can easily be worn by children.

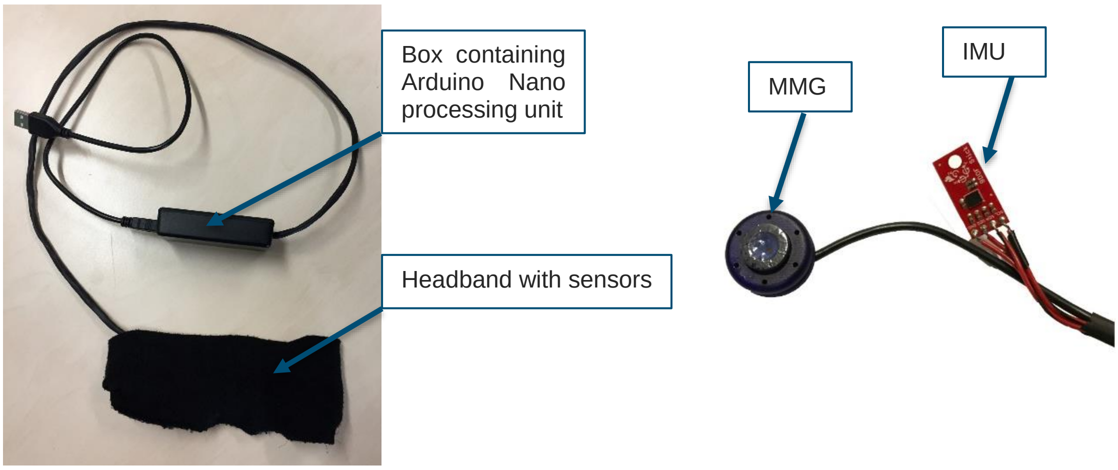


Figure 7 – The eyebrow control: final design (left) and sensors (right)

Only one MMG sensor was found on the market but it was too large for mounting above the eyebrow and the supplier shipped only in bulk. A researcher at Imperial College London, Dr Ravi Vaidyanathan, is currently developing a small MMG sensor that has a high signal to noise ratio and donated a prototype to this project. The sensor consists of a diaphragm over a sealed chamber which together detect low frequency pressure changes (see Figure 8) [23]. The material used for the diaphragm, Mylar, has no known allergic risks [23]. The pressure changes are recorded by a microphone (Knowles SPU1410LR5H-QB; nominal current consumption: 120 µA) on the opposite side of the chamber [23]. The size and diameter of the chamber allow for maximum gain and reduction of noise. [23]

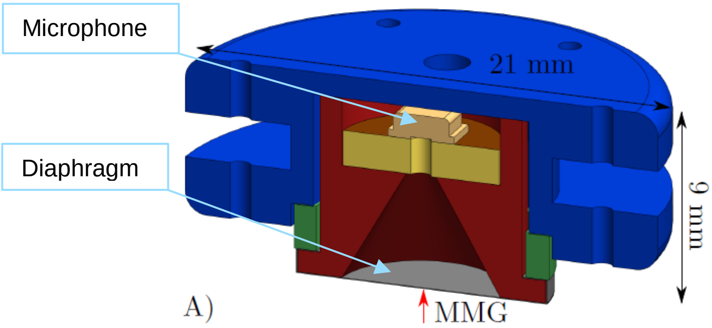


Figure 8 – Detail of the MMG sensor [23]

In order to achieve the integration of inputs, an Arduino Nano was used. Circuit showing connections between MMG and different components is shown in Appendix E.5. After testing, the threshold for a sound input to be considered as a valid eyebrow movement was found to be 200mV. A low pass filter was added in the first iteration, but unexpectedly the resulting signal had more noise than the unfiltered one. The reason behind this might have been due to the noise prone components of the circuit, such as capacitors and operational amplifiers. The filter was thus removed from the circuit. Further testing also showed that the gyroscope was more relevant than the other components of the IMU (accelerometer and magnetometer) for detecting head movements. The action is considered an eyebrow movement when the output value of the gyroscope is below 50 °/s. A simple code explained in the flow chart in figure 9 and shown in Appendix F.7 was implemented to take into account the two thresholds and only activate the control for an eyebrow movement when the head is almost static. The Arduino Nano has serial input/output which means that it is only necessary to write to the serial output to control the flippers through the USB connection.

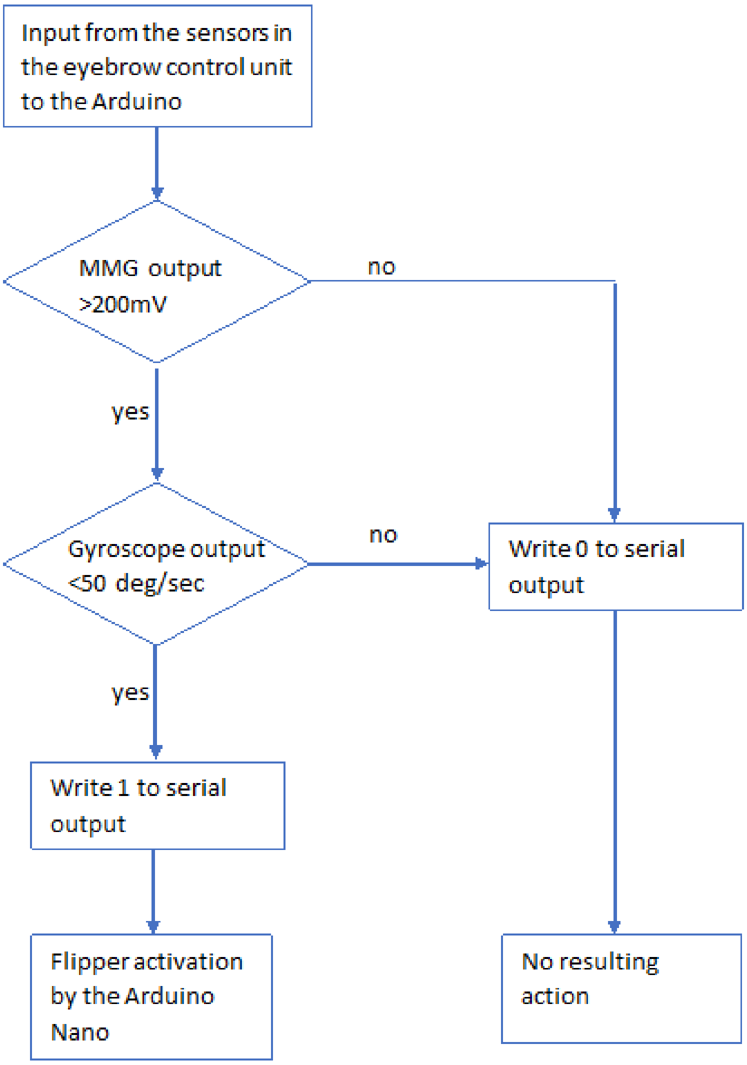


Figure 9 – Eyebrow control code schematic (see Appendix F.7 for code)

3.5 Ball Return Mechanism

In order to meet user requirement 6 (i.e. to build an autonomous game), a mechanism was developed on each side to return balls to the playfield when a goal is scored. Figure 10 shows the final design. When a ball enters the goal, it rolls down a slope until it reaches a large wheel (shown in detail in Figure 11) with 4 compartments. Both the slope and the wheel were 3D printed. The wheel is rotated by a timing belt connected to a pulley on a DC motor (951D Series) [24]. Circuitry involving motors is shown in Appendix E.4. The DC motor has a speed of 230rpm, and as the pulley has diameter 3 mm and the large wheel has diameter 13.5 mm, the large wheel rotates at a much lower speed of approximately 1rev/sec (i.e. 60rpm). This slower speed enables the ball to fall from the ramp into one of the 4 compartments. The wheel rotates clockwise such that the ball is trapped between the compartment walls and the game wall until reaching the release hole above the playing field level. The ball rolls out of the compartment as the floor of the compartment is sloped towards the field. To reduce the friction at the connection of the wheel with the static chassis, a bearing was added at the attachment.

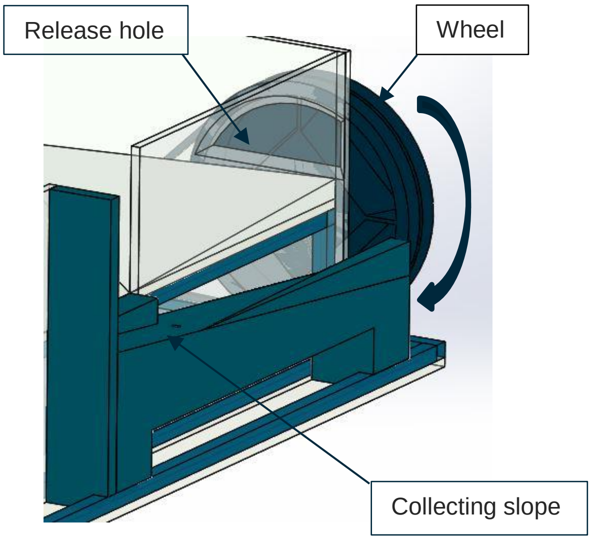


Figure 10 – The final design of the ball return mechanism, comprising a ramp and a large wheel

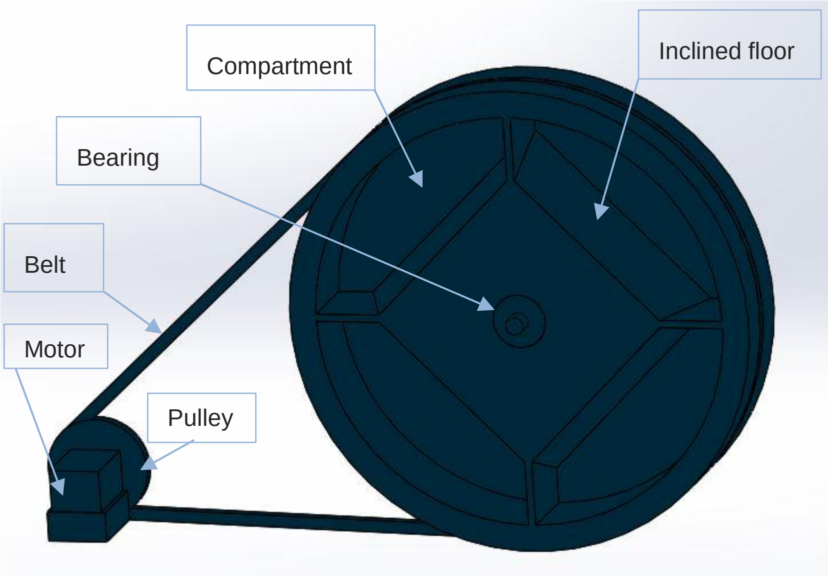


Figure 11 –The large wheel is driven by the timing belt connected to a pulley on a DC motor

3.6 Ball Counting Mechanism

The mechanism for counting the number of goals conceded by each player (i.e. the number of balls entering that goal), uses the principle of Infrared beam break sensors, which works by having a pair of an infrared (IR) emitter and a Photodiode (IR receiver), positioned opposite each other, mounted on the ball return mechanism collecting slope as shown in Figure 12.

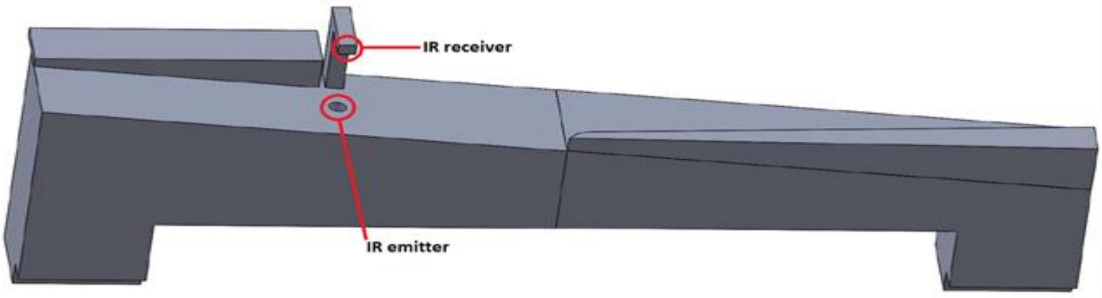


Figure 12 – Model showing the location of IR Receiver and Emitter pair on one of the slopes

The board contains four pairs of IR emitters and detectors, one on each side connected as in Figure 13, which are operated by one Arduino Nano on each side. In addition, one of the four Arduinos is operating as the Control Unit, keeping track of all other ball counting mechanisms and exchanging game status information with the Central Processing Unit. The number of goals conceded by each player is displayed on an LCD Display Module (HD44780 Blue) on each side (Appendix E.1), controlled by the respective Arduino. For this mechanism to work, the ball must be opaque such that it does not allow the IR beam to go through. The ball material, Delrin polyoxymethylene was chosen to satisfy this condition.

When a voltage of 5 Volts is applied between the leads of the semiconductor light source, by the corresponding Arduino, the emitter side sends out a continuous beam of human-invisible IR light of wavelength 940nm, which is detected by the receiver that is sensitive to that particular frequency of light. In the case when no ball rolls between the emitter and receiver, the receiver detects a continuous infrared beam and sends an output of digital zero (0 Volts) to the Arduino responsible for that side of the board. In the case of a goal being conceded, the continuous IR beam is interrupted for a brief time period by the moving ball. As a result, the output of the photodiode is a digital 1 (5 Volts) which is sent to the corresponding Arduino Processing Unit that is responsible for increasing the counter displayed on the LCD screen. This produces a short sound on a buzzer and a short flash from the central pyramid LEDs to indicate to the player that a goal was conceded, which achieves objective 8 (i.e. to provide audio and visual feedback). The LCD and buzzer are mounted in the outer wall of the board game.
When the threshold of 20 goals is reached on one of the sides, the counting is stopped, and the information is sent to the Central Processing Arduino and the RaspberryPi3 to enter the game-over state.

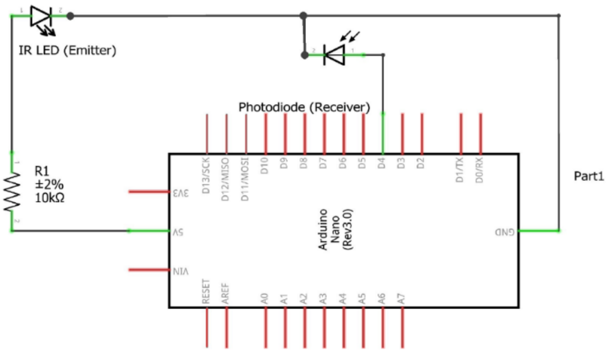


Figure 13 – Schematic Diagram of Emitter/Receiver pair

3.7 Flipper assembly

3.7.1 <u>Mechanical Components of flipper</u>

Figure 14 below summarizes the mechanical assembly of the flipper:

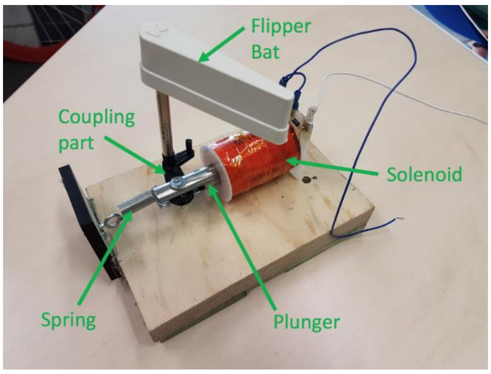


Figure 14 – Mechanical Flipper Assembly

The flipper assembly was computationally designed using SolidWorks. It consists of a double winding copper coil, a plunger (cylindrical solid shaft), a plastic (TPU) coupling part, a brass rod, a spring and a flipper bat. It was aimed to use magnetic materials of low remanence (low permanent magnetization ability; i.e. iron-nickel alloy) for the plunger design, but mild-steel was ultimately used due to its availability and due to time constraints. Low remanence prevents the permanent magnetization of the plunger which would hinder its ability to be linearly translated after extended periods of usage. The coupling part is 3D printed and allows the conversion of the linear plunger translation to the required angular flipper bat rotation. TPU is used as it is a flexible material which facilitates the press fitting of the flipper bat on the part (see Figure 14 above). The extension spring restores the initial position of the plunger in the coil, following a pulling action due to the current-induced magnetic field.

3.7.2 Electrical Components of flipper

The flipper design is based on a pair of pinball-standard high voltage coils (Rating: 48V), operating as solenoids. Upon control input detection, the CPU activates an nMOSFET controlling the corresponding flipper coils, allowing current to flow and hence magnetic field generation through the copper coils. As a result, the plunger is pulled in the coil, which induces the flipper bat rotation due to the plastic coupling part-plunger connection. The coils are powered by the power supply. An nMOSFET board controlled by the CPU supplies the coils with power upon detection of an input signal from the corresponding control module, as seen in Figure 15.

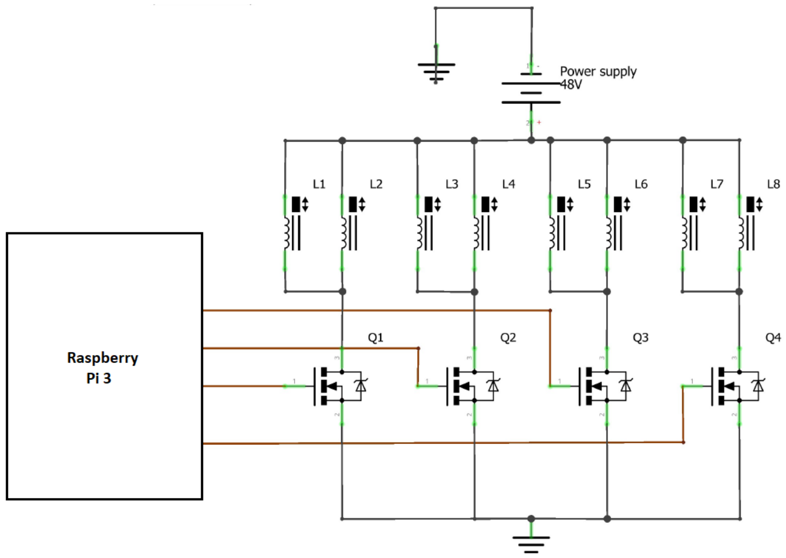


Figure 15 – Schematic of flipper solenoid control by CPU via nMOS transistors. L1-L8 represent coils and Q1-Q4 represent nMOSFETs

The flipper solenoids used in the board game are three-terminal, dual winding inductors of series FL-11630 with diodes connected between the terminals. The primary coil, which has a nominal impedance of 5.1 Ω, acts as the power coil and consumes a maximum of 450 W. The secondary coil, which has an impedance of 161 Ω, acts as a hold coil and is activated only when the solenoid is fully activated with full extension of the flipper bat and consumes a maximum power of 14 W.

As flipper manufacturers suggest, for proper and safe operation of solenoids, a special type of switch called EOS (End of stroke) switch made of two metal conductors, should be used (see E.3 section in Appendix for explanation). During the design and manufacturing stage, the circuit suggested in figure 34 (Appendix E.3) was modified, due to the unavailability of the EOS switch.

To resolve this, terminals 1 and 2 of solenoid are connected by a short-circuit, thus bypassing completely the primary coil (Figure 16 below). This decreases the maximum power available for each solenoid to 7W, i.e. 30 times less than the predicted power if the circuit suggested by manufacturers was to be implemented. Despite this limitation to the performance of the solenoids, this modification was implemented as it ensures that the design is protected in case the CPU controlling the MOSFET switches fails. In case of faulty continuous activation of the flipper, which can only occur if the CPU sends a constant digital 1 signal to the MOSFETs, the power supplied to each solenoid is very low and does not cause any damage to the device.

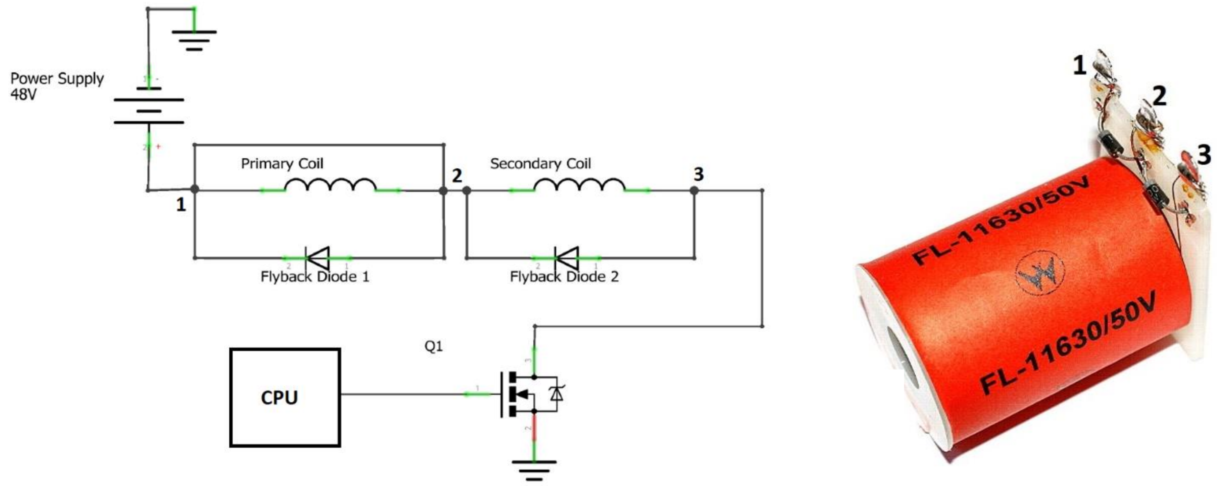


Figure 16 – Detailed schematic diagram of the modified flipper solenoid circuitry to compensate for the absence of EOS switch. This was the circuit implemented in the final product. Numbers 1-3 represent the three terminals on solenoid shown on picture on the right.

The schematic diagram of solenoids used also contains flyback diodes of model 1N4004 [25], which prevent the occurrence of a voltage spike on the switching transistors as explained in appendix E.2. Such voltage spikes can create sparks or electric arcs which shorten the lifetime of the MOSFET switches.

3.8 Power Distribution Unit

Due to the high power and voltage requirements of the solenoids and other parts of the game, batteries could not be used for powering the whole game as was initially considered. Therefore, an AC-DC power supply unit, which is connected to mains (240V AC), and has maximum output ratings of 48V DC, 6.7 A (321 W) is used. This power supply unit RSP-320 series [26], manufactured by company Mean Well, with these specifications is used since the solenoids require 50V DC for normal operation – similar to all pinball games. All 8 solenoids are connected directly to the output of the power supply using a power distribution strip board connected to the CPU through MOSFET transistors, as shown in figure 17. The solenoids are only activated once a game control input is identified.

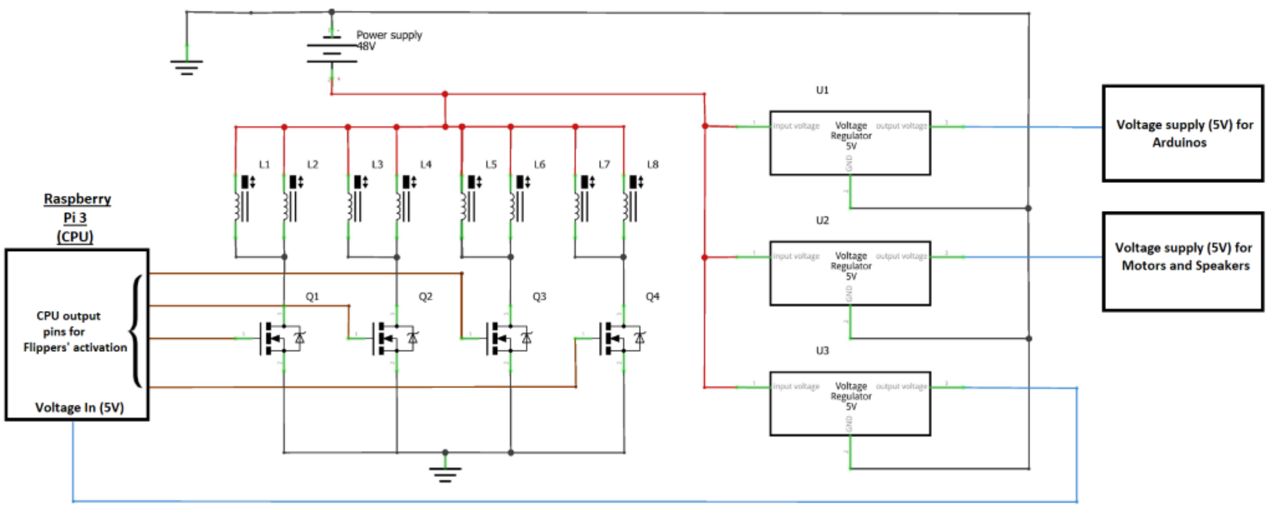


Figure 17 – Schematic Diagram of Power distribution for main board components. Components U1-U3 represent the voltage regulators used (5003MC)

All other electrical components (i.e. Raspberry Pi, Speakers and Arduino) require a maximum supply voltage of 5V DC. Hence, step-down voltage Buck converters were used to convert 50V DC to 5V DC. These ensure that the voltage supplied to each Processing unit has a continuous constant supply of 5V DC.

The power supply purchased has additional specifications that satisfy some user requirements. The supply is comparatively light (0.90kg) allowing it to be easily supported on the base of the board and its dimensions are ideal for positioning the component inside the board to be inaccessible to the user.

3.9 Central Processing Unit

The integration and coordination of all electrical components is done with a Raspberry Pi3 which is the CPU of the game. There are two main parts to its function, as illustrated in figure 18. The first is the activation of flippers on each side, according to the information received from the control plugged in on that side. The second is having the game on or off conforming to the goal count. The complete code can be found in Appendices F.1 and F.2.

3.9.1 Control – Flipper coordination

The controls are connected through USB to one of the four ports of the RaspberryPi3. They use this to send a digital 1 when requesting a flipper activation or a digital 0 otherwise. In order for all controls to be checked simultaneously, parallel processing is used, meaning that the 4 sides are looked after by 4 separate processes which run at the same time. This avoids the case when a player would activate its control but the flipper would not move, since the CPU is checking another port at that time. Each sub process is also responsible for the audio feedback so when a control is activated, it plays a short sound through the speakers. The language chosen for the code in order to allow parallel processing was Python.

3.9.2. Game state control

When the game starts, the CPU checks the USB ports in order to determine which players have plugged in a control and are therefore playing. Sub processes for those sides are started as soon as the control is detected. Ten seconds are given for this step, followed by the actual start of the game marked by a song played through the speakers. The ball return and ball counting mechanisms are also turned on. The motor of the ball return mechanism receives power through a MOSFET switch. The switch is turned on and off with the same signal that indicates the start and end of game. When on, the wheel rotates continuously.

The three slave Arduinos signal their state to the master Arduino with a digital variable (0 while under goal limit, 1 when goal limit reached). The master Arduino regularly samples that digital state and checks its own counter. This is shown in figure 19. In turn, it signals the state of the game to the Pi (CPU) with a second digital variable (1 while no side has lost, 0 when any side has lost).

When the game is over, the Central Arduino signals the rest of the Arduinos of the ball counting mechanism to reset their counters and wait for the game to start again and also does that for the side it controls itself. The CPU responds to this signal as well by turning off the ball return mechanism, playing the game over song through the speakers and killing the processes on each side. For the game to restart, all the players need to activate their controls at least once. When that is done, the 10 seconds loop, which checks the USB ports starts again.

Blue: Controllers-pi communication

Orange: Arduino-pi communication (to Arduino, from Arduino)

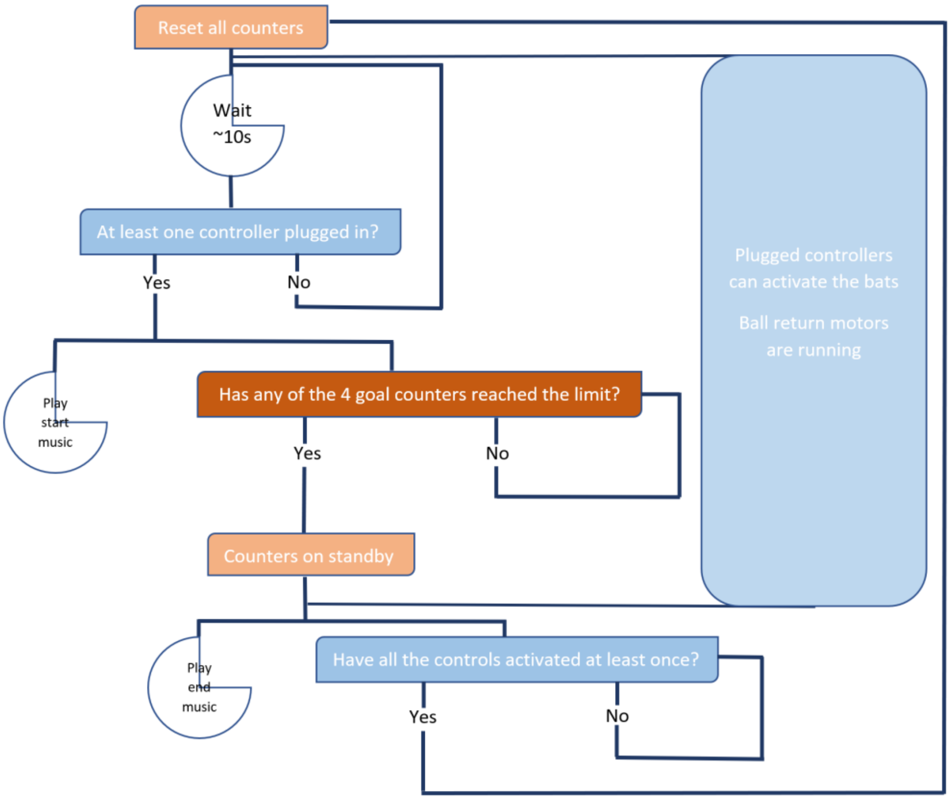


Figure 18 – CPU (Central Raspberry Pi) Flow chart. Temporal flow from the top down (Appendix F.1&F.2 for code)

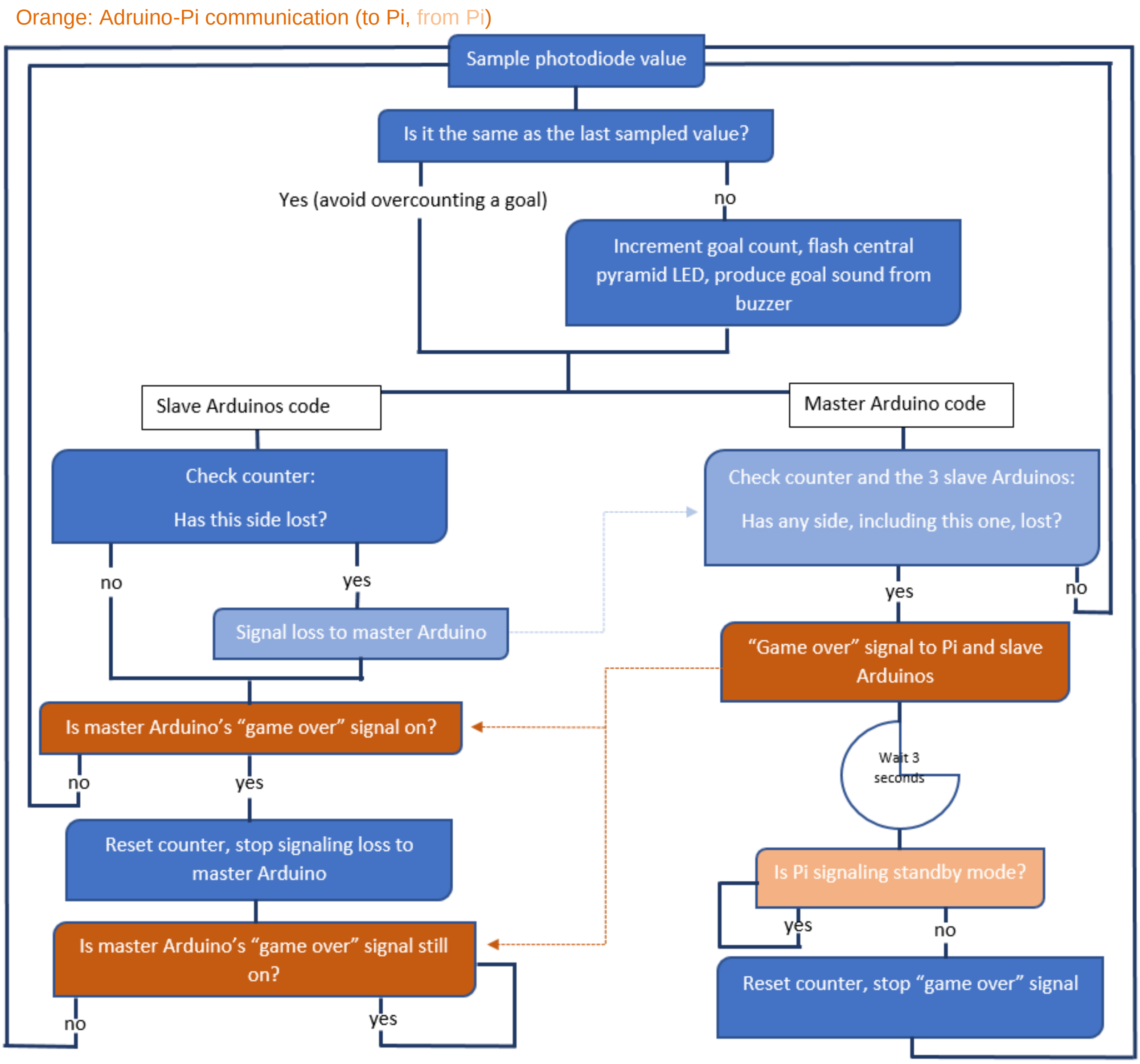


Figure 19 – Master (central ball counting) and Slave (simple ball counting) Arduinos Flow chart. Temporal flow from the top down. Note that the top portion is common to both master and slave codes (Appendix F.3&F.4 for code)

3.10 Sound Feedback

A pair of stereo speakers is used as the main output of sound for the game. These are fixed inside the board at two opposite corners to ensure all sides receive the same intensity of sound. The speakers are connected via a jack to the CPU (Raspberry Pi3) which is in charge of playing each audio file at the right time.

There are 3 different sounds emitted by the speakers: start of the game song, flipper action effect and ending game song (note: these were all downloaded from an open free sound source database [27]). The first and last have a medium length and are used at the start and end of the game respectively, whereas the action of the flipper sound is short and powerful, subjected to the input of the players, triggered whenever any of them activates their controls.

An extra sound is also emitted by a buzzer whenever a goal is scored to indicate to each player the addition of a score to their goal. There are 4 buzzers in total, one on each side of the game. When a goal is scored only the corresponding buzzer activates.

The parallel processing ensures the sounds play at the exact moment the action happens, ensuring the audio information is interpreted correctly.

3.11 Safety and Security

3.11.1 Electrical Safety

Power distribution unit

To avoid any contact with the users or any other components, the power supply unit is enclosed in a plastic case with two metal sides. The earth wire of the mains plug was electrically connected to the two metal sides to ensure it is automatically set to ground in the case of a fault or any voltage driven into the case. The whole power distribution unit also has 2 fuses: one 13A fuse inside the mains plug and one 7A fuse inside the power supply.

The electrically-insulated case contains slits on each of its sides to allow for air exchange with surroundings to prevent overheating. The power supply itself contains a built-in DC fan for forced air cooling.

The device is also protected by Hiccup mode [28] where it switches off the power in cases of overload (too much current passing through electric wires) or short circuit and carries out periodic attempted restarts until the fault is eliminated.

The unit also offers over voltage protection which is a feature that shuts down the supply when the voltage exceeds a pre-set level and requires re-power of the supply to recover, thus preventing damage to the electronic components. Sources of voltage surges in the power network (mains) are primarily lightnings and over-voltages generated by operations or incidents on the network.

Power supplies of RSP-320 series also have RoHS Certificate of Compliance [29] [30] by satisfying the EU Directive 2011/65/EU concerning restriction of the use of certain hazardous substances (Lead, Mercury, Polybrominated Biphenyls, Polybrominated Diphenyl Ethers, Hexavalent Chromium and Cadmium) in electronic and electrical equipment.

The device meets Low Voltage Directive (2014/35/EU) and Electromagnetic Compatibility Directive (2014/30/EU) for electromagnetic interference and susceptibility.

Thermal Insulation Tape

The solenoids and electrical boards are wrapped in several layers of thermal insulation tape to prevent from overheating and avoid contact between the electrical components.

External Plug

An external standard plug connected to mains is used to power the game. This is necessary to allow the users to switch the game on and off without being exposed to the electrical components inside the board. This prevents any electrical risks in case of a faulty component in the circuit.

Fuses

A 2A fuse was used to prevent overpowering the CPU (RPi3) in case of a short-circuit, to ensure it would function properly. If a higher current was to be input, the fuse would melt but the Rpi3 would not have to be replaced.

Control Systems Casing

The electronics for all the controls are enclosed inside a plastic box closed by screws (insulated from the electrical components)**.** This ensures that there are no electrical components exposed to the users and to other components when using the controls. It is worth noting that the digital signals' power level is too low to harm users.

3.11.2 Mechanical Safety

Filing

The sharp edges of the game are filed to prevent any risk of cuts and scratches for user.

Acrylic Glass

In the future, an Acrylic Glass cover should be bolted on top of the playing field to keep the balls within the game and prevent harm to the users from deflected projectiles.

4. Testing and Evaluation

Testing against hazards and assessing the risks of the electricity dependent, power consuming board game was characterized as top priority before the children gained access to the board game. The sections below provide an insight into the physical testing against both user-threatening risks and game failure risks. An evaluation matrix summarises the tests and results gained on different gameplay and risk aspects.

4.1 Physical Testing

4.1.1 Electrical Testing

Power Supply

The power supply regulates mains to 48V DC, providing maximum current and power of 6.7A and 321W respectively. The robustness and integrity of the supply was tested by measuring the voltage output of each of the 3 output supply terminals with a multimeter. All output terminals provided a constant, stable 48V supply with minor variations.

Earthing/Insulation.

The earthing of the metal sides was assessed with continuity tests of multimeters carried out between each of the metal sides and the common ground. The mains plug was equipped with a 13A fuse for safety purposes. Further safety precautions include a 2A fuse between the power supply and the CPU Raspberry Pi 3 and an 8A fuse integrated within the power supply. The integrity of the insulation and fuses were tested with continuity tests.

4.1.2 Mechanical Testing

Performing tensile and compression tests on the playing field were considered but not executed since material properties are widely available (MDF compressive yield stress: 10MPa [31]).

4.1.3 Performance Testing

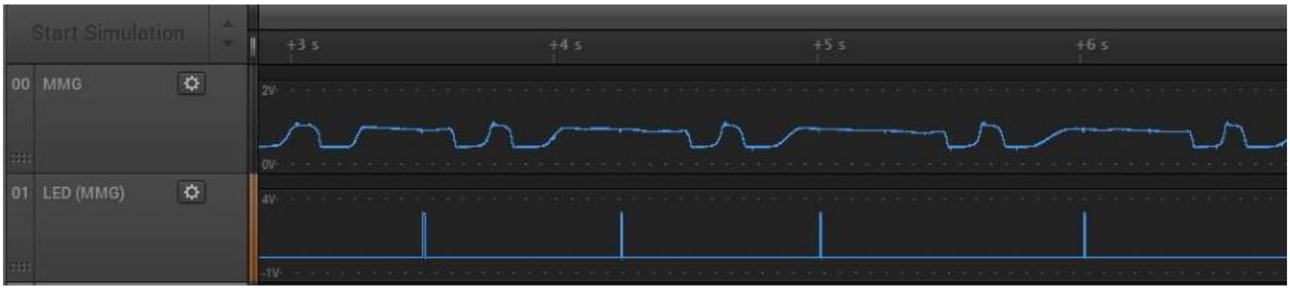
Testing of Control Module Performance

The performance of the control modules in the flipper bat activation was assessed in terms of success rate and activation delay. The circuit of each control was tested with an LED instead of the flipper mechanism.

The method for determination of success rate was as follows: a group of participants (N=32) was asked to try each control 20 times while the person performing the test counted the number of times the LED flashed in response to the control activation. The keyword detection efficiency of the word detection unit was tested with people who had previously trained the model as well as with individuals who had not, but the difference was not significant, with 71% success in the first case and 73% in the second case. The results of success rate for each control are shown in Figure 21a.

The method for determination of delay was as follows: A Saleae Logic 8 Logic Analyzer was used to detect the input and output signals of each control. The waves were used to calculate the delay between control activation and flipper activation. For the Word Detection unit, there was no point in the circuit where the probe could have been placed in order to record the input signal from the microphone, requiring another method to be used in this case. The test was done with the subject simultaneously pressing the button and saying “Go”. The probes were placed at the immediate output of the button (before reaching its Arduino Nano unit) and at the LED was controlled by the Word Detection Control. In the case of the Button, the setup was the same, but the LED was controlled by the button instead. For the eyebrow control the first probe was placed at the Arduino Nano pin receiving the input from the microphone and the LED was controlled by the eyebrow control. The latency period was computationally measured with the Saleae Logic Software platform as the time difference between the response (of the LED) impulse and the rising edge of the input waveform. The difference was recorded in a table later used to give the results in Figure 21b. An example of the resulting waves is shown in Figure 20 for each control.

a) Response delay of the Pushbutton



b) Response delay of the MMG component of the Eyebrow control

c) Response delay of the Word Detection Control (VR)

Figure 20 – Response delay analysis using Saleae Logic Software

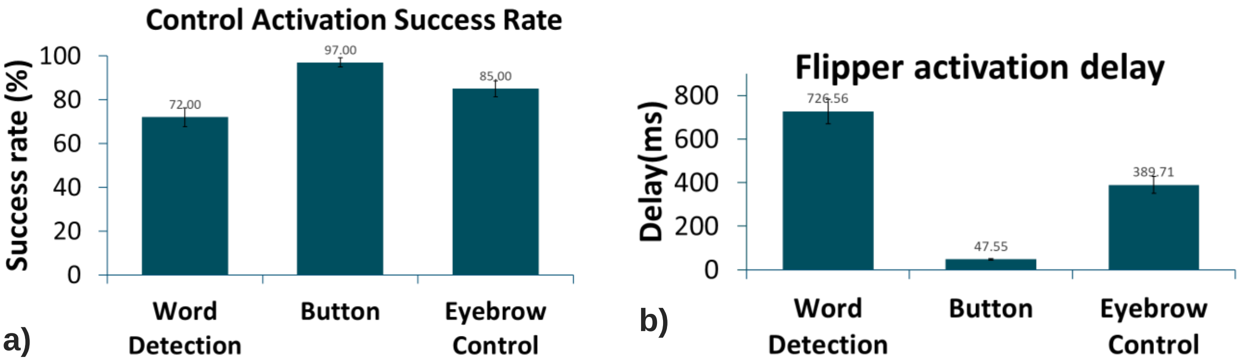


Figure 21 – a) Control activation success rate b) Activation delay comparison

The Button control had an almost perfect response with a high success rate and a small delay, while the word detection control was characterized as the control with the least success rate and highest flipper activation delays. The eyebrow control showed reasonable results, with good success rate but quite a long delay of around 400ms.

Testing of flipper bat extension and ball deflection performance

The flipper bat extension efficiency was measured in terms of the deflected balls velocity and the time required for full bat extension. The ball velocity was approximated by computationally (video analysis) measuring the time needed by the deflected ball to travel 15cm. The full bat extension time was determined by video analysis of the test conducted. Using the software platform Media Player Classic (MPC-HC) the time was determined as 0.38s. The ball velocity is 10cm / 0.38 s or 0.26 metres per second. The full bat deflection time was determined as 0.092s.

Ball counting performance.

The ball counting effectiveness was evaluated in terms of the maximum frequency rate at which objects are recognized by the emitter-receiver pair. The maximum frequency rate was determined by continuously probing the ball counting mechanism by moving a team member’s index finger below the receiver to stimulate the sensor. The stimulation frequency was continuously decreased until the IR sensor detected the finger. The minimum time between two successful finger detections (as seen from the activated LEDs in the playfield centre) was evaluated by computational video analysis of the test conducted. The minimum time difference between two IR sensor activations was determined as 1.446 seconds and the maximum corresponding frequency is 0.69Hz.

4.1.4 Miscellaneous Testing

Weight Testing

The prototype was weighed on a digital scale - displayed total weight: 18.7 kg.

Temperature and Overheating
In order for the risk of overheating of power supply to be assessed, measurements of temperature were taken over different operating times using a thermocouple probe connected to a multimeter displaying the device's temperature. The graph of figure 22 summarizes the measurements taken:

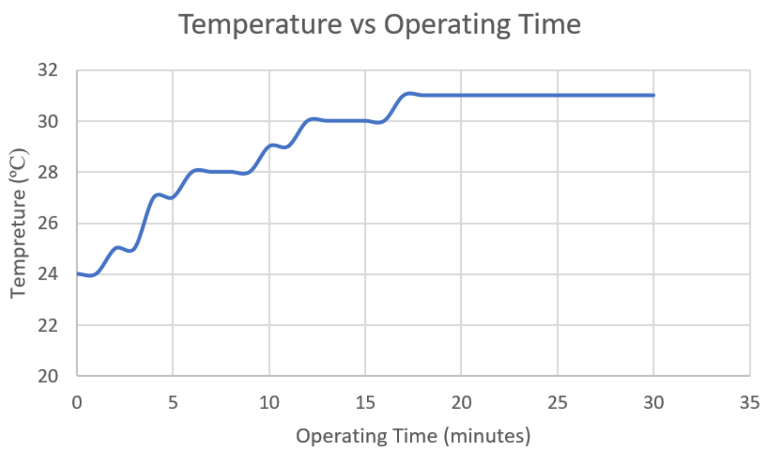


Figure 22 – Temperature of power supply over operating time

According to the specification sheet of the power supply, the working temperature range is between -30°C and +70°C. The graph above illustrates that the temperature of the device reaches a plateau of 31°C.
The temperature of different board components was measured after a 2-minute time interval of gameplay with a multimeter thermocouple. The solenoid temperature was 53ºC and the Raspberry Pi 3 CPU temperature was 54ºC.

4.2 User Feedback

The final prototype was presented on Demonstration Day and Imperial Festival - feedback was collected from both sessions where around 300 individuals of different ages had the chance to try the game. Most of the user requirements were met although some extra assistance and calibration was frequently needed at Imperial Festival from members of the team.

The game evoked a lot of interest amongst the users of all ages, and especially the target users of the young age. The novelty of the game was not a problem for any of the users and everyone could easily understand the concept of the game with little help, making the game intuitive to play and user friendly. Users reacted positively to the aesthetics and size of the game, highlighting the colourfulness and dimensions as suitable for the intended users and had no problems on the visibility of their goal or any part of the playing field. The use of audio-visual feedback from the game was well-received, especially the goal counting mechanism which provided a positive competitive atmosphere amongst the players.

The users were impressed by the variety of controls available for them to use. Some problems were encountered by a few individuals when trying to use the word detection (WD) and eyebrow control units (no problems were encountered with the button). Improvements were suggested including the reduction of the delay and increase on the success rate of WD (which was sensed to give several false positives) and a better calibration method for the eyebrow control unit, which was very difficult to calibrate on several individuals. The headband with the eyebrow control was also hard to fix sometimes which meant that many children were only able to activate it by tapping. These need to be addressed and tested again with different users to provide fairness between all the controls and ensure they all work successfully.

The users were frustrated that the balls were becoming stationary at the centre of the field, which was due to the low power provided by the flippers and the friction between the playing field and the ball.

The feedback on Demonstration Day was considerably better than that at Imperial Festival, which was due to the loosening of some fixings from the controls and the ball return mechanisms, making them perform poorly at the festival. This was caused mainly by the rigorous use by young children. Hence the design of the game needs to be improved to be more robust to such disturbances if it is to be used within a child-friendly setting. This should be dealt with by using more durable fixings and long-term testing.

4.3 Evaluation Matrix

Table 4 - Evaluation Matrix

Requirement	Evaluation	Met (Yes (Y) / No (N))
Variety of Control Technologies	Three different control systems can be used to play.	Y
Reliable Game Response	• Push Button – 97% activation success rate • Word Detection – 72% • Eyebrow Control – 85%	Y
Audio-visual feedback	Sounds and LED lights work successfully and coincide with game actions.	Y
Dynamic Game Play	The ball movement is not continuous across the play field thus game flow is stagnated. The ball return mechanism jams 30% of the time.	N
Flipper Power	The power provided to the ball is not sufficient to propel it continuously across the play field.	N
Visibility of Playing Field	The game is painted a bright cyan and balls of various colours are used to play. The pyramid and slopes are a contrasting stark white	Y
Light weight	The game weighs approximately 18.7kg without plexiglass. Slightly above the requirement.	N
Suitable size for a player on a wheelchair	Dimensions are met (70cm x 70cm)	Y

Universal Ports for Control	All three control units have USB output thus can connect universally to a USB port on any side of the game.	Y
Control Response Time Homogeneity	The three controls have different time responses, thus are not perfectly homogeneous. • Word detection –730ms • Button –50ms • Eyebrow control –400ms	N
Autonomous Game	Once plugged in, the game flow is fully controlled by the Raspberry Pi. No input is required from the users, assuming the ball return mechanism is functioning.	Y
Safety and Security	Live electrical components are insulated within the game and the metal chassis is earthed. Fuses are installed to protect sensitive components.	Y
Cost	The total cost of the prototype exceeds the initial budget of £600 by £151.52	N

5. Discussion

5.1 Limitations

The main limitation of this prototype is that the force with which the flippers hit the ball is not adequate for exciting and dynamic play. This results in dead spots on the play field where the ball rolls to a stop and cannot be hit by any of the players. Dynamic game play is also hindered by the ball return mechanism as there is too much friction between the ball and the inside wall, causing the ball to get stuck in the wheel at times.

In terms of performance, the control systems are limited in different ways. The word detection control and the eyebrow control mechanism both have a less than 80% success rate, so they can give false activation of the flippers or conversely can be nonresponsive. The three control systems have different delays in response time which puts some players at a disadvantage.

Autonomy is obtained, but over time the ball return timing belts loosen, causing the balls to be trapped as the wheel is inhibited from proper rotation. There is also too much friction between the ball and the inner wall of the game, also causing a considerable resistance to the wheel rotation.

5.2 Future improvements

5.2.1 Control systems

From the measured time delay between the action of the player and the movement of the flipper bat, it can be seen that the word detection control system has the most delayed response. Thus, it is necessary in the future iterations, to precisely measure each time response and add a delay to the eyebrow control and the push-button in the central processing stage to compensate for the larger latency period of the word detection control. Increasing the accuracy of the hotword detection by the Snowboy software is a key point in making the game fair for every player. This can only be done if the hotword model is trained online by more children of John Chilton School. The eyebrow control system could also be improved after further testing on the children at the school. Recording the output values of the gyroscope and the microphone would allow adjustment of the thresholds used in the Arduino code (see section 3.4. Eyebrow Control and appendix F.7), which would make the response more personalized to specific players. During testing, some students experienced pain due to the headband. The system should thus be made more comfortable to wear.

5.2.2 Dynamics of the gameplay

Flipper bat assembly

Use of an EOS switch in flipper assembly circuitry, ensures higher power supplied to each solenoid generating higher pulling forces, making the game more dynamic. Simultaneously, this will also ensure higher safety for user and electrical and electronic components, due to lower power supplied in the case of continuous activation of solenoid as explained in section E.3 in Appendix. An alternative option is purchase of a ready-made complete flipper assembly from a pinball machine company. Such an option was not plausible due to the high price of assembly, which would be completely beyond the available budget, as eight of such assemblies were required for the board game.

Increasing the energy transmitted to the ball by the flipper bat is the next step to make the game more dynamic as this would increase the ball's velocity. This can be realized by changing the material of the solenoid plunger. Opting for iron-nickel alloy would decrease the magnetic remanence of the plunger, which means its permanent magnetization ability decreases. A lower permanent magnetization ability results in a greater force on the plunger which would allow the flipper bats to hit the ball with more energy. This modification would also increase the life span of the flipper solenoids as their permanent magnetization would increase more slowly over time than it did in the first iteration.

Power supply
Increasing the power supplied to each solenoid would also increase the ball's velocity as it would increase the bat extension speed.

Ball return mechanism
To decrease the probability of the ball getting stuck in the wheel, it is necessary to fix the motors on their supports more firmly, using screws for example. This would keep the belt under higher tension, thus rotating the wheel more forcefully. This will overcome the friction of the ball with the internal wall. The friction can also be addressed by choosing a material with a lower coefficient of friction for the inner wall or coating the part facing the wheel with a material that reduces friction.

Playing field
Additional components such as bumpers or fans could be added on or around the playing field to make the game more dynamic and keep the ball in constant movement. A thin sheet of material, suitable to the material of the ball, could be used to cover the playing field and reduce the friction coefficient of their interaction, thus further increasing the ball's velocity.

5.2.3 Microprocessors

The number of Arduino boards could be reduced by integrating the codes of the Arduinos responsible for the LCD display on each side into the single "mother" board. This would reduce the power consumption and increase the available space for electrical components inside the game, as well as reducing expenses.

5.2.4 Improved Ventilation and Temperature Control

Due to high temperature of certain components in the circuit, temperature control is vital to prevent overheating. This issue can be tackled by the introduction of slits on the side walls of the board to allow recycling of air inside the board. Fans could be installed under the playfield to reduce the risk of overheating electrical components while ensuring low power consumption of cooling units.

5.2.5 Chassis

The outer walls could be connected through hinges in order to open easily and allow for quick maintenance if any part fails.

The weight of the game could be reduced by manufacturing the supporting chassis and walls out of plastic rather than metal and MDF.

5.3 Cost Considerations

The total cost of the prototype exceeds the limit of £600 set initially (see appendix G). This is mainly due to the dimensions of the board which required large quantities of expensive raw materials. Some products were faulty so extra money was spent to buy the same item from a different source (for example: the power supply). Several components were bought at the initial stage of the project for testing but are not used in the final prototype. Due to these findings, the cost of future iterations will be reduced.

6. Conclusion

In order to create an inclusive and entertaining game for children with disabilities, a novel board game has been prototyped.

Overall the proof of concept has been achieved with a successfully autonomous game, which was user-centered and popular with children, as determined during testing. The different controls were implemented and allowed to cover a range of disabilities as desired.

Despite the game being functional on most levels, the main shortcoming of the design was in the low power of the flipper bat strikes. This could have been eliminated by implementing an EOS switch in the flipper assembly, as is conventionally done in pinball machines (see appendix E.3). The game flow was hindered by the delay response of the controls. The push button control proved the most reliable with the highest success rate and lowest delay, while the eyebrow movement detection and word detection introduced more delay and false detection. The ball return mechanism also limited the game play due to balls not returning to the game, which could have been prevented by fixing the motors more firmly so that the belts are always tensed. These limitations should be accounted for in the future iterations of the design or similar projects. The other parts of the game such as the goal counting display and the rest of the audio-visual feedback functioned as desired, allowing players with audio or visual disabilities to follow the game easily.

This product is one of the first of its kind in the field of toys using assistive technology, a field not very well catered to [32]. Further iterations of the design might suffice to make the game deliverable to the school and cut the excess costs to create a marketable design that can be accessed and benefitted from by a wider community.

References

[1] The Richest, “The top 10 most sold board games ever,” [Online]. Available: https://www.therichest.com/rich-list/most-popular/the-top-10-most-sold-board-games-ever/?view=all. [Accessed 17 06 2018].

[2] Center for Persons with Disabilities and Utah State University, “Assistive Technologies,” [Online]. Available: https://webaim.org/articles/motor/assistive. [Accessed 17 06 2018].

[3] S. Sivasankaran, “Why isn't Voice Recognition Software More Accurate,” [Online]. Available: https://www.forbes.com/sites/quora/2016/12/20/why-isnt-voice-recognition-software-more-accurate/#7b2c853c1c6a. [Accessed 17 06 2018].

[4] C. H. Morimoto, “Eye Gaze Tracking Techniques for Interactive Applications,” [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314204001109.

[5] Md. Anamul Islam, Kenneth Sundaraj, R. Badlishah Ahmad and Nizam Uddin Ahmed, “Mechanomyogram for Muscle Function Assessment: A Review,” *PMC US National Library of Medicine, National Institutes of Health,* 2013.

[6] N. Alves and T. Chau, “The design and testing of a novel mechanomyogram-driven switch controlled by small eyebrow movements,” *Journal of NeuroEngineering and Rehabilitation,* 2010.

[7] M. T. Tarata, “Mechanomyography versus Electromyography, in monitoring the muscular fatigue,” 2003. [Online]. Available: https://doi.org/10.1186/1475-925X-2-3.

[8] Edward Sazonov and Michael R. Neuman, “Wearable Sensors, Chapter 7.3: Wearable and Non-Invasive Assistive Technologies,” 2014. [Online]. Available: goo.gl/M9y27p.

[9] “Bangoli Surface EMG Sensor,” [Online]. Available: https://www.delsys.com/products/desktop-emg/surface-emg-sensors/. [Accessed 17 06 2018].

[10] E. Zion-Golumbic and Hebrew University of Jerusalem, “What is EEG?,” [Online]. Available: https://www.mada.org.il/brain/articles/faces-e.pdf. [Accessed 17 06 2018].

[11] Group5, Imperial College London , BE2-HEDP , “EEG Scalextric,” London, 2017.

[12] “Snowboy,” [Online]. Available: https://snowboy.kitt.ai/. [Accessed 17 06 2018].

[13] P. S. Giovagnoli, “Ball return conveyor system for baseball pitching machine cages; https://patents.google.com/patent/US5056782”. USA Patent US5056782A, 29 09 1988.

[14] Easy Experiments, “How to Make a Model Conveyor Belt,” [Online]. Available: https://www.youtube.com/watch?v=EOPCZhKGPLA&t=123s. [Accessed 10 6 2018].

[15] Mechanical Kits, “Build a Marble Climbing Machine from Mechanical Kits,” [Online]. Available: https://www.youtube.com/watch?v=sv-snt6LMGo. [Accessed 10 6 2018].

[16] Robotix, “How to Use IR LED and Photodiode with Arduino,” 31 May 2013. [Online]. Available: http://startrobotics.blogspot.co.uk/2013/05/how-to-use-ir-led-and-photodiode-with-arduino.html. [Accessed 17 06 2018].

[17] Instructables, “Simple IR Proximity Sensor with Arduino,” [Online]. Available: http://www.instructables.com/id/Sim ple-IR-proximity-sensor-with-Arduino/. [Accessed 17 06 2018].

[18] “Ikea,” [Online]. Available: http://www.ikea.com/us/en/catalog/categories/departments/workspaces/11844/. [Accessed 17 06 2018].

[19] “United States Consumer Product Safety Comission,” [Online]. Available: https://www.cpsc.gov/Business--Manufacturing/Business-Education/Toy-Safety. [Accessed 17 06 2018].

[20] “Intertek,” [Online]. Available: http://www.intertek.com/sparkles/safety-electric-toys/. [Accessed 17 06 2018].

[21] “US Archive,” [Online]. Available: https://ia800801.us.archive.org/10/items/en.71.1.2014/en.71.1.2014.html. [Accessed 17 06 2018].

[22] Design-Technology, “Medium Density Fibreboard,” [Online]. Available: https://design-technology.org/mdf.htm. [Accessed 17 06 2018].

[23] R. B. Woodward, S. J. Shefelbine and R. Vaidyanathan, “Pervasive Monitoring of Motion and Muscle Activation: Inertial and Mechanomyography Fusion,” 2017.

[24] RS Components, “951D Series 12mm diameter spur sub miniature Geared Motor,” MFA Como Drills, [Online]. Available: https://docs-emea.rs-online.com/webdocs/1380/0900766b813806c7.pdf. [Accessed 17 06 2018].

[25] ON Semiconductor, “Data Sheet of Diode IN4001-IN4007 series,” [Online]. Available: http://www.onsemi.com/pub/Collateral/1N4001-D.PDF. [Accessed 17 06 2018].

[26] RS Components, “RSP-320 series,” Mean Well, 14 7 2017. [Online]. Available: https://docs-emea.rs-online.com/webdocs/1621/0900766b81621dcc.pdf. [Accessed 17 06 2018].

[27] FreeSound. [Online]. Available: https://freesound.org/. [Accessed 17 06 2018].

[28] P. S. Australia, “Mean Well Power Supply Hiccup Mode,” 19 December 2017. [Online]. Available: https://www.power-supplies-australia.com.au/blog/mean-well-power-supply-hiccup-mode. [Accessed 17 06 2018].

[29] RS Components, “Declaration of Conformity of RSP-320 series,” Mean Well, 12 January 2017. [Online]. Available: https://docs-emea.rs-online.com/webdocs/1630/0900766b816307e8.pdf. [Accessed 17 06 2018].

[30] National Measurement & Regulation Office, “Guidance to RoHS Directive 2011/65/EU,” November 2015. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/679655/rohs-directive-guidance.pdf. [Accessed 17 06 2018].

[31] MakeItFrom, “Medium Density Fiberboard (MDF),” 13 January 2018. [Online]. Available: https://www.makeitfrom.com/material-properties/Medium-Density-Fiberboard-MDF. [Accessed 17 06 2018].

[32] “Special needs toys,” 2018. [Online]. Available: https://www.special-needs-toys.co.uk/blog/benefits-switch-adapted-play/. [Accessed 17 6 2018].

[33] T. Agarwal, “Freewheeling or Flyback Diode Working and Their Functions,” [Online]. Available: https://www.elprocus.com/freewheeling-or-flyback-diode-circuit-working-functions/. [Accessed 17 06 2018].

[34] Wikipedia, “Flyback diode,” [Online]. Available: https://en.wikipedia.org/wiki/Flyback_diode. [Accessed 17 06 2018].

[35] J. Jacob, in *Power Electronics: Principles and Applications*, Cengage Learning, 2001.

[36] “Pinterest,” [Online]. Available: https://www.pinterest.com/pin/391883605060098981/?lp=true. [Accessed 17 06 2018].

[37] “Purpose of the diode and capacitor in a motor circuit,” January 2014. [Online]. Available: https://electronics.stackexchange.com/questions/95140/purpose-of-the-diode-and-capacitor-in-this-motor-circuit. [Accessed 17 06 2018].

[38] “Snowboy,” [Online]. Available: http://docs.kitt.ai/snowboy/. [Accessed 17 06 2018].

[39] life.augmented, “IMU Datasheet,” November 2014. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/3/7/3/LSM9DS1_Datasheet.pdf. [Accessed 17 06 2018].

[40] S. Kulpa, “How Flippers Work,” 19 1 2012. [Online]. Available: http://stevekulpa.net/pinball/bally_flipper1.htm. [Accessed 17 06 2018].

[41] D. Nedelkovski, “Arduino LCD Tutorial,” How to Mechatronics, [Online]. Available: https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/. [Accessed 17 06 2018].

[42] Ile De France Amusement. [Online]. Available: http://www.idfamusement.com/bobine-williams-bally-fl-11630-xml-262_398_404_409-853.html. [Accessed 17 06 2018].

Appendices

Appendix A – Project Management

Given the time limitations for this project, several tasks had to be performed simultaneously in order to complete it in time. Hence, the sub-components (controls, ball return, flipper bats, central processing unit) were built, tested and redesigned while the chassis (hollow compartments, passage holes for ball, playing field) of the game was manufactured.

Throughout the project the team was split into several sub-groups working on different parts of the design or stages of manufacturing. Whole team weekly meetings were held for making general strategic decisions and sub-group progress updates. The breakdown of the responsibilities is shown below.

-Project manager: Maksim (organisational responsibilities, budget, user communication)

-Manufacturing manager: Emmanuel (managing the manufacturing process)

-Procurement manager: Marta (ordering materials, budget)

Control units sub-groups

1. Word Detection: Emmanuel, Charalambos, Marta, Maksim
    - Research, software selection, installation, training of the model, testing
2. MMG (mechanical)/IMU (myoelectric): Lucille, Beatrice, Linnea
    - Research, collaboration with Dr Raavi Vaidyanathan, Mr Paschal Egan, sensor integration, testing
3. Joystick/Button: Amira, Gionieva, Wesley
    - Research, button selection, adapting for the use within the game

Manufacturing sub-groups

1. CAD & Drawings: Wesley, Beatrice, Amira, Maksim
2. The flipper mechanism: Emmanuel, Charalambos, Lucille, Marta
    - Design and building the flipper assembly, ordering coils and bats
3. The ball return mechanism: Beatrice, Linnea, Maksim
    - Design and building the mechanism, ordering motors and timing belts
4. Power Consumption: Emmanuel, Charalambos
    - Research, ordering the power distribution unit and powering the game
5. Playing field: Marta, Amira, Lucille
    - Design of the playing field (slopes, central pyramid)
6. Outside structure: Maksim, Marta, Charalambos, Gionieva, Wesley, Emmanuel, Beatrice, Lucille, Linnea, Amira
    - Building the metal chassis and attaching the walls, collaboration with the Mechanical Engineering manufacturing lab for metalwork and Mr Daniel Nardini for manufacturing
7. Central processing unit: Beatrice, Amira, Emmanuel
    - Programming the raspberry Pi, Arduinos and integrating the game logic: game initiation, control detection, ball return control, flipper control, sound feedback output, ball counting
8. Sound feedback: Marta, Maksim

    - Choosing the speakers and game sounds, integrating into the game

9. Ball counting mechanism: Charalambos
    - Building the IR-sensor counting mechanism with an LCD screen to display score
10. Electrical Integration: Linnea, Lucille, Maksim, Marta, Charalambos, Emmanuel, Beatrice

The project Gantt chart is presented below (Figure 24) with the milestones marked with blue marks preceded by the Task allocation table (Figure 23) and responsible people indicated.

	Task Name	Planned Start Date	Planned Finish Date	Planned Duration	Assigned	Complete
1	Background Research	09/10/2017	31/10/2017	17 days	Everyone	☑
2	Specifications Definition	26/10/2017	31/10/2017	4 days	Everyone	☑
3	Working on controls in groups (research, ordering parts, etc)	30/10/2017	27/11/2017	21 days	Each control sub...	☑
4	Each group presenting initial results	20/11/2017	20/11/2017	1 day	Each control sub...	☑
5	Orders	20/11/2017	20/11/2017	1 day	Marta	☑
6	Finalizing the controls	20/11/2017	06/12/2017	13 days	Each control sub...	☑
7	Board Detailed Design	04/12/2017	13/12/2017	8 days	Everyone	☑
8	Visit John Chilton School	06/12/2017	06/12/2017	1 day	Everyone	☑
9	Ordering the parts for the board	11/12/2017	19/12/2017	7 days	Marta	☑
10	Christmas Break	18/12/2017	04/01/2018	14 days		☑
11	Prototype of the ball counting	02/02/2018	12/02/2018	7 days	Charalambos	☑
12	Build chassis	02/02/2018	13/02/2018	8 days	Maksim, Beatrice...	☑
13	Manufacture the slopes, CAD, 3D printing	02/02/2018	13/02/2018	8 days	Maksim, Beatrice...	☑
14	Attach the wooden Structure	12/02/2018	14/02/2018	3 days	Charalambos, Mar...	☑
15	Build the ball-return	31/01/2018	23/02/2018	18 days	Beatrice, Maksim...	☑
16	Order and install a power supply	07/02/2018	26/02/2018	14 days	Charalambos, Li...	☑
17	Build the flipper mechanism	31/01/2018	23/02/2018	18 days	Charalambos, E...	☑
18	Assemble the mechanisms	01/03/2018	19/03/2018	13 days	Everyone	☑
19	Poster	01/03/2018	12/03/2018	8 days		☑
20	Working prototype of each control unit	31/01/2018	21/02/2018	16 days	Each control sub...	☑
21	CPU programming	02/03/2018	12/03/2018	7 days	Beatrice, Amira	☑
22	Sound feedback integration	08/03/2018	12/03/2018	3 days	Marta, Maksim, G...	☑
23	Final assembly	08/03/2018	12/03/2018	3 days	Everyone	☑
24	Demonstration day	12/03/2018	12/03/2018	1 day	Everyone	☑
25	Report writing	30/03/2018	19/07/2018	80 days	Everyone	☑

Figure 23 – The table view of the project activities

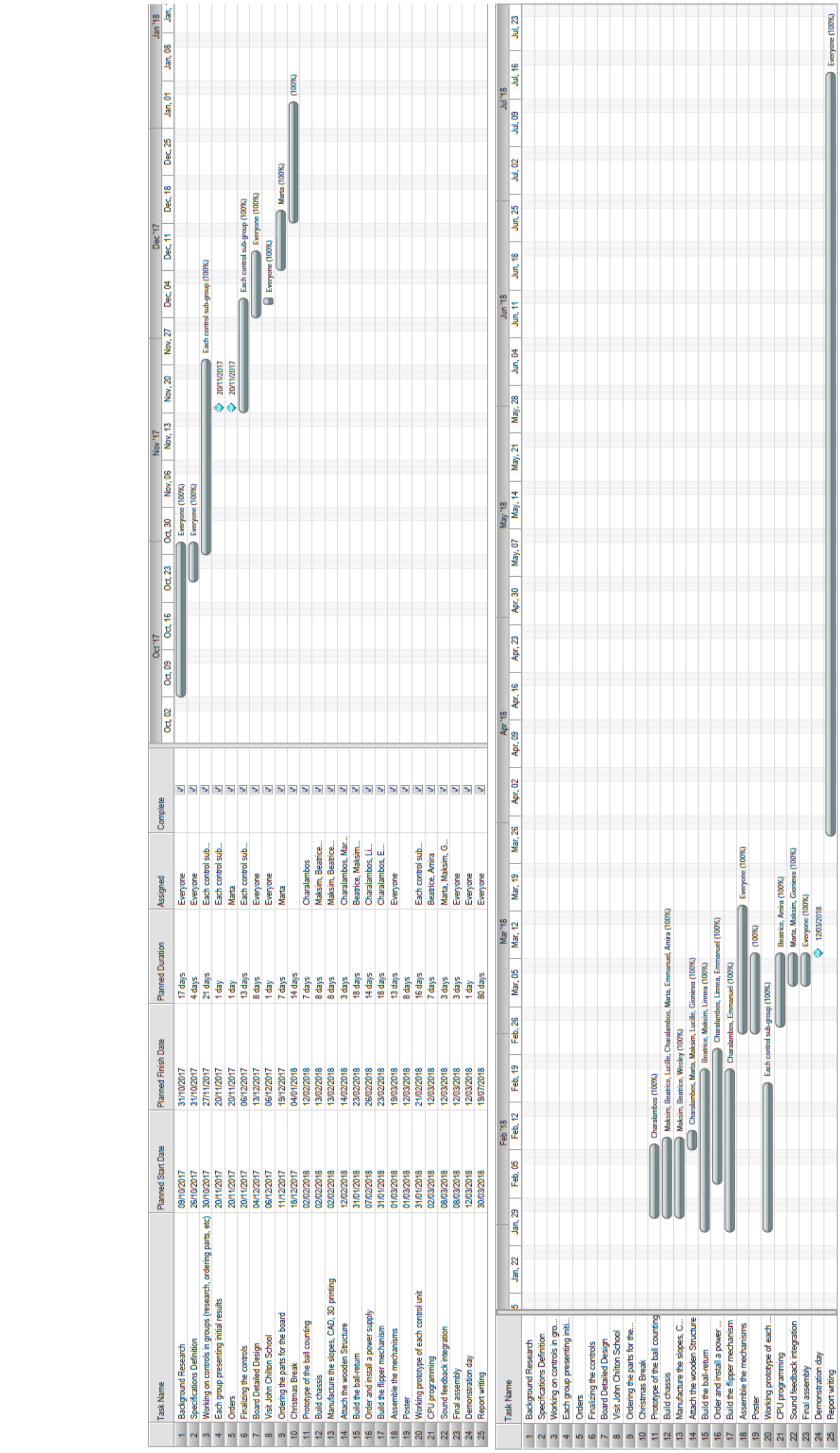


Figure 24. The Gantt chart view of the project

Appendix B – Risk Management

Critical Risk Priority Number

During the risk analysis, each risk or failure is analysed and rated with respect to its severity (S), probability of occurrence (O), and detection rate (D). The rating for each of the three aspects ranges from 1 (low security risk/failure, low probability of occurrence, high detection probability) to 10 (severe injuries or death, high probability of occurrence, no/low probability for detection). The product out of these three ratings is called Risk Priority Number (RPN). In case, the RPN is greater than a critical threshold, preventing measures are required in order to reach a final RPN below or equal to the critical threshold by means of reasonable and justifiable security measures.

A critical **RPN threshold of 75 is defined**.

Table 5 - Severity (S) Key

Rating S	Criteria: Severity of effect	Consequence	Treatment
10	Death	-	-
9	Quadriplegia	Life-long medical care necessary / coma / permanent damage	Hospital stay
8	Amputations, paraplegia, blindness, deafness, traumatic brain injury (severe), fourth-degree burns	Life-long medical care necessary / coma / permanent damage	Hospital stay
7	Complex fractures, open fracture, inner injuries, traumatic brain injury (severe), third-degree burns	Permanent damage possible	Hospital stay
6	Gash, fractures, torn muscles, articular cartilage injury, traumatic brain injury (moderate), second-degree burns	Permanent damage possible	Hospital stay
5	Gash, fractures, torn muscles, articular cartilage injury, traumatic brain injury (mild), second-degree burns	Reversible injury	Hospital stay or ambulant treatment
4	Severe cuts, severe scratches, severe contusions, strains, first-degree burns	Reversible injury	Ambulant treatment or self-treatment
3	Minor cuts, minor scratches, minor contusions, stiff muscles, tension, blisters, excoriations, sickness, first-degree burns	Discomfort during application up to three days after application	Self-treatment
2	Slight sickness, pressure marks	Discomfort	-
1	No harm	-	-

Table 6 - Occurance (O) Key

Rating O	Criteria: Probability of occurrence
10	Occurs or may occur very likely during every use of the session
9	Occurs or may occur likely during every use of the session
8	Occurs in 1 of 5 sessions (less than once a day)
7	Occurs in 1 of 10 sessions (less than once a day)
6	Occurs in 1 of 50 sessions (less than once half a month)
5	Occurs in 1 of 100 sessions (less than once a month)
4	Occurs in 1 of 500 sessions (less than once half a year)
3	Occurs in 1 of 1000 sessions (less than once per year)
2	Occurrence very unlikely
1	Occurrence nearly impossible

Table 7 - Detection (D) Key

Rating D	Criteria: Likelihood of detection by design control
10	No chance of detection
9	Very remote chance of detection
8	Remote chance of detection
7	Very low chance of detection by indirect methods (hardware or software)
6	Low chance of detection by indirect methods (hardware or software)
5	Moderate chance of detection by indirect methods (hardware or software)
4	High chance of detection by indirect methods (hardware or software)
3	High chance of detection by direct or indirect methods (hardware/software)
2	Direct and indirect detection: Hardware or software
1	Direct detection: Hardware or safe software (category 4, performance level e)

Table 8 - Risk Analysis

Assembly	Failure & Effect	S1	O1	D1	RPN before	Preventing measures	S2	O2	D2	RPN after
Eyebrow control unit	Exposure of electrical components which operate at very low current & possible minor discomfort for the user	2	6	1	**12**	Enclosure of the electronics within electrically insulating tape, a headband and a plastic box closed by screws	2	1	1	**2**
	Failure to detect the eyebrow movement	1	9	5	**45**	Compartmentalisation of control units for efficient error checking	1	9	4	**36**
Word Detection Headset	Exposure of electrical components & possible minor discomfort for the user	2	2	1	**4**	Enclosure of the electronics in a plastic box closed by screws	2	1	1	**2**
	"False positive" detection of the hotword go & over activation of the flipper	1	8	1	**8**	Training the Snowboy software with as many voices as possible	1	7	1	**7**
Button control unit	Exposure of electrical components & possible minor discomfort for the user	2	2	1	**4**	Enclosure of electronics in a plastic box closed by screws	2	1	1	**2**
Ball Return Mechanism	The mechanism not being able to rotate at a required speed	1	2	1	**2**	Power to the motors is regulated to ensure it does not fluctuate from the desired speed	1	1	1	**1**

	Ball blocking the mechanism & game being impeded	1	10	1	**10**	Use of balls with a low coefficient of friction, and smoothing of the inner wall in contact with the ball as it rotates. High tension in the timing belt maintaining sufficient force during rotation	1	8	1	**8**
Ball counting mechanism	Failure to detect the ball & thus not increasing the corresponding counters	1	9	1	**9**	Opaque balls only must as any semi-transparent balls will not be detected by the mechanism	1	1	1	**1**
Power Supply and distribution unit	Overheating of the power supply & the whole game	2	9	1	**18**	The game should by switched off to cool down when the user notices that it is warm.	2	2	1	**4**
	Exposure of 40V circuitry within the game & contact with other conducting game components	7	2	1	**14**	This circuitry is covered with electrically insulating tape and is contained within the board which cannot be opened by users.	7	1	1	**7**
	Exposure of the metallic case parts of the power supply & connection with the 60V voltage	8	2	1	**16**	The power supply is grounded with all conducting elements in the structure of the board and is itself enclosed within a plastic box separate from the rest of the electronics.	8	1	1	**8**
	Malfunction of the device & shorts within the circuitry	1	2	4	**8**	Fuses installed to prevent breaking components.	1	1	2	**2**
	Outer plug not functioning & failure to power the game	10	2	1	**20**	Standard outer plug used.	10	1	1	**10**

LCD Score Display	Malfunction & failure to display to the score	1	7	1	**7**	Wires soldered to the screen should be braided type and should be reinforced with electrical tape to prevent open circuit.	1	3	1	**3**
Speakers and sound feedback	The sound level exceeding the recommended & discomfort to the user	6	2	1	**12**	The speakers used have a standard maximum volume below the threshold for user discomfort	1	1	1	**1**
Internal game circuitry	Exposure of the electrical components & their contact with the wooden parts	2	2	5	**20**	Insulating tape used	1	1	5	**5**
Central Processing unit	Overheating of the central Raspberry Pi & contact with the internal parts	1	10	8	**80**	Thermal tape used. The holes introduced in the walls for air circulation.	1	5	8	**40**
	Malfunction of the Raspberry Pi & game not functioning	1	7	1	**7**	The game must be switched off after some period to allow the Raspberry Pi to cool.	1	6	1	**6**
External game structure	Sharp edges or not-smooth surfaces exposed & possible discomfort to the user	2	5	1	**10**	Edges must be filed and abrasive surfaces covered with smooth material.	1	2	1	**2**
Internal game structure	The wooden walls not being able to withstand the weight of the game & breaking of the game or the playing field	1	2	1	**2**	Strong chassis structure implemented.	1	1	1	**1**

Appendix C – Ethics

Ethical concerns of various types were raised throughout the entire project work. The aim of this project was to adapt a board game for children with motor disabilities. It involved the development of different control systems as well as user trials to make sure those were suitable. The main ethical issues encountered were thus related to human testing. The other concerns were related to discrimination and competence, data usage and the respect of Intellectual Property.

Carefulness and openness

During the visits to the school, it was necessary to be strict and open with measuring, estimating and calculating risks especially involving electrical and mechanical safety. The risks were always overestimated to a reasonable safety factor.

Honesty and openness with the user about the performance and limitations of the control systems was important: they were to be explained clearly in a user manual so that undesired reactions of angriness or disappointments from the children can be avoided.

Social Responsibility

One of the main aspects of the project was to make contact with John Chilton School, first with Jane Hales, Physical Education Teacher and Jonathan Wijnen the Assistive Technologist. Understanding the needs of the children in order to respond best to their requirements was a key goal at the beginning of the project, thus visits at the school were necessary to have feedback from the users. The main ethical objectives were to adopt appropriate behaviour with the children and explain simply yet clearly the rules of the game and the use of the control systems.

Testing of the prototype by the children was always done in a safe manner (see risk assessment), and on a voluntary basis. This project concerns only humans, thus no animal testing was performed.

Non-discrimination and Competence

Three different systems were used to control the game to allow the widest range of children to play the game, regardless of their disability. As it might be difficult to use the word detection control system for children with speech impairment, other control systems like the push-button were developed.

For the word detection control system, words similar to the hotword "go" activated the flippers against the will of the player (see 5.1. Limitations).

For the eyebrow control system, the code controlling the activation of the flippers was written such that the flippers would get activated more often than it should rather than less (see 3.3. Eyebrow Control)

Each control system has a different time delay between the user input and the output of the flipper bats. Further improvement would be needed to adjust this difference between the three controls such that the game is fair to play and no player is disadvantaged by using one particular device.

Data usage

The question of data storage and usage can be raised about the eyebrow control system or the word detection control system. This was not an issue as data is translated within the Arduinos into a binary output to activate the flippers. Thus, personal data is not stored within any of the systems used.

Similarly, during the user trials, tests were done to observe the direct output response of the systems and thus data was not stored to be analysed later.

Respect of Intellectual Property

For the word detection control system, the open-source software Snowboy was used to detect the chosen hotword. The training of the word was done online, by voice recording on the Snowboy platform. The ethical aspects raised by the trainings are covered by the software's ethical policy.

The software Snowboy was downloaded freely and used on Arduino. During every explanation or demonstration, the name of the software was mentioned and its importance in the game was clearly defined.

The MMG sensor used in the eyebrow control system was given in November 2017 by Dr Ravi Vaidyanathan, researcher in the Mechanical Engineering Department at Imperial College London. His help and donation were always specified when explaining the role of this control system. His name can be found in the acknowledgements of every written work about this project.

Environment

The materials chosen to build the game are commonly used and the only environmental concerns lie in the disposal of the non bio-degradable and non-recyclable parts. Detailed instruction and warnings about the disposal of such parts should be included in the user manual.

The sound of the speakers was adjusted not to cause noise pollution (see 2.3. Technical Requirements).

Energy consumption and the effective usage of power were considered. Friction was reduced wherever possible and the idea of introducing an EOS switch in the future (see E.3) would allow to further decrease power loss.

Appendix D – Manufacture

The manufacturing process of the board game prototype of the board consisted of three main stages: creating the control units, manufacturing the mechanical components, programming and assembling the electronic components in the board. The manufacturing Gantt chart is presented below (Figure 25). Several stages of the manufacturing process are depicted in the sequence of pictures (Figures 26 – 30).

The three control units were developed by each of the focus-teams as specified above in the final design section of this report.

The metal chassis manufacturing was performed by Mr Daniel Nardini who cut the steel tubes and by the members of the team who drilled, bolted and assembled in the metal manufacturing lab.

The wooden walls and the 3D printed parts were assembled in the Bioengineering lab.

The flipper plungers were manufactured by Mr Daniel Nardini. The members of the group drilled the flipper bats and assembled them together.

Testing and casing of the power supply and electronic components was performed in the Bioengineering electronics lab under the supervision of Mr Paschal Egan.

The final assembly and testing were performed in the Imperial College Robotics Society lab.

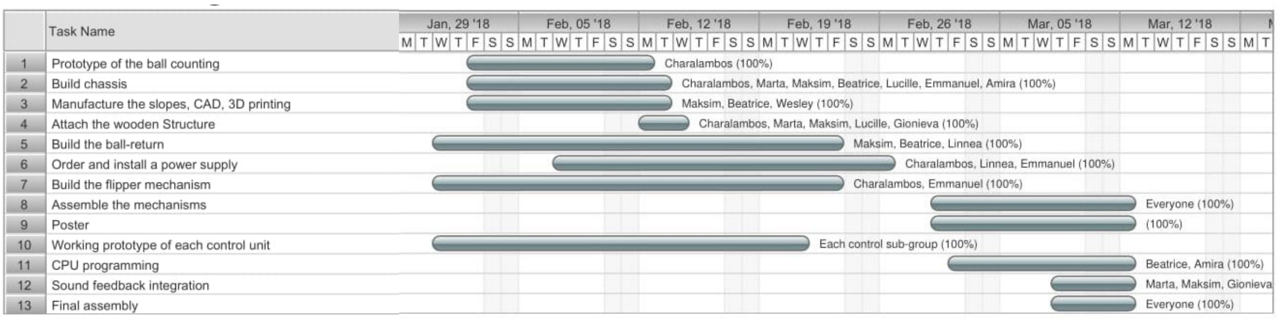


Figure 25 – The Gantt chart of the manufacturing process

Figure 26 – The manufacturing process of the chassis and bottom MDF base

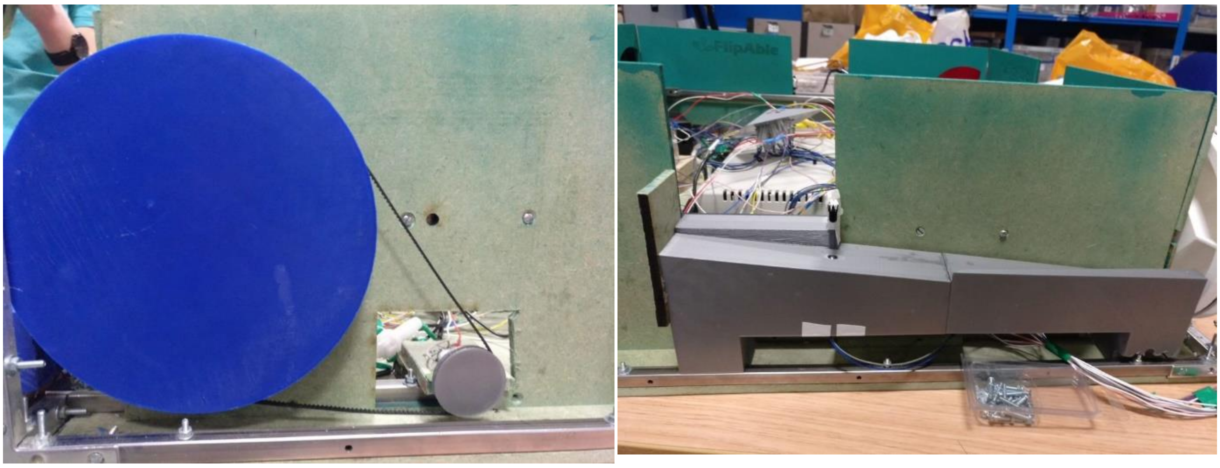


Figure 27 – The 3D printed ball-return wheel, pulley and the goal rolling slope

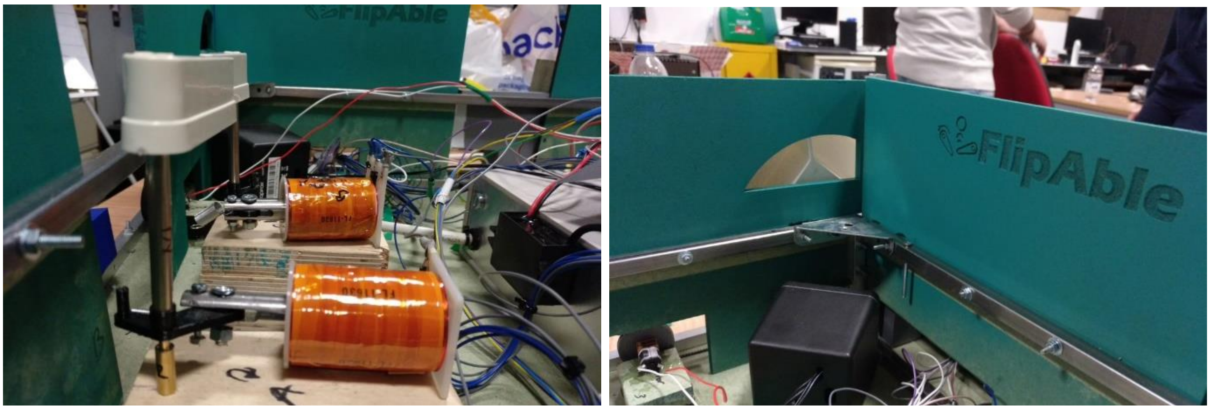


Figure 28 – The flipper assembly and the corner of the game from the inside with the speaker, ball return motor and ball return exit hole

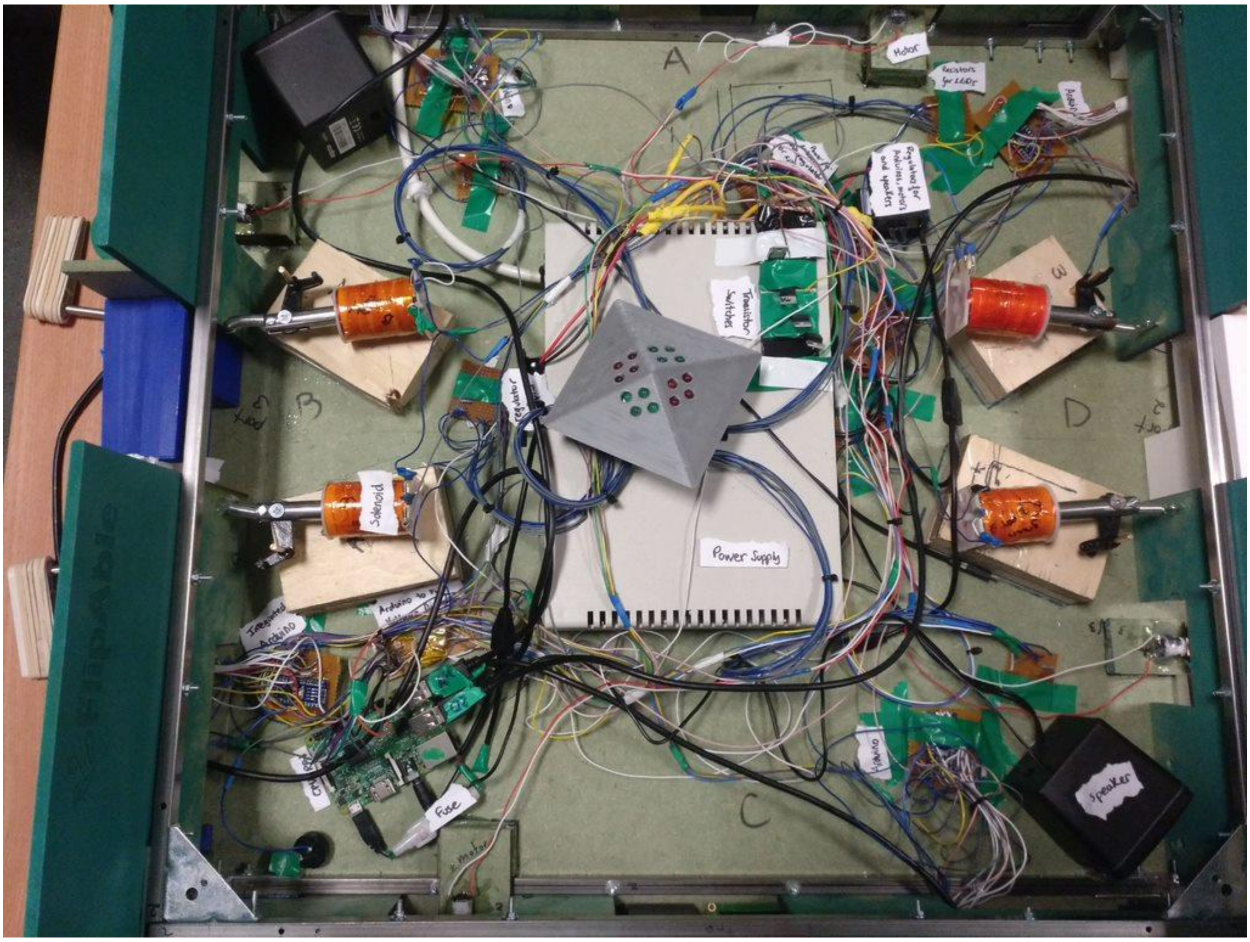


Figure 29 – The internal configuration of the game at latest stages of manufacture

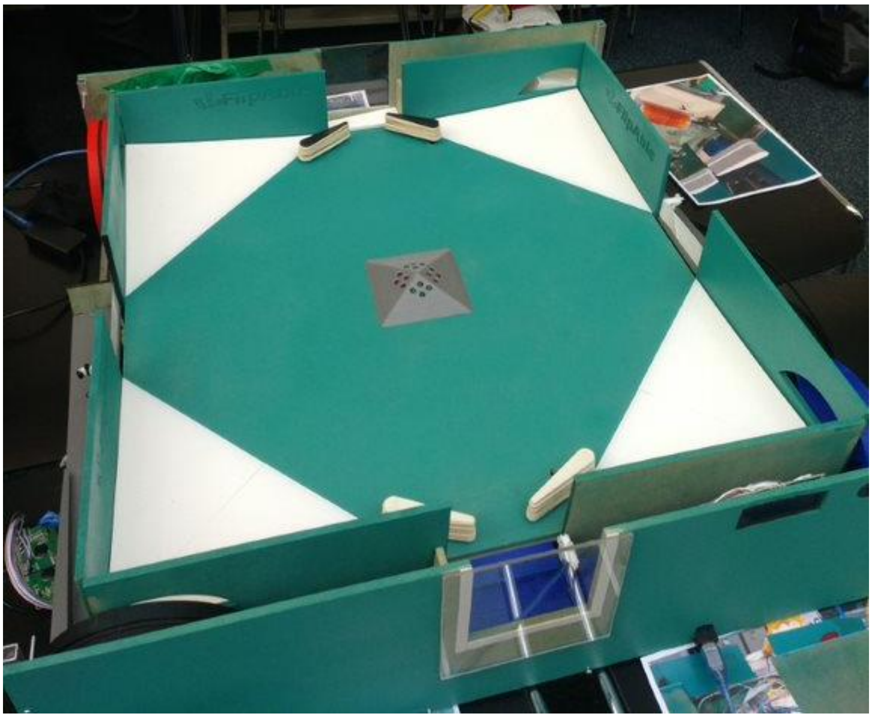


Figure 30 – The final prototype

Appendix E - Detail Design Drawings and Electrical Schematics

E.1 - LCD Circuitry

Each of the four LCDs used has 16 pins, which, in this case, are connected as seen in Figure 31. The details about the pins are shown in figure 32 and table 9.

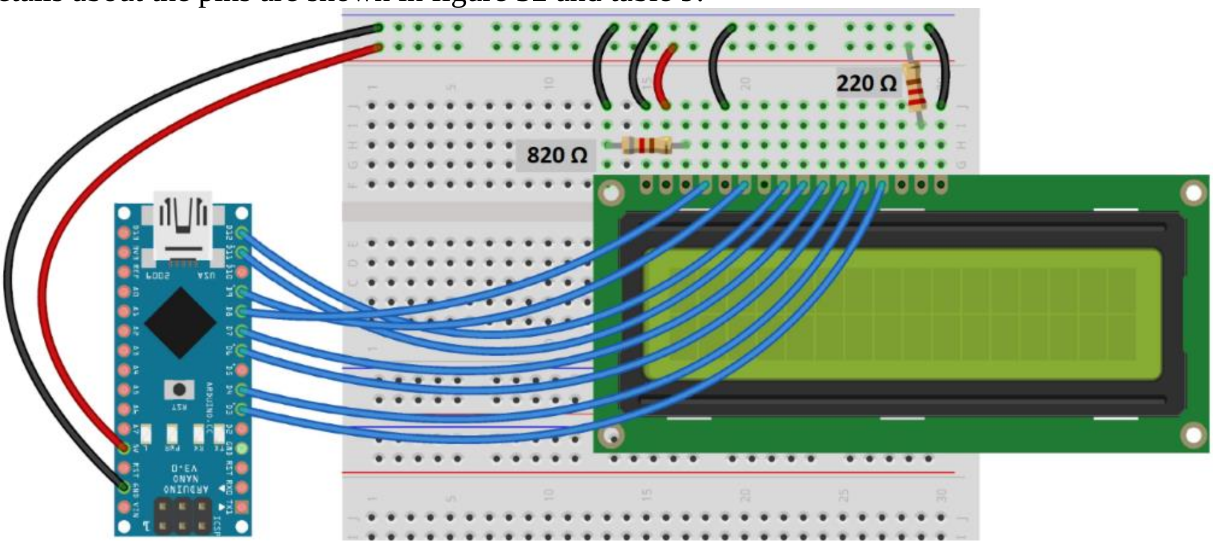


Figure 31 – Diagram showing the physical layout of the LCD screen, the Arduino board and the connections between them on each side of the board game [41]

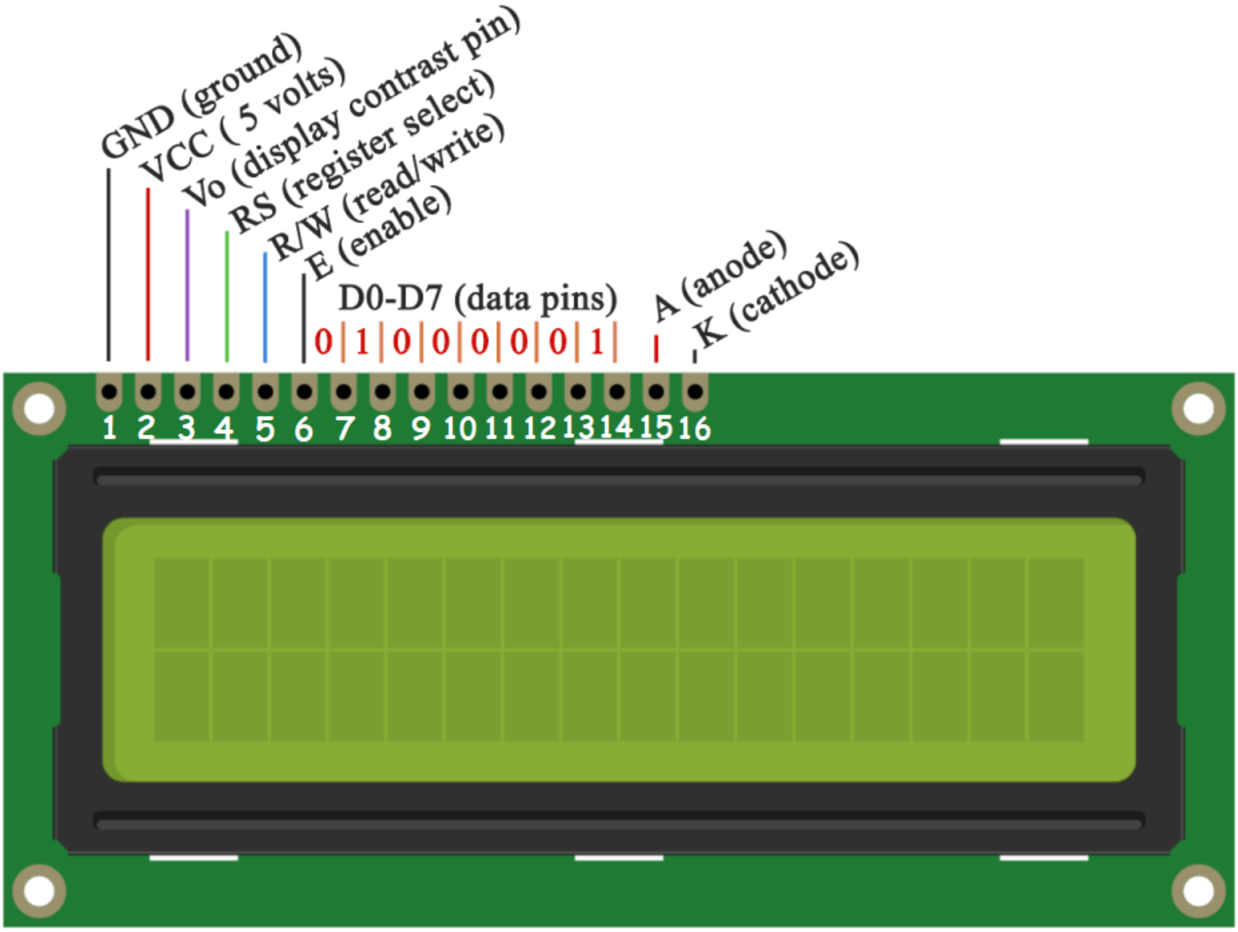


Figure 32 – Diagram showing pin configuration on LCD screen

Table 9 - Detailed explanation of the connections

Pin number	Connection
Pin 1	GND connected to ground pin on Arduino
Pin 2	VCC connected to the 5 volts pin on the Arduino, responsible for supplying voltage to LCD screen
Pin 3	Vo pin is responsible for controlling the contrast of the display. This was connected to ground through an 820 Ω resistor. Initially contrast was controlled by a three-terminal potentiometer whose terminals were connected to Vo pin, ground and 5 volts pins on Arduino. In the final design, resistance between the terminals connected to Vo and ground was measured by an ohmmeter to determine the resistor value to be used. This ensured that screen has constant contrast as potentiometer's resistance value was vulnerable to any physical movement of components due to transport of the game
Pin 4	RS pin or register select pin is used for selecting whether commands or data would be sent to the LCD. This was connected to digital pin 8 of Arduino allowing commands to be sent to the LCD, such as setting the location at which subsequent text written will be displayed, clearing the display, turning off the display etc.
Pin 5	R/W pin which selects read or write mode on the LCD. Connecting this pin to GND pin on Arduino enables the write mode allowing writing or sending commands and data to the LCD
Pin 6	E pin enables writing to the registers and this is connected to digital pin 9 of Arduino
Pin 7-14	Data pins from D0 to D7. Through these pins 8-bit data (corresponding to characters in ASCII table) is sent from the Arduino to the Screen to display score
Pin 15 and 16	Pins A and K respectively, are used for connecting the LED backlight of the screen

E.2 - Use of Flyback Diodes to prevent circuit from sparking

Importance of flyback diodes is illustrated (Figure 33) through a simple circuit diagram involving a power supply, a coil (inductor), a minor resistance (due to resistance of the wires) and a switch, in the absence of flyback diodes [33] [34] [35].

Once the switch is closed, there is an increase in current flowing through the coil given by $\frac{dI}{dt} = \frac{V}{L}$, inducing a back EMF (voltage) across it to oppose this change in current. It should be noted that in the equation above, I represents electric current, t time, V voltage and L inductance. As a result, energy from the power supply is slowly being stored in the coil's magnetic field and net voltage drop across inductor decreases. After the inductor has completely charged up and reached steady state, it behaves like a short circuit as shown in Figure 33a with zero net voltage drop, so all voltage of power supply appears across the minor resistance.

When the switch is open, the inductor resists this abrupt decrease in current by inducing an initially very large voltage across it (higher than supply's output voltage), according to $V = L\frac{dI}{dt}$. This voltage is in the same direction as the one across the power supply that is produced from the magnetic field energy stored in the inductor. This causes the end of the switch that is connected to positive terminal of the power supply, to be at +48V and the other end to be at a voltage much less than -48V. As shown in Figure 33b, this large voltage difference creates an electric arc or sparks across the switch contacts until all the energy stored in the inductor is dissipated as heat, which can cause damage to the switch.

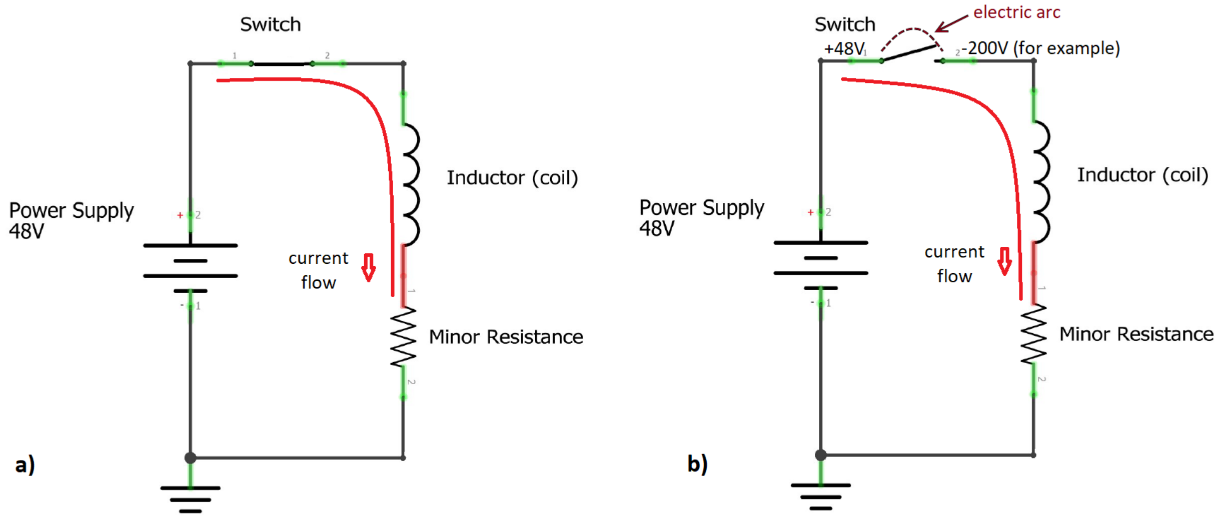


Figure 33 – Schematic diagrams of the circuit including power supply, inductor, minor resistance, and switch in the absence of flyback diode. a) Illustration of current flow through circuit when switch is closed. b) Illustration of electric arc between switch contacts.

Using flyback diodes, connected in parallel with the inductive element, prevents the formation of electric arcs across the switch. Diodes only conduct current once the switch is opened, since otherwise they are reverse biased, so they do not affect operation of circuit but only help during the time when switch opens. These flyback diodes provide a path for current flow, causing voltage drop across them as current flows through, allowing the inductor's energy to dissipate for a few milliseconds.

E.3 – Flipper circuitry explained

The circuit shown below in Figure 34a, is implemented in standard flipper machines by various manufacturers. This ensures both high power supplied to each solenoid for the plunger to be pulled with high force, so that the bat is extended fully faster. In the case there is impact with a ball, this will ensure that the ball has a high acceleration so that the game is more dynamic. At the same time, the EOS switch used provides safety to solenoid from overheating.

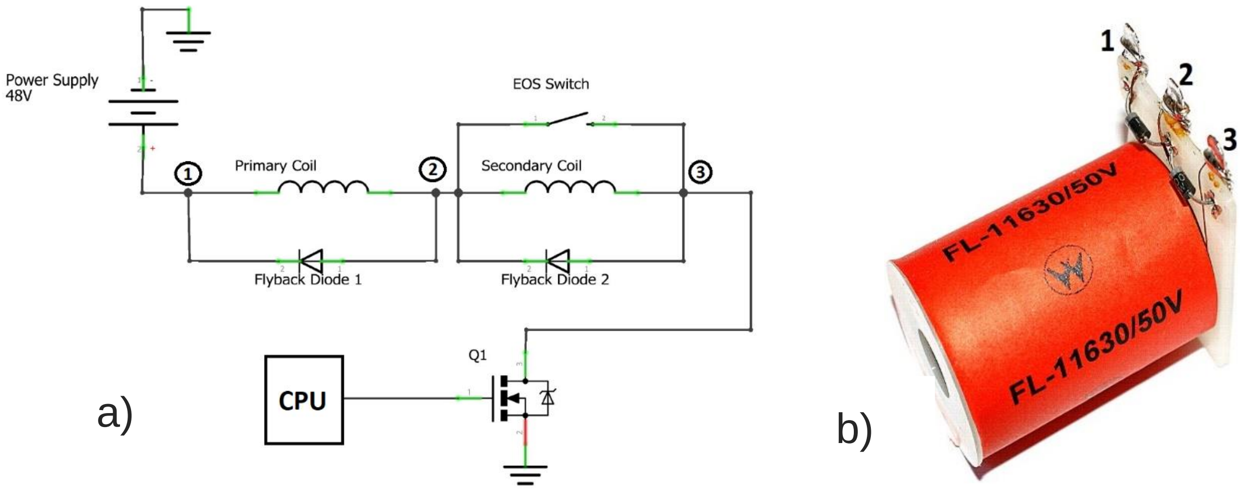


Figure 34 – a) Detailed schematic diagram of each flipper solenoid, with numbered nodes representing the three terminals of the device [40]. Each solenoid is made of two coils and is controlled by CPU through nMOS transistor (component Q1) b) Diagram [42] showing the location of these terminals on the actual device.

An EOS switch is a special type of switch made of two metal conductors, which make a contact when the flipper is in its resting state or partially energized. Contact is lost only in the case of the flipper being fully energized, creating an open circuit, as illustrated in Figure 35.

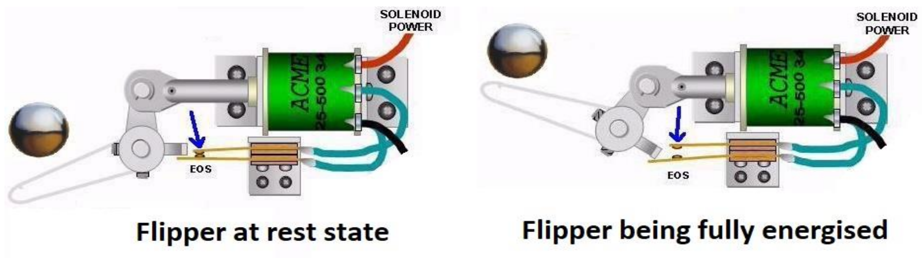


Figure 35 – Diagram [36] illustrating the action of EOS switch for rest state and fully energized state

At the rest state, or when the solenoid is partially energized, the EOS switch is closed, causing the secondary coil to be by-passed. This allows current to flow only through the primary coil, thereby producing the strong pulling effect on flipper plunger. The input resistance of the solenoid in this case is only 5.1 Ω, so power consumed by the solenoid is 451W (Even though the two solenoids on each side were connected in parallel, therefore the power consumed by each is 226 W). When the flipper reaches its fully energized state and the flipper bat reaches the end of its stroke, the protrusion on the coupling part of the assembly opens the EOS switch, forcing current to flow through both the primary and secondary coils. This increases the input resistance of the solenoid to a value of 166.5 Ω, thus decreasing the current flow and power consumption to a total of 14 W, which allows the flipper to be continuously in its fully energized mode in the case of a fault, without the risk of overheating and damaging the device. Once the plunger returns to its original position by the passive elastic properties of the spring attached to it, the EOS switch conducts current and the device returns to its rest state.

If EOS switch is not available, an alternative modification would have been a short-circuit connection between terminals 2 and 3. In that case the power consumed will be constantly at 226 W by each solenoid providing enough energy for pulling force on plunger, but at the same time, the risk of overheating would increase and the board game itself would have high energy consumption and thus its operation is more expensive.

The choice of modification to be used was determined so that it maximises the safety for the user as well as for electrical and electronic components in the game. For this reason, the modification discussed in section 3.7.2 was implemented.

E.4 - Circuitry Involving DC Motors

The circuit implemented for controlling the DC motors used in the Ball Return mechanism (section 3.5) is shown in figure 36. The circuit involves 4 DC Motors (951D Series) connected in parallel, an nMOSFET transistor controlling motor operation, a 1nF capacitor in parallel with all motors and a flyback diode. Motors are supplied with the voltage coming from the 5V output of the voltage regulators.

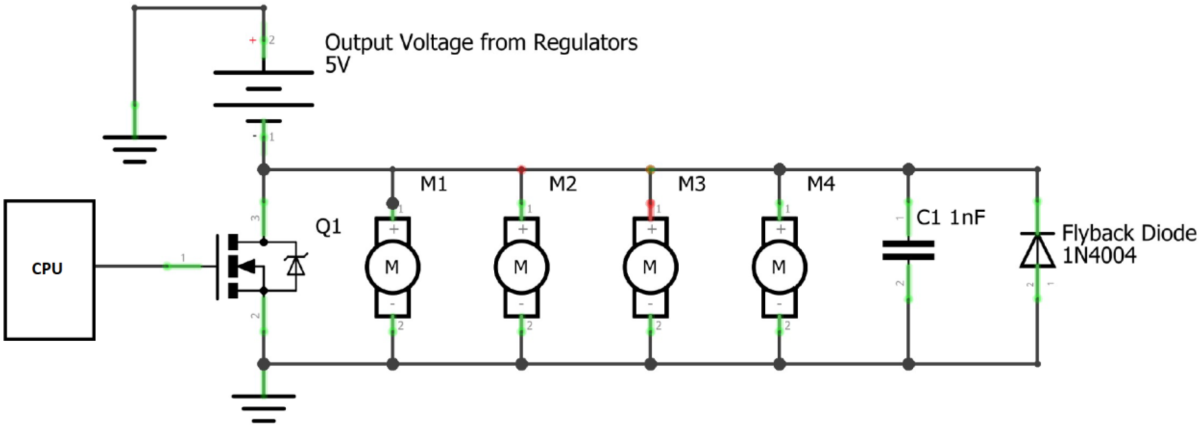


Figure 36 – Schematic Diagram of circuit implemented for Motors operation involved in Ball Counting Mechanism. In the diagram above Q1 represents the nMOSFET transistor used and M1-M4 the four DC motors used.

The purpose of the 1nF capacitor, connected in parallel with the motors, is to filter electrical noise (interference radiation) created from the motor and smooth out current fluctuations during motor's start up. This prevents current spikes from affecting other components in the circuit. [37]

Additionally, a flyback diode across the motors prevents spark formation as explained in Appendix E.2 section since a motor is partially an inductor.

E.5 – Eyebrow Control connection with Arduino and IMU

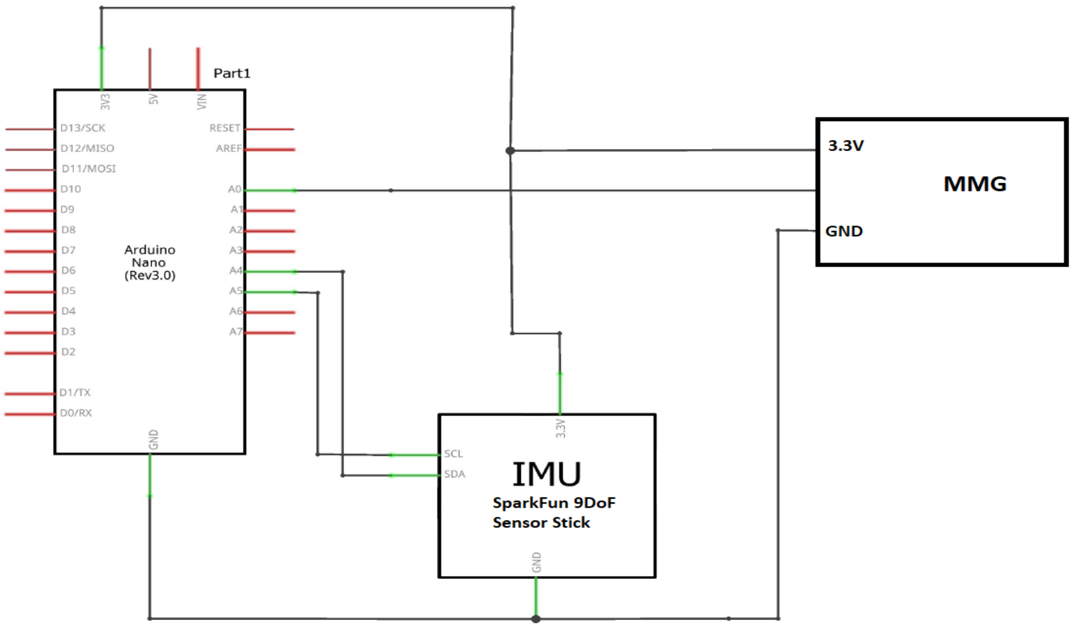


Figure 37 – Schematic Diagram of connections between MMG, Arduino Nano and IMU (SparkFun 9DoF). Both MMG and IMU operate with 3.3V [39] supplied from Arduino. MMG provides analogue input to Arduino through pin A0 and IMU through pins A4 and A5.

E.6 – Word Detection circuitry

Word Detection unit requires connection between a Raspberry Pi0 and an Arduino Nano. Connections between them are illustrated in Figure 38 below.

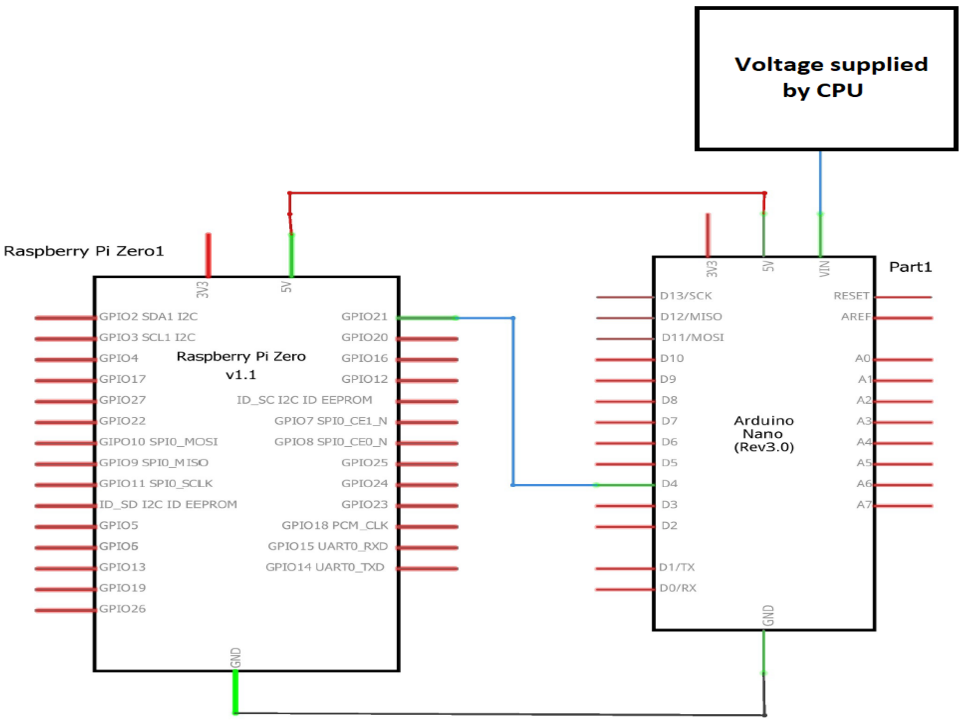


Figure 38 – Schematic Diagram of connections between Arduino Nano and Raspberry Pi0 used for WD unit. Input voltage for Raspberry Pi0 is provided by Arduino and input voltage to Pi0 is provided by the CPU once the control is connected to the USB port on one of the sides of the board.

Appendix F – Complete codes

F.1 – CPU main code:

```
#echo Running at boot
#sudo python /home/pi/Desktop/CPU/CPU_SOUND_COUNT.py


import subprocess
import os
import sys
import time
import RPi.GPIO as GPIO
import serial

master_out = 23 #pin 16 to 'master' arduino nano D5, for standby (HIGH) and resetting (LOW) commands to counters
master_in = 22  #pin 15 from 'master' arduino nano D4, for gameover (LOW) cue from counters

GPIO.setmode(GPIO.BCM)

GPIO.setup(master_out,GPIO.OUT)
GPIO.setup(master_in,GPIO.IN)


while True:

   GPIO.output(master_out, GPIO.LOW)                              #tells arduino to reset counters


   timing = 10

   p1 = 1
   p2 = 1
   p3 = 1
   p4 = 1

   while timing:
      if ((not os.system("ls /dev/ttyPORT2")) and p1):                    #check if a controller is plugged into the port
         PORT2 = serial.Serial('/dev/ttyPORT2',115200)
         player1 = subprocess.Popen(["python", "/home/pi/Desktop/CPU/PORT2Pin40.py"])   #PORT*Pin*.py files contain subprocess for flipping the bats from controllers
         p1 = 0                              #prevent loop from opening a subprocess for this port more than once
      if ((not os.system("ls /dev/ttyPORT3")) and p2):
         PORT3 = serial.Serial('/dev/ttyPORT3',115200)
         player2 = subprocess.Popen(["python", "/home/pi/Desktop/CPU/PORT3Pin13.py"])
         p2 = 0
      if ((not os.system("ls /dev/ttyPORT4")) and p3):
         PORT4 = serial.Serial('/dev/ttyPORT4',115200)
         player3 = subprocess.Popen(["python", "/home/pi/Desktop/CPU/PORT4Pin36.py"])
         p3 = 0
      if ((not os.system("ls /dev/ttyPORT5")) and p4):
         PORT5 = serial.Serial('/dev/ttyPORT5',115200)
         player4 = subprocess.Popen(["python", "/home/pi/Desktop/CPU/PORT5Pin22.py"])
         p4 = 0
      timing = timing - 1
      time.sleep(1)                              #delay gives time for players to plug in

   if not p1 or not p2 or not p3 or not p4:                    #at least one controller must be plugged in
```

```
    subprocess.Popen(["aplay", "/home/pi/Desktop/CPU/start_game_sound_low_bitrate.wav"])    #start music

    game_on = GPIO.input(master_in)                                    #start listening for game over cue (ie. at least
one side's counter went over the limit)
    while(game_on):                                                    #this is the loop the script stays in for most of the
game
        game_on = GPIO.input(master_in)

    GPIO.output(master_out, GPIO.HIGH)                                 #tells arduino to standby

    if p1 == 0:                                                        #stop the bats from flipping
        player1.kill()
    if p2 == 0:
        player2.kill()
    if p3 == 0:
        player3.kill()
    if p4 == 0:
        player4.kill()


    subprocess.Popen(["aplay", "/home/pi/Desktop/CPU/end_game_sound.wav"])


    reset = 0

    while not reset:                                                   #users exit this loop/reset the game by all activating
their controller
        if p1 == 0:
            p1 = int(PORT2.readline())
        if p2 == 0:
            p2 = int(PORT3.readline())
        if p3 == 0:
            p3 = int(PORT4.readline())
        if p4 == 0:
            p4 = int(PORT5.readline())
        reset = p1 and p2 and p3 and p4
        if p1 ==0:
            PORT2.reset_input_buffer()
        if p2 ==0:
            PORT3.reset_input_buffer()
        if p3 ==0:
            PORT4.reset_input_buffer()
        if p4 ==0:
            PORT5.reset_input_buffer()

sys.exit(0)
```

F.2 – CPU subprocess

```
# -*- coding: utf-8 -*-
#Previous line allows script be read by subprocess.open()
import subprocess
import RPi.GPIO as IO          # calling header file for GPIO’s of PI
import time                    # calling for time to provide delays in program

import serial

PORT2 = serial.Serial('/dev/ttyPORT2',115200)
```

```
IO.setmode(IO.BOARD)
IO.setup(40,IO.OUT)

while True:

   while PORT2.in_waiting:
      PORT2.readline()
      IO.output(40,1)
      subprocess.Popen(["aplay", "/home/pi/Desktop/CPU/flipper_sound.wav"])
      time.sleep(0.3) #flipper on
   IO.output(40,0)
```

F.3 – Ball counting central processing unit code

```
/*
  Arduino Mario Bros Tunes
  With Piezo Buzzer and PWM

  Connect the positive side of the Buzzer to pin 3,
  then the negative side to a 1k ohm resistor. Connect
  the other side of the 1 k ohm resistor to
  ground(GND) pin on the Arduino.

  by: Dipto Pratyaksa
  last updated: 31/3/13
*/

/**************************************************
 * Public Constants
 **************************************************/

#define NOTE_C7  2093
#define NOTE_E7  2637
 #include <LiquidCrystal.h>

 LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
 int last=0;
 int counter = 0;

 #define melodyPin 7
//Mario main theme melody
int melody[] = {
  NOTE_E7, NOTE_E7, 0, NOTE_E7,
  0, NOTE_C7, NOTE_E7
};
//Mario main them tempo
int tempo[] = {
  12, 12, 12, 12,
  12, 12, 12
};

 void setup()
 {
  pinMode(7, OUTPUT);//buzzer
  pinMode(LED_BUILTIN, OUTPUT);//led indicator when singing a note

  counter = 0;

  lcd.begin(16, 3);
  lcd.print("Counting");
```

```
 lcd.setCursor(0, 1);
 lcd.print("Mechanism");
 delay(3000);
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Goals Received:");
 lcd.setCursor(0, 1);
 lcd.print(counter);

 pinMode(2,OUTPUT);  //pd
 pinMode(3,OUTPUT);  //led
 pinMode(4, OUTPUT); //reset to pi and nanos (game on/off)
 pinMode(5, INPUT); //reset from pi
 pinMode(6, INPUT); //sensor read
 pinMode(14,INPUT); //counter check from nano1
 pinMode(15,INPUT);  //counter check from nano2
 pinMode(16,INPUT);  //counter check from nano3
 digitalWrite(2,HIGH);       //supply 5 volts to photodiode
 digitalWrite(3,LOW);      //set the led in off mode (initial condition)
 digitalWrite(4, HIGH); //start with the game on
 Serial.begin(9600);         //setting serial monitor at a default baund rate of 9600
}

void loop()
{
 lcd.display();
 lcd.setCursor(0, 1);
 int val=digitalRead(6);  //variable to store values from the photodiode
 Serial.println(val);         // prints the values from the sensor in serial monitor
 delay(1);                    //For better stability



 digitalWrite(4, HIGH);  //game on for children and pi
 Serial.println(counter);

 if (val && !last)
 {
    counter++;
    lcd.print(counter);
    digitalWrite(3, HIGH);
    sing(1);
 }
 last=val;
 digitalWrite(3, LOW);

 if(digitalRead(14) || digitalRead(15) || digitalRead(16) || counter >=20)
 {
   digitalWrite(4,LOW);

   delay(3000);

   while(digitalRead(5)); //wait for players to restart

   last=0;
   val=0;

   counter = 0;

   lcd.clear();
   lcd.setCursor(0, 0);
```

```
    lcd.print("Goals Received:");
    lcd.setCursor(0, 1);
    lcd.print(counter);
  }
}

int song = 0;

void sing(int s) {
  // iterate over the notes of the melody:
  song = s;


    Serial.println(" 'Mario Theme'");
    int size = sizeof(melody) / sizeof(int);
    for (int thisNote = 0; thisNote < size; thisNote++) {

      // to calculate the note duration, take one second
      // divided by the note type.
      //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
      int noteDuration = 1000 / tempo[thisNote];

      buzz(melodyPin, melody[thisNote], noteDuration);

      // to distinguish the notes, set a minimum time between them.
      // the note's duration + 30% seems to work well:
      int pauseBetweenNotes = noteDuration * 1.30;
      delay(pauseBetweenNotes);

      // stop the tone playing:
      buzz(melodyPin, 0, noteDuration);


  }
}

void buzz(int targetPin, long frequency, long length) {
  digitalWrite(LED_BUILTIN, HIGH);
  long delayValue = 1000000 / frequency / 2; // calculate the delay value between transitions
  //// 1 second's worth of microseconds, divided by the frequency, then split in half since
  //// there are two phases to each cycle
  long numCycles = frequency * length / 1000; // calculate the number of cycles for proper timing
  //// multiply frequency, which is really cycles per second, by the number of seconds to
  //// get the total number of cycles to produce
  for (long i = 0; i < numCycles; i++) { // for the calculated length of time...
    digitalWrite(targetPin, HIGH); // write the buzzer pin high to push out the diaphram
    delayMicroseconds(delayValue); // wait for the calculated delay value
    digitalWrite(targetPin, LOW); // write the buzzer pin low to pull back the diaphram
    delayMicroseconds(delayValue); // wait again or the calculated delay value
  }
  digitalWrite(LED_BUILTIN, LOW);

}
```

F.4 – Ball counting simple Arduino nano code

```
/*
  Arduino Mario Bros Tunes
  With Piezo Buzzer and PWM
```

```
  Connect the positive side of the Buzzer to pin 3,
  then the negative side to a 1k ohm resistor. Connect
  the other side of the 1 k ohm resistor to
  ground(GND) pin on the Arduino.

  by: Dipto Pratyaksa
  last updated: 31/3/13
*/

/**************************************************
 * Public Constants
 **************************************************/

#define NOTE_C7  2093  //note frequencies
#define NOTE_E7  2637


 #include <LiquidCrystal.h>

 LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

 int counter=0; //goal count
 int last=0; //flag used to avoid ove rcounting

#define melodyPin 5
//Mario main theme melody, our goal sound
int melody[] = {
  NOTE_E7, NOTE_E7, 0, NOTE_E7,
  0, NOTE_C7, NOTE_E7
};
//Mario main them tempo
int tempo[] = {
  12, 12, 12, 12,
  12, 12, 12
};

void setup(void)
{
  pinMode(5, OUTPUT);//buzzer
  pinMode(LED_BUILTIN, OUTPUT);//led indicator when singing a note

  lcd.begin(16, 3); //initialising lcd, displaying initial message
  lcd.print("Counting");
  lcd.setCursor(0, 1);
  lcd.print("Mechanism");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Goals Received:");
  lcd.setCursor(0, 1);
  lcd.print(counter);

  pinMode(2,OUTPUT);  //pd
  pinMode(3,OUTPUT);  //led
  pinMode(4,INPUT);  //sensor read
  pinMode(6, INPUT); //game on/off from D4 of master arduino
  pinMode(14, OUTPUT);  //output gameover to master

  digitalWrite(2,HIGH);       //supply 5 volts to photodiode

  digitalWrite(3,LOW);      //turn led off (initial condition)
```

```
  digitalWrite(14,LOW);      //signal to master arduino that tells it this player's side hasn't lost yet
  Serial.begin(9600);         //setting serial monitor at a default baund rate of 9600

}
void loop()
{
  lcd.display();
  lcd.setCursor(0, 1);
  int val=digitalRead(4);  //variable to store value sampled from the photodiode (0 if no ball passing through beam)
  delay(1);                   //For better stability

  if(val && !last)
  {
     digitalWrite(3, HIGH);    //led on
     counter=counter+1;
     lcd.print(counter); //display new score
    sing(1); //sound
  }

  else          //If obstacle is not in range
  {
                      //Once a zero is encountered, temp is reset to 0.
    digitalWrite(3,LOW);      //led will be in OFF state

  }
  last=val;

  if(counter>=20) digitalWrite(14, HIGH); //This side lost! Send game over signal to master

  if(digitalRead(6)) //Check master arduino
  {
     digitalWrite(3, LOW);  //led off
  }
  else  //A side has lost and master is signaling it's game over
  {
    while(!digitalRead(6)); //wait for players to restart

    counter=0;
    last=0;
    val=0;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Goals Received:");
    lcd.setCursor(0, 1);
    lcd.print(counter);
    digitalWrite(14, LOW);
  }
}

//song function
int song = 0;

void sing(int s) {
  // iterate over the notes of the melody:
  song = s;


    int size = sizeof(melody) / sizeof(int);
    for (int thisNote = 0; thisNote < size; thisNote++) {

      // to calculate the note duration, take one second
```

```
    // divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000 / tempo[thisNote];

    buzz(melodyPin, melody[thisNote], noteDuration);

    // to distinguish the notes, set a minimum time between them.
    // the note's duration + 30% seems to work well:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);

    // stop the tone playing:
    buzz(melodyPin, 0, noteDuration);


  }
}

 //Buzz function used by song function
void buzz(int targetPin, long frequency, long length) {
  digitalWrite(LED_BUILTIN, HIGH);
  long delayValue = 1000000 / frequency / 2; // calculate the delay value between transitions
  //// 1 second's worth of microseconds, divided by the frequency, then split in half since
  //// there are two phases to each cycle
  long numCycles = frequency * length / 1000; // calculate the number of cycles for proper timing
  //// multiply frequency, which is really cycles per second, by the number of seconds to
  //// get the total number of cycles to produce
  for (long i = 0; i < numCycles; i++) { // for the calculated length of time...
    digitalWrite(targetPin, HIGH); // write the buzzer pin high to push out the diaphram
    delayMicroseconds(delayValue); // wait for the calculated delay value
    digitalWrite(targetPin, LOW); // write the buzzer pin low to pull back the diaphram
    delayMicroseconds(delayValue); // wait again or the calculated delay value
  }
  digitalWrite(LED_BUILTIN, LOW);
 }
```

F.5 – Button control code

```
void setup()   /****** SETUP: RUNS ONCE ******/
{
  Serial.begin(115200);
  pinMode(13,INPUT);
}
//--(end setup )---
int last=0; //makes sure one push, no matter how long activated the flipper only once
// last = 0 means the last state is not pressed
//last = 1 means that the last state is pressed

void loop()   /****** LOOP: RUNS CONTINUOUSLY ******/
{
  int ButtonValue = digitalRead(13);
  if (ButtonValue && !last) Serial.println(ButtonValue); //only activate if last = 0 so the last
                                                          //check found it not pressed
  last = ButtonValue;

  delay(80);
}
```

F.6 – Word detection Control code

Ready to use code packages of the word detection software utilised can be found online in the Snowboy detection platform website [38]. The following code was uploaded on Arduino nano: The microprocessor outputs a digital HIGH at the USB connected to the CPU, once the Raspberry Pi 0 processor detects a voice input. (see Figure 5)

```
void setup()   /****** SETUP: RUNS ONCE ******/
{
  Serial.begin(115200);
  pinMode(4,INPUT);
}
//--(end setup )---
int last=0;

void loop()   /****** LOOP: RUNS CONSTANTLY ******/
{
  int VR_Value = digitalRead(4);
  if (VR_Value && !last) Serial.println(VR_Value);
  last = VR_Value;

  delay(80);
}
```

F.7 – Eyebrow control code

```
/*MMG*/
/*Integrated version*/

//Reference:
//LSM9DS1_Basic_I2C.ino
//SFE_LSM9DS1 Library Simple Example Code - I2C Interface
//Jim Lindblom @ SparkFun Electronics
//Original Creation Date: April 30, 2015
//https://github.com/sparkfun/LSM9DS1_Breakout

// The SFE_LSM9DS1 library requires both Wire and SPI be
// included BEFORE including the 9DS1 library.
#include <Wire.h>
#include <SPI.h>
#include <SparkFunLSM9DS1.h>

//IMU initialising
LSM9DS1 imu;

// SDO_XM and SDO_G are both pulled high, so our addresses are:
#define LSM9DS1_M  0x1E // Would be 0x1C if SDO_M is LOW
#define LSM9DS1_AG  0x6B // Would be 0x6A if SDO_AG is LOW

//microphone variables (naming pins used)
const int microphoneInPin = A0;  // Analog input pin that the MMG is connected to
const int mic_threshold = 200; //setting the thresholds
const int gyro_threshold = 50;

int microphone_value = 0;        // value read from the microphone of the MMG device

void setup() {
  // initialize serial communications at 115200 bps:
```

```
  Serial.begin(115200);
  // Before initializing the IMU, there are a few settings
  // we may need to adjust. Use the settings struct to set
  // the device's communication mode and addresses:
  imu.settings.device.commInterface = IMU_MODE_I2C;
  imu.settings.device.mAddress = LSM9DS1_M;
  imu.settings.device.agAddress = LSM9DS1_AG;
  // The above lines will only take effect AFTER calling
  // imu.begin(), which verifies communication with the IMU
  // and turns it on.
  if (!imu.begin())
  {
    Serial.println("Failed to communicate with LSM9DS1.");
    Serial.println("Double-check wiring.");
    Serial.println("Default settings in this sketch will " \
                   "work for an out of the box LSM9DS1 " \
                   "Breakout, but may need to be modified " \
                   "if the board jumpers are.");
    while (1);
  }
}

int last = 0; //last state considered (static (0) or moving (1))

void loop() {
  // read the microphone value:
  microphone_value = analogRead(microphoneInPin);

  // Update the gyroscope value whenever new data is available
  if ( imu.gyroAvailable() )
  {
    imu.readGyro();
  }

  if (microphone_value > mic_threshold)
  {
    if(Gyro() < gyro_threshold) //eyebrow movement when gyro is below 50
    {
      if(!last) {Serial.println(1);last = 1;} //avoiding multiple activation for the same movement
    }
    else last = 0;
  }
  else last = 0;
  delay(250);      // wait 250ms before checking again

}

float Gyro()
{
  return(sqrt(pow(imu.calcGyro(imu.gx),2)+pow(imu.calcGyro(imu.gy),2)+pow(imu.calcGyro(imu.gz),2)));
}
```

Appendix G – Bill of Materials

Item	Description	Part Number	Supplier	No off	Item Price	Total Price	Web link
1	TeckNet Aluminum USB Sound Card	B010N8UP6C	Amazon	1	£5.99	£5.99	https://www.amazon.co.uk/gp/product/B010N8UP6C
2	Chord FNM-35 Discreet Headset Microphone	-	Terralec Ltd	1	£13.60	£13.60	http://www.terralec.co.uk/radio_mics/chord_fnm35_discreet_headset_microphone/33833_p.html
3	Linear Actuator Solenoid Electromagnet DC 24V 350mA	B00EK6N2W8	Amazon	1	£8.52	£8.52	https://www.amazon.co.uk/gp/product/B00EK6N2W8
4	Flipper Bats	515-5133-01	Pinball Mania	8	£4.50	£40.40	https://pinparts.co.uk/products/515-5133-02-data-east-yellow-flipper-bat-with-de-logo
5	Flipper Coil FL-11630	FL-11630	Pinball Mania	2	£12.00	£28.50	https://pinparts.co.uk/products/flipper-coil-fl-11630-red
6	Flipper Coil FL-11630	FL-11630	1stoppinball	4	£15.30	£68.10	https://www.onestoppinball.co.uk/collections/coils/products/coil-fl-11630-red
7	Arduino Nano Pack of 3	B072BMYZ18	Amazon	1	£8.99	£12.98	https://www.amazon.co.uk/gp/product/B072BMYZ18
8	9DoF Sensor Stick (IMU's)	B01LBLPN44	Amazon	2	£16.74	£33.48	https://www.amazon.co.uk/gp/product/B01LBLPN44
9	Chrome Steel Ball Bearings	-	Simply Bearings Ltd	1	£7.18	£7.18	https://simplybearings.co.uk/shop/p35801/25mm-Diameter-Grade-20-Hardened-52100-Chrome-Steel-Ball-Bearings/product_info.html
10	25mm Diameter Solid Delrin Polyoxymethylene Ball	-	Simply Bearings Ltd	1	£5.44	£6.53	https://simplybearings.co.uk/shop/p61059/25mm-Diameter-Solid-Delrin-Polyoxymethylene-(POM)-/-Celcon-Plastic-Balls/product_info.html
11	ERW Square, Steel, 10x10mm, 1mm, 3.05m Tube and Pipe	-	Steeltubedirect.com	2	£6.64	£31.28	http://www.steeltubedirect.co.uk
12	MDF moisture resistant wood 6mm (2440 x 122)	-	Hitchcock and King	1	£20.22	£20.22	https://www.hitchcockandking.co.uk/
13	Mayor Brand Synchronous Toothed Belt	238.4MXL-012-298	Simple Bearings Ltd	6	£7.07	£57.72	https://simplybearings.co.uk/shop/p670053/2384MXL012-Major-Brand-Synchronous-Toothed-Belt-3.18mm-Wide,-2-mm-Pitch,-298-Teeth/product_info.html

Figure 39 – Bill of materials pt.1

Item	Description	Part Number	Supplier	No off	Item Price	Total Price	Web link
14	Silver Steel Ground Shaft	271277844956	Ebay	1	£2.99	£2.99	https://www.ebay.co.uk/itm/271277844956
15	Alumium tubes 10mm(H), 10mm(W), 1m(L)	-	B&Q Acton	5	£5.25	£26.25	https://www.diy.com/departments/aluminium-square-tube-h-10mm-w-10mm-l-1m/254111_BQ.prd
16	USB to microUSB OTG Converter Shim	-	The Pi Hut	1	£2.00	£4.50	https://thepihut.com/products/usb-to-microusb-otg-converter-shim
17	Raspbian Preinstalled Micro SD Card	-	The Pi Hut	1	£9.00	£9.00	https://thepihut.com/products/raspbian-preinstalled-sd-card
18	40-pin 2x20 HAT Dual Male Headers	-	The Pi Hut	1	£0.50	£0.50	https://thepihut.com/products/40-pin-2x20-hat-dual-male-headers
19	Adafruit Mini HDMI Plug to Standard HDMI Jack Adapter	-	The Pi Hut	1	£2.40	£2.40	https://thepihut.com/products/adafruit-mini-hdmi-plug-to-standard-hdmi-jack-adapter
20	Miniature Ball Bearing	619-0014	RS	4	£1.60	£6.40	https://uk.rs-online.com/web/p/ball-bearings/6190014/
21	DC Geared Motor	752-1999	RS	4	£7.04	£28.16	https://uk.rs-online.com/web/p/dc-geared-motors/7521999/
22	RFP30N06LE 30A 60V 0.047Ohm Logic Level N-Channel Power Mosfet (Pack of 10)	B071VXRQYR	Amazon	1	£13.99	£13.99	https://www.amazon.co.uk/Cylewet-RFP30N06LE-0-047Ohm-N-Channel-Arduino/dp/B071VXRQYR
23	NOOBS Preinstalled Raspberry Pi Micro SD Card	B01D4TW25Y	Amazon	2	£8.96	£19.22	https://www.amazon.co.uk/gp/product/B01D4TW25Y
24	Raspberry Pi Micro SD Card	-	Fotofast	1	£10.00	£10.00	Fotofast Store – South Kensington
25	50V to 5V 3A DC Step Down Voltage Regulator	B1700527EU	Amazon	3	£12.56	£37.68	https://www.amazon.co.uk/Yeeco-Waterproof-Converter-Transformer-Regulator/dp/B06XWRH3MS
26	USB 2.0 Type A Male - Mini B Male	B00DY2NAYE	Amazon	4	£1.99	£7.96	https://www.amazon.co.uk/gp/product/B00DY2NAYE/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1
27	StarTech.com 3ft USB Extension Cable A to A - M/F	B003YKX6VI	Amazon	4	£1.52	£6.08	https://www.amazon.co.uk/gp/product/B003YKX6VI/ref=od_aui_detailpages00?ie=UTF8&psc=1

Figure 40 – Bill of materials pt.2

Item	Description	Part Number	Supplier	No off	Item Price	Total Price	Web link
28	Raspberry Pi 3 Model B	896-8660	RS	1	£29.99	£29.99	https://uk.rs-online.com/web/p/processor-microcontroller-development-kits/8968660/
29	Brackets	-	Margaret Mill Hardware Store	5	£2.00	£10.00	http://mill-margaret.business.site
30	Washers	-	Margaret Mill Hardware Store	8	£1.75	£14.00	http://mill-margaret.business.site
31	Nuts & Bolts	-	Margaret Mill Hardware Store	7	£2.00	£14.00	http://mill-margaret.business.site
32	JoyNano Power Supply 48V 7.5A 360W AC-DC Converter	B075ZT5TNF	Amazon	1	£29.99	£29.99	https://www.amazon.co.uk/JoyNano-Switching-Transformer-Industrial-Automation/dp/B075ZT5TNF
33	Power Supply Switch Mode 48V 6.7A 321W	777-2866	RS	1	£68.57	£68.57	https://uk.rs-online.com/web/p/embedded-switch-mode-power-supplies-smps/7772866
34	Arduino Nano with Board and USB cable	B07143JN73	Amazon	3	£5.99	£17.97	https://www.amazon.co.uk/Arduino-Elegoo-board-ATmega328P-compatible/dp/B07143JN73
35	Character LCD Display Module HD44780	B01M4R0NSD	Amazon	3	£5.29	£15.87	https://www.amazon.co.uk/1602-16x2-Character-LCD-Display-HD44780-Blue-Raspberry/dp/B01M4R0NSD
32	3D printed parts (wheels, slopes, pulleys, coupling parts, corners)	-	ICRS (Robotics)	-	-	£41.50	http://icrs.io/facilities
Total Price						£751.52	

Figure 41 – Bill of materials pt.3

Appendix H – Acknowledgements

The authors would like to express their gratitude to the following people and organisations for supporting this project:

- Dr Ian Radcliffe for his supervision throughout all stages of the project and giving the team a large degree of independence in making their design and product decisions.
- Ms Jane Hales from John Chilton School for kindly agreeing to meet the team and respond to the requests regarding users' needs as well as her advice and feedback throughout the design process.
- Mr Jonathan Wijnen from John Chilton School for providing advice on assistive technologies and supplying the push button.
- The students of John Chilton School for being excited about the project and willing to provide as much feedback to the team as possible.
- Dr Ravi Vaidyanathan for kindly providing the MMG sensor developed by him and advice on sensor's implementation.
- Mr Paschal Egan for his support with the electrical design and supervision in electronics lab.
- Mr Niraj Kanabar for support with the electrical design.
- Mr Daniel Nardini and Mr Russell J Stracey for their help with manufacturing and advice regarding the manufacturing process.
- Dr Angela Kedgley for providing advice and feedback on the poster.
- Mechanics lab technicians for providing the facilities and advice for metalwork.
- Imperial College Robotics Society, and namely Ben Withers and Nick Hafner, for providing advice, 3D printing and laser-cutting facilities, and Aksat Shah for helping with debugging the circuitry.
- Imperial College London Department of Bioengineering for funding the project.

Appendix I – Nomenclature

MMG = mechanomyogram

IMU = Inertial Measurement Unit

EMG = electromyogram

EEG = electroencephalogram

CAD = computer-aided design

Rpm = Revolutions per minute

IR = Infra-red

CPU = Central Processing Unit

WD = Word Detection

RPi = RaspberryPi

LCD = Liquid Crystal Display

LED = Light Emitting Diode

TPU = Thermoplastic Polyurethane

MOSFET = Metal Oxide Semiconductor Field Effect Transistor

EOS switch = End of Stroke switch

MDF = Medium-density fibreboard