# Design Document

ENGG2K/3K
Comms - Marble Machine

MACQUARIE
University
SYDNEY · AUSTRALIA

# Table of Contents

**Table of Contents**....1
**Preface**....3
**Document History**....3
**Division of Labour and Workload Acknowledgement**.... 4
**0. Preface**....5
**1. Introduction**....5
**2. Problem Definition**....5
2.1 Subsystems.... 5
LED Lighting.... 6
Elevator System....6
Wiring System....6
Power Supply....7
Track Construction....7
Marble Path....7
Box Construction....7
Notes....7
2.2 Assumptions.... 7
**3. Deliverables**....9
**4. Requirements**....10
4.1 Interface Requirements & Sign-Offs.... 13
**5. Constraints**....14
**6. Inclusions and Exclusions**....16
6.1 Inclusions....16
6.2 Exclusions....17
**7. Conceptual Design**.... 18
7.1 Stages....18
7.2 Design Approaches.... 18
**8. Detailed Design**....19
8.1 Arduino Code Design:....19
8.2 LEDs:.... 20
8.3 Sensors:....20
8.4 Motors:....21
8.5 Event Queue....21
**9. Design traceability**....21
**10. Test cases**....22
**References**.... 24

**Appendix** ..... **24**
Terms, Definitions, and Abbreviated Terms ..... 24
Physical Diagram ..... 26
Electrical diagram ..... 27
Figure from requirements doc ..... 28

# Preface

# Document History

| Version | Date | Comments |
|---|---|---|
| V-01 | 23/08/23 | Create document, format and add headings. Division of labour assigned, sub headings and descriptions added |
| V-02 | 31/08/23 | Sub components added to Problem Definition - LED lights, elevator system |
| V-03 | 03/09/23 | Assumptions and requirements added |
| V-04 | 06/09/23 | Arduino Code Design added to Detailed Design, Stages and Design Approaches added |
| V-05 | 09/09/23 | Conceptual Design and Deliverables added |
| V-06 | 10/09/23 | Conceptual Design broken down into Stages and Design Approaches |
| V-07 | 11/09/23 | Requirements divided into Functional, Performance and Interface Requirements, addition of Constraints and Inclusions & Exclusions |
| V-08 | 12/09/23 | LEDs, Sensors and Motors added to Detailed Design |
| V-09 | 14/09/23 | Test Cases and Design Traceability, Sign offs complete, final review |

# Division of Labour and Workload Acknowledgement

| **Member name** | **SID** | **Cohort** | **Sections Responsible** | **Overall Workload %** |
|---|---|---|---|---|
| Flynn  Lauridsen | 46375864 | ENGG3000 | Test Cases, Traceability, Design | |
| Safwan Abrar | 47275332 | ENGG2000 | Doc Structure | |
| Debashish Kumar | 47080159 | ENGG2000 | Design, Conceptual design | |
| Gavin Yap | 47113103 | ENGG2000 | Formatting | |
| Jenifer Surenthirakumar | 46987150 | ENGG2000 | References, Appendix, Introduction | |

# 0. Preface

This document outlines the collaborative contribution made to bring the Massive Marvellous Marble Machine to life, which is an engineered structure that guides a marble through a cube adhering to the requirements set by the client. This document provides a blueprint for the system, designed through the collaboration of Comms 1 as well as its partner groups. It contains the creative and innovative ideas resulted through the collective endeavour of the engineers within this project. The team's vision, approach and rationale behind the design choices are explored in this document.

# 1. Introduction

The Massive Marvellous Marble Machine is designed to operate a series of components that guide marbles made of various materials through a delineated path while simultaneously satisfying the requirements outlined by the client.

This document contains the collaborative design put forth by Comms 1 along with partner groups; Motions and Structures, for the Massive Marvellous Marble Machine. The documented design aims to fulfil the requirements set forth by the client, the School of Engineering and provides a breakdown of the mentioned requirements and how they are fulfilled. It also contains a detailed description of the several sub-components of the system's design.

This document is expected to assist in the development of the Marble Machine and act as an overview of what and how it needs to be built. It also provides a detailed description of the high-level solution to be presented to the client and stakeholders.

# 2. Problem Definition

## 2.1 Subsystems

Each team in this project is responsible for a set of subsystems, with some subsystems having overlap in team responsibility. These subsystems are illustrated in Figure 1 below. This subsection will explicitly define and describe each subsystem relevant to the Comms team and outline known responsibilities and tasks for the Comms team in regard to each subsystem. This subsection will also outline other subsystems by their functions in the overarching system.

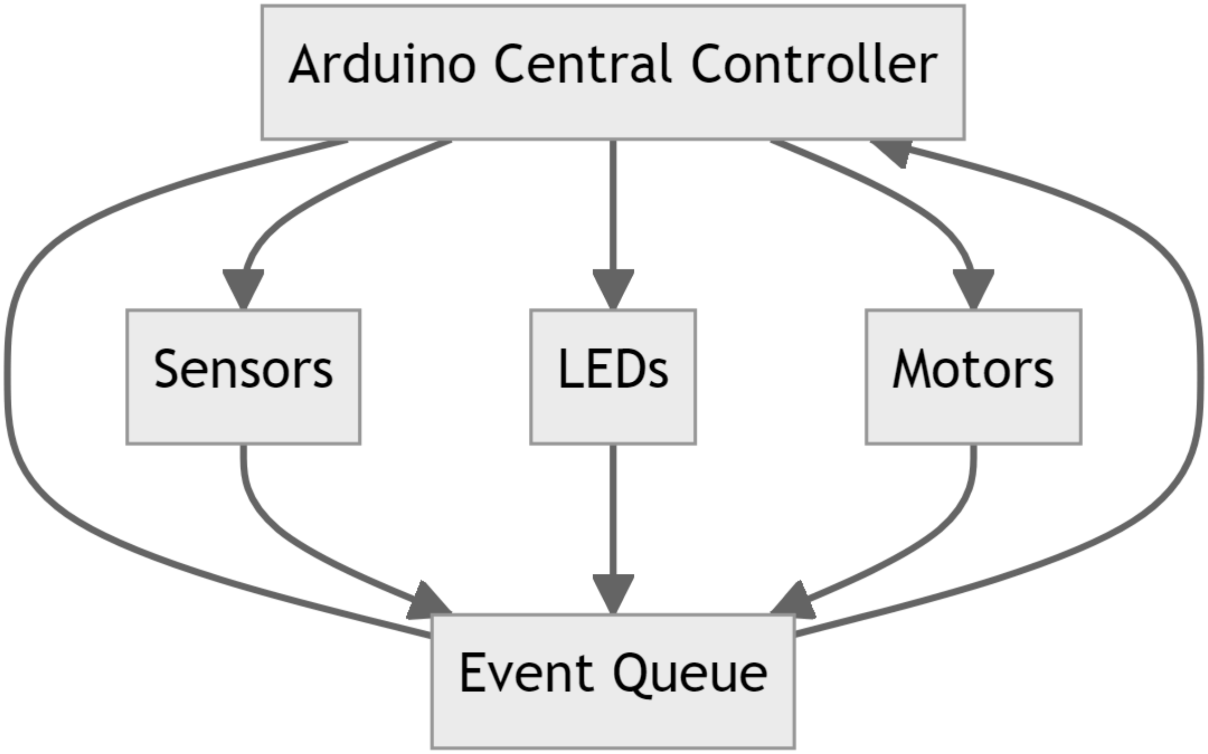


*Figure 1*

The code base as a subsystem pertains to the functions of all electronic components within the box. In the final product, the code base will be stored on the Arduino microcontroller and will define almost all variables within the electronic components of the box, including the speed of the elevator system and how the LED strips will react to a given input.

**LED Lighting**

The reactive lighting system will be composed of an LED strip and multiple sensors, providing an aesthetic visual aspect to the box. Sensors placed along the track will receive input from marbles, triggering a response from the LED strip. The LED system will provide an indication of a function being performed in the box as per requirement RC3, and therefore will take in inputs from other subsystems and operate concurrently.

An Arduino microcontroller will be used to both control and power this subsystem. The Arduino will allow for an easier implementation of more dynamic and complex responses to sensor inputs. These responses could be to indicate the marble's speed, location, or other behaviours of the box and environmental factors.

The LED strip will contain an array of individually addressable LEDs. The LED strip will be controlled by the Arduino microcontroller and can be any length; ignoring the constraint of power available to be supplied by the Arduino.

Infrared sensors used will be used to provide real-time inputs of the proximity of the marble to a given point along the marble path.

**Elevator System**

The elevator system is pivotal to driving the marbles against gravity. The system will primarily consist of continuous rotation servos controlled by the Arduino and a variable resistor. The Arduino will send signals to the servo motor to adjust its position, moving a 'bin' that transports the marbles. The variable resistor is necessary in regulating the speed of the servo to ensure that future teams will have a dynamic way to control the operation. The elevator system will pause for 1 second intervals to accommodate requirement LC2.

**Wiring System**

The wiring system will serve as the connection between the Arduino and all other electronic components within the box. This subsystem will require planning to ensure that none of the wires used interfere with other aspects of the box and present no significant risks in regard to events like short circuits or crossing wires.

Both Comms and Motions will be responsible for this subsystem as it will interface with the Elevator and the LED Lighting systems. Thus Motions holds responsibility for the former and Comms for the latter. It is essential that the Wiring system is agreed upon by both teams.

**Power Supply**

The power supply will serve as an additional source of power to ensure that the Arduino microcontroller only needs to supply power within its capacity.

**Track Construction**

The Track Construction subsystem refers to the platforms the marble will be moving along, also referring to any transitions the marble will make onto other components of the box or leaving the box. The track will be developed to adhere to the Level Requirement LC1 such that it rotates around an imaginary line.

**Marble Path**

The Marble Path subsystem refers to the exact path the marble will take throughout the box. The important aspect of this subsystem is in ensuring that the marble is moving at the correct speed and acceleration such that it can move throughout the box successfully and predictably.

**Box Construction**

The Box Construction subsystem refers to the components within the box maintaining the structural integrity of the box and the components inside it. This ensures that the box can withstand reasonable forces when being moved and that the box will remain durable to longevity testing with small, repeated forces from multiple marbles.

Notes

Additionally, it is imperative that the team strictly adheres to all aforementioned level conditions and the cube's requirements. The successful integration of both subsystems will lead to a successful and fully operational final product.

## 2.2 Assumptions

*Table 2 - Assumptions about the project*

| **ID** | **Notes** |
|---|---|
| A-1.1 | It is assumed that the engineering solution will be undertaken within a domestic environment, with access to standard household electrical power supplies and basic hand tools. |
| A-1.2 | It is assumed that it is possible to procure all specified components including the LEDs, IR sensors, Arduino microcontrollers, servos and any other requisite materials for this project within the agreed budget. |
| A-1.3 | It is assumed that the client's requirements will remain stable as discussed prior to undertaking this project and there will be no major addition to the requirements. |
| A-1.4 | It is assumed that the partnered groups are able to work collaboratively and provide access to the required hardware in order to implement and test the software. |
| A-1.5 | It is assumed that the scope of the project falls within legal and safety standards |

# 3.Deliverables

*Table 3 - List of deliverables*

| ID | Name | Description | Due |
|---|---|---|---|
| D-1.1 | Scoping Document | Define the specific goals and requirements of the Marble Machine project. | 20/08/23 |
| D-1.2 | Design Document | Provides a solution fulfilling the functional and non-functional requirements of the Marble Machine project. | 09/09/23 |
| D-1.3 | Testing Document | Outlines the testing procedures and fulfilment of TPM'S. | 21/10/23 |
| D-1.4 | Statement of work | Declaration of the work completed by the members of the group. | 21/10/23 |
| D-1.5 | MVP 1 | First product prototype. | |
| D-2.1 | MVP Improvements | Improvements to the prototype. | |
| D-2.2 | Final product | Product ready for demonstration. | 28/10/23 |
| D-2.3 | Final documentation | Product documentation is complete to the details of the project and how it was implemented. | |
| D-2.4 | Product demonstration | Demonstrates a working product meeting the scope and requirements as outlined in this document. | 28/10/23 |

# 4.Requirements

*Table 5 - Requirements as stated by client*

| | **FUNCTIONAL REQUIREMENTS** | |
|---|---|---|
| **ID** | **Requirements** | **Description** |
| RC1 | Each cube MUST have the external dimensions of 250mm by 250mm by 250mm. | The cube's dimension **MUST NOT** exceed the given specifications. |
| RC2 | Each cube MUST mount to the sculpture using only the four mounting positions. | The cube **MUST** be mounted in the way provided by the requirements. |
| RC3 | A cube MUST provide some visual indication using LEDs or other lighting system approved by the customer as to the function being performed. | The cube **MUST** have some sort of LED light in operation when a function is being performed. |
| RC4 | A cube MUST access external services, such as power or telecommunications. | The cube **MUST** access the external services provided. |
| RC5 | Each cube location on the backing board SHALL be in the middle of a square 265mm horizontally by 265mm vertically. This requirement implies that there is a 15mm distance between cubes. | The cube **MUST** be placed on the location clarified via the requirements. |
| RC6 | Each cube MUST be designed to accept marbles that are a solid steel sphere with a diameter of 16mm. The cube SHOULD also accept marbles of the same diameter that are made of lighter materials. It is RECOMMENDED that lighter marbles still pass through the cube. | The cube **MUST** accept different types of marbles with a diameter of 16mm. |
| RC7 | Each cube MUST accept marbles that are spaced no closer than 1 second apart through the opening port. | The Cube is given a specifications which **MUST** be followed as it is a part of a fixture which has common features across each cube. This feature |

| | | |
|---|---|---|
| | | includes a standardised time gap of 1 second between each marble. |
| RC8 | Cubes MUST be manufactured from 3mm MDF for the back and side pieces, with the additional requirement that the front surface MUST be transparent. | The cube **MUST** adhere to the manufacturing requirements and additional requirements of the project. |
| RC9 | The front surface MUST NOT be structural, and MUST be easily removable for maintenance purposes for the cube. | The front surface of the marble machine **MUST** be easily removable for future maintenance/improvements. |
| RC10 | The only forces provided to a cube are gravity and electrical power. A cube MUST NOT use any liquid in its operation, and MUST NOT use compressed air or other gases. | The cube **MUST NOT** use any other forces than the ones provided such as gravity and electrical power. |
| RC11 | Each cube MUST have a bill of materials, and each bill of materials MUST total to less than AUD$100.00. | The cube's BOM **MUST NOT** exceed. |
| | **PERFORMANCE REQUIREMENTS** | |
| RC20 | Each cube MUST accept a marble on a through an input opening on a surface other than the top surface of the cube. This location must be agreed to by the customer. | The cube **MUST** accept a marble from anywhere but the top surface dependent upon the client. |
| RC21 | Each cube MUST expel all marbles out through an exit opening on a different surface to that where the marble entered. | The cube **MUST** expel marbles from a different surface to the entry point. |
| RC22 | Cubes SHALL NOT deliberately expel marbles at a speed significantly higher than that attributable to gravity alone. | The cube **MUST NOT** deliberately exert marble faster than the speed of gravity. |
| RC23 | Each cube MUST control the marble in such a fashion that the bottom of the exit opening is at least 100mm higher than the top of the input opening. | The cube **MUST** have an exit located higher than the starting point. |

| | | |
|---|---|---|
| RC24 | Each cube MUST control the marble in such a fashion that satisfies ONE of the LEVEL conditions LC1 to LC3. | The cube **MUST** adhere to one of the level conditions provided. |
| | **LEVEL REQUIREMENTS** | |
| LC1 | The marble makes a full rotation around an imaginary line that MUST NOT intersect the marble, drawn through the cube. In other words the marble's direction must change in at least two dimensions by the size of the marble. | |
| LC2 | The marble is held stationary in all physical dimensions for a minimum of 0.5 seconds in time. | |

## 4.1 Interface Requirements & Sign-Offs

*Table 6 - Table of interface requirements*

| ID | Description | TPM | Linked Requirement | Subsystem Involved | Sign-Offs |
|---|---|---|---|---|---|
| IR-1 | An Arduino Uno will be responsible for the control of all electrical components | Electrical components prove reliability during operation | RC4 | Wiring System/LED Lighting/Code Base | Flynn Lauridsen<br><br>Rhys Jones |
| IR-2 | The LED Lighting subsystem will use LED strips as a mechanism for generating light. | The LED strips should not move from installation location and emit light when powered accordingly. | RC3 | LED Lighting | Flynn Lauridsen<br><br>Rhys Jones |
| IR-3 | The box will have a set of sensors which will be used as input to trigger the LED subsystem | The sensor should not move from installation location and provide correct input to the Arduino | RC3 | LED Lighting | Flynn Lauridsen<br><br>Rhys Jones<br><br>Sean Mitchell |
| IR-4 | The Arduino will be uploaded with code to define its functional operation | The Arduino will perform according to design specification with minimal deviation. | RC23<br>RC24 | Code Base | Flynn Lauridsen |

# 5.Constraints

## 5.1 Constraints

The marble machine project faces several constraints that must be considered during its design and implementation.

First, the mounting of the cubes and access to external services must adhere to specific four mounting positions and service ports. This ensures uniformity and compatibility.

Furthermore, the hardware and software limitations of the Arduino board pose constraints. The number of pins and processing power is limited, requiring careful consideration to manage the load and performance efficiently. Additionally, the sketch size and complexity must be within the memory capacity of the Arduino board.

The use of Arduino boards for prototyping is advantageous, but scalability in terms of cost, efficiency, and size may be limited. Similarly, constraints related to interfaces, data storage, I/O pins, weight, torque, signal control, positional accuracy, speed, lifecycle, noise, and physical size must be accounted for during the project's design.

*Table 7 - Table of constraints*

| ID | Constraint | Description |
| --- | --- | --- |
| C-1 | Connection | All cubes should be mounted to the sculpture using only four specified mounting positions and should access external services like power or telecommunication only through a specified service port. |
| C-2 | Hardware | The number of pins and processing power available on the Arduino board is limited; therefore, design considerations need to take into account maximum load and performance. |
| C-3 | Software | Arduino uses limited memory for storing and running codes. The sketch size and complexity must be within the limits imposed by the Arduino's memory capacity. |

| | | |
|---|---|---|
| C-4 | Timing | Uninterrupted operation of the Arduino for a long period may cause timing issues if the built-in millis() and micros() functions are used. |
| C-5 | Power Supply | The Arduino board has a specific voltage and current requirement that should not be exceeded, as it may cause the board to overheat or may render it useless. |
| C-6 | Scope | Arduino boards are great for prototyping and suitable for a vast range of applications, but if you are planning to build a product to be produced on a larger scale, you may face certain limitations with Arduino in terms of cost, efficiency, and size. |
| C-7 | Interfacing | Sensor and servo interfaces should not exceed the voltage specifications of the Arduino board to prevent damaging the board's digital I/O pins. |
| C-8 | Data Storage | Arduinos have minimal on-board data storage. If the project requires data logging or high memory usage, additional memory may be needed which can add complexity and cost to the project. |
| C-9 | I/O Pins | The number of Input/Output (I/O) pins are limited in Arduino. If your design requires a large number of I/O operations, you might be restricted by these quantity limitations. |
| C-10 | Weight | The weight that the servo has to move cannot exceed its lift capacity. |
| C-11 | Torque | Each servo motor can provide a certain maximum torque. Overloading a servo by demanding more torque than it can supply can lead to poor performance and potential failure. |
| C-12 | Signal control | Servo motors are controlled by PWM (Pulse Width Modulation) signals, which should maintain a certain frequency and duty cycle for proper operation. |
| C-13 | Positional Accuracy | The precision of a servo motor is limited and high precision control may require costly high-end components. |

| | | |
|---|---|---|
| C-14 | Speed | The rotational speed of a servo is finite and trying to drive a servo faster than its maximum speed could lead to poor performance and potential damage. |
| C-15 | Lifecycle | The continuous rotation may wear out the gears of a servo, limiting the overall life expectancy of the device. |
| C-16 | Noise | Servos can create noise during operation, which could be a factor in certain environments. |
| C-17 | Physical Size | The size of servos may be a constraining factor in applications where space is limited. |
| C-27 | Energy Efficiency | Servos consume power even when they're maintaining a stationary position which could lead to power efficiency concerns in power-sensitive systems. |

# 6.Inclusions and Exclusions

## 6.1 Inclusions

*Table 8 - Table of services and deliverables included in the project*

| ID | Description |
|---|---|
| IN-1 | Investigate the feasibility of the design and implementation |
| IN-2 | Investigating required products and materials |
| IN-3 | Investigate cost of products and alternative options |
| IN-4 | Ensure the cost of materials does not exceed $100 *(Refer to requirement RC11)* |
| IN-5 | Ensure the usage of durable and high quality materials |
| IN-6 | Present a breakdown of budget and finalise a list of materials |
| IN-7 | Present a finalised design that meets client requirements |
| IN-8 | Investigate additional potential design features at client's request |
| IN-9 | Consider power needs for electrical components |
| IN-10 | Collaborating with external teams to construct electrical components |
| IN-11 | Integrating software component with the physical system |
| IN-12 | Generate an LED arrangement |
| IN-13 | Ensure LED lights flash when the server is active to indicate the functions being performed in the cube (*refer to requirement RC3)* |
| IN-14 | Perform a complete and through test of the system |
| IN-15 | Perform module tests of separate software components |
| IN-16 | Provide debugging and troubleshooting for software issues within the system |
| IN-17 | Test compatibility and integration with the hardware |
| IN-18 | Ensure software components satisfy requirements |

## 6.2 Exclusions

*Table 9 - Table of services and deliverables not included in the project*

| ID | Description |
|---|---|
| EX-1 | Physical construction |
| EX-2 | Assembly of the cube, structural components and construction/arrangement of electrical components |
| EX-3 | Installation of track system and spiral track |
| EX-4 | Investigate the components and design of the spiral tracks |
| EX-5 | Investigate the structural integrity of the design |
| EX-6 | Perform calculations regarding the velocity and momentum of the marble in the system |
| EX-7 | Test beyond software integration |
| EX-8 | Test physical structure of the system or marble tracking |
| EX-10 | Implement user-interface within the physical system |
| EX-11 | Integrate UI for the user to interact with the software |
| EX-12 | Provide maintenance of the physical components of the system |
| EX-13 | Provide solutions for inoperative equipment/components |

# 7. Conceptual Design

This chapter describes the conceptual design of the marble machine, covering concepts and themes in the design as well as design and implementation approaches with reasoning for these choices.

## 7.1 Stages:

The conceptual design of this project is divided into 5 primary stages:

1. First Straight: This stage involves a straight track at the entry point of the box, where the marble rolls down. An LED strip will "trace" the marble's movement, matching its speed as it progresses down the track.

2. Roundabout/Second Straight: This stage features a circular track followed by another straight track. As the marble enters the circular track, a Darth Vader MDF cut-out will have backlighting and pulsing effects. The backlighting of Darth Vader begins when the marble enters the circle and ends after the marble exits.

3. Queue: The queue stage acts as a transitional point between the track sections and the mechanical sections of the box. The marble will wait at the queue stage, resting against the ferris wheel until the ferris wheel collects it in a bucket.

4. Ferris Wheel: In this stage, the marble is carried by a bucket on a spinning ferris wheel from the track to the conveyor belt. The ferris wheel will only start rotating if there is a marble waiting in the queue stage.

5. Conveyor Belt: The conveyor belt is the final stage of the box. It receives marbles from the ferris wheel stage and carries them upwards towards the exit. The conveyor belt movement alternates between 1-second intervals of motion and 1-second intervals of stopping. This ensures that the marble remains stationary for 1 second within the box.

## 7.2 Design Approaches:

The design approach for this project incorporates the use of the Arduino Uno 3 microcontroller and Arduino's superset of the object-oriented language C++. This choice allows us to develop a modular and testable design, aligning with common practices in the hobbyist and electronics community.

To ensure reliability and compatibility, we will utilise commonly available electronic components and standard libraries that interface with these components. By leveraging well-supported and widely-used components and libraries, the design will be more robust and align with established best practices.

These design choices will facilitate the development of a functional and efficient marble machine while embracing established conventions and availing the benefits of readily available resources within the Arduino ecosystem.

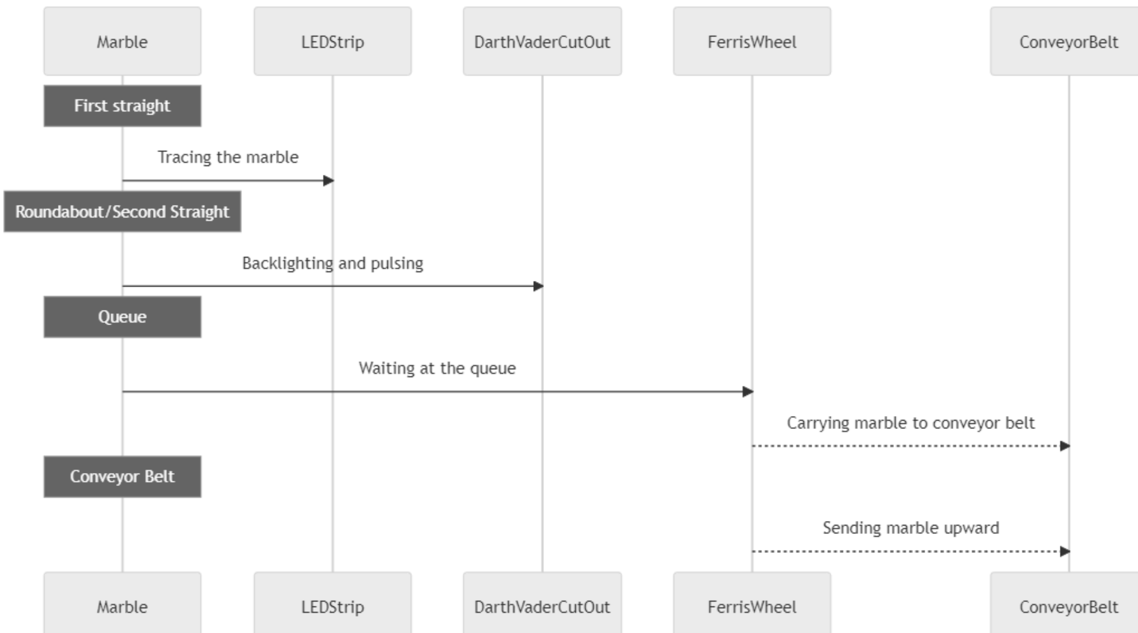


*Figure 2 - Typical event cue*

# 8.Detailed Design

This chapter describes the detailed design of the marble machine. It will cover components, code blocks, and their interfaces, with detailed explanations for these choices.

Below, a diagram of the approximate Arduino design can be seen, in figure 3. Due to limitations of TinkerCad, push buttons have been used to simulate the proximity sensors, which simulate the HIGH | LOW signal output of the sensor. Note: the sensors used in the physical construction of the box do not require the resistors depicted in this diagram.

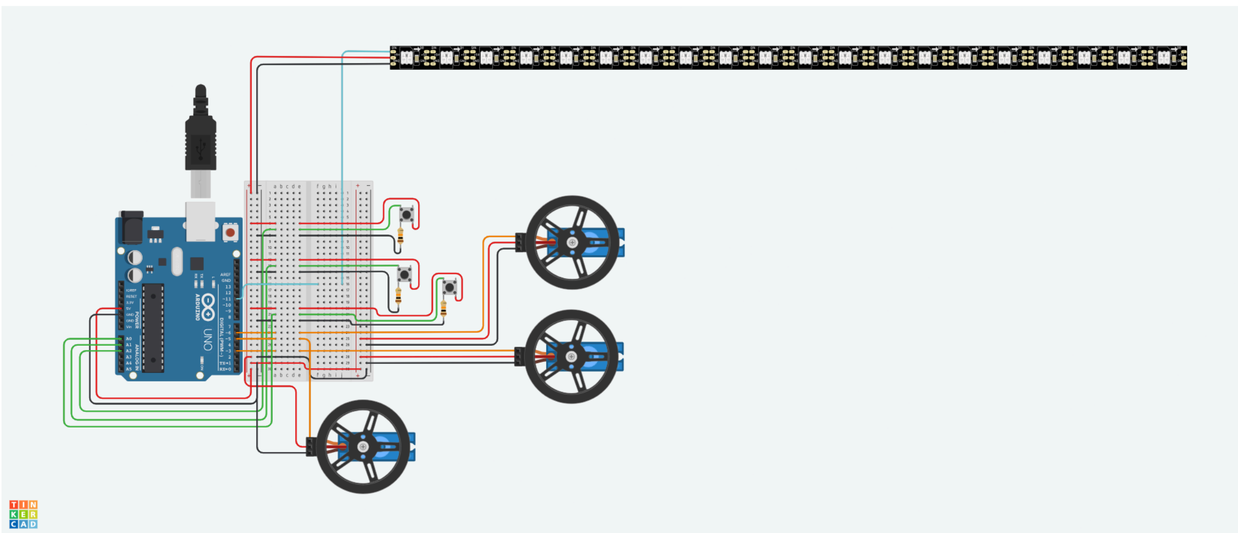

*Figure 3 - TinkerCad Diagram*

## 8.1 Arduino Code Design:

The Arduino code for the marble machine is developed to control various components, including LEDs and sensors. The code is structured using object-oriented programming principles to promote modularity and code reusability.

The code is divided into several sections:
- Libraries: The necessary libraries, such as Servo and Adafruit_NeoPixel, are included for controlling servos and LEDs.
- Pins Configuration: Pin numbers and constants are defined for each component, such as sensor pins, servo pins, and LED strip pin.
- Objects Instantiation: Objects are created for each component, such as button sensors, servos, and the LED strip.
- Class Definitions: Classes are defined for specific components, including ProximitySensor, SensorDrivenServo, ContinuousServo, and LEDStrip. These classes encapsulate specific

functionalities and provide methods for initialization and control of the respective components.
- Setup Function: The setup function is responsible for initialising the components by calling the initialise method of each object.
- Loop Function: The loop function continuously checks the sensor states and performs actions accordingly. It reads the sensor values, triggers animations on the LED strip, and moves the servos.
- Additional Functions: Additional functions can be added to perform specific tasks as per project requirements.

The code incorporates concepts such as conditional statements and loops to handle sensor inputs and control the behaviour of LED animations and servo movements. By utilising custom classes for components, the code follows a clear and organised structure, enabling easy modification and scalability.

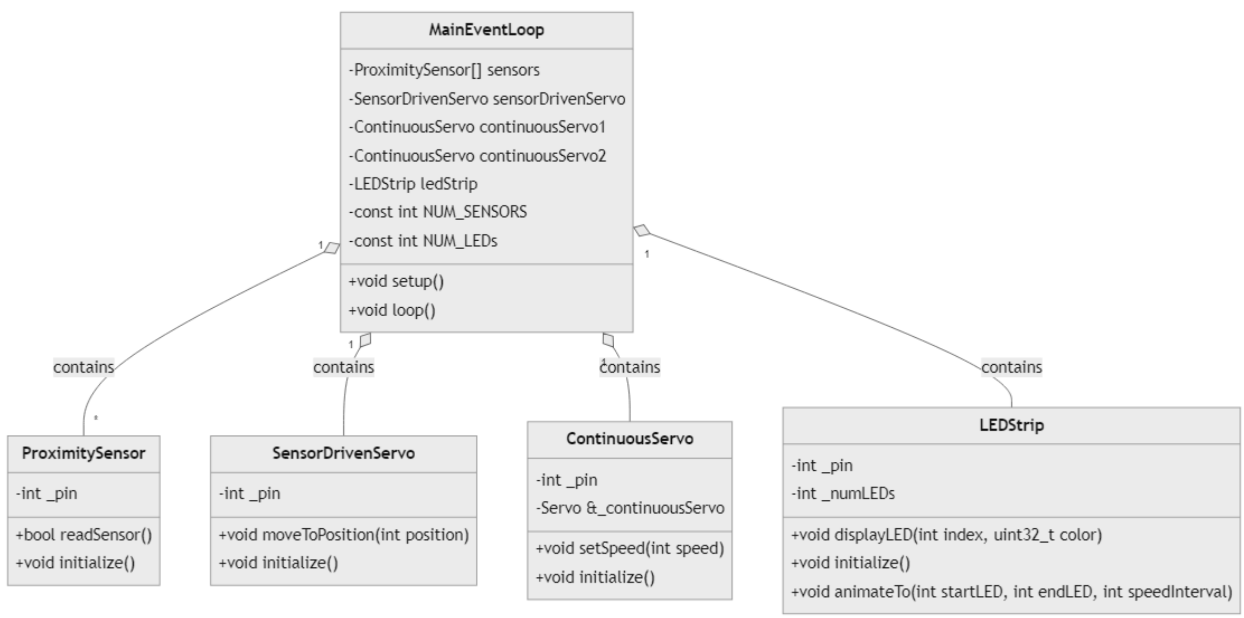


*Figure 4 - Abstract class diagram showing the hierarchy of the intended implementation*

## 8.2 LEDs:

The LEDs are controlled using the Adafruit_NeoPixel library. An instance of the Adafruit_NeoPixel class is created to control the LED strip connected to a specified pin. The LEDStrip class encapsulates functionalities related to LED control, including displaying colours, animating LED patterns, and initialising the LED strip. Additionally, the LEDStrip class has been expanded to include the animateTo method, which allows for the animation of LED patterns from a starting LED to an ending LED, with customizable animation speeds.

## 8.3 Sensors:

The sensors used in the marble machine are button sensors. To emulate the behaviour of proximity/IR sensors during prototyping, buttons are connected to Arduino analog pins with internal pull-up resistors enabled. The ButtonSensor class is implemented, which encapsulates the functionality to read the button status and initialise the pins with internal pull-up resistors. These button sensors serve as a proxy for the actual proximity/IR sensors. The loop function continuously checks the button states and triggers animations on the LED strip based on the button states. Each button's state is associated with a specific LED animation speed defined in the SENSOR_PINS_SPEED_INTERVALS array.

The marble machine employs a modular and flexible design approach, allowing for easy integration of additional sensors, actuators, or LED patterns. The code and components are designed to work together harmoniously, providing an interactive and visually appealing display.

## 8.4 Motors:

There will be 3 motors in total for the marble machine. 2 motors will be DC motors, used to actuate the conveyer belt and 1 motor will be a continuous servo, used to actuate the ferris wheel.
The motors will spin in 45° increments, with a 1 second pause between each increment. The purpose of this behaviour is to add visual appeal and meet client requirements.

## 8.5 Event Queue:

The Event Queue is a critical part of the design which aids in asynchronous operations of the Marble machine. It allows the system to handle multiple marbles at different stages of the run without waiting for one to complete the whole cycle before starting with the next.

The events, in this case, are the triggering of sensors as the marbles pass various stages in the machine. As a marble enters a particular stage, a corresponding event is created and added to the queue. These events are continuously checked in a loop and processed respectively.

The Event Queue is implemented as an abstract data type (typically as a First In First Out collection - FIFO), where the oldest event (the one that happened earliest) is processed first. As events are processed, they are removed from the front of the queue.

Each event triggers corresponding actions, such as starting/stopping the motors, activating LEDs, etc. This design allows the system to keep track of multiple marbles at different stages since each marble trigger is recognized as an independent event.

The Event Queue based approach provides a robust design to handle multiple asynchronous tasks in an organised fashion, increasing the efficiency, responsiveness, and scalability of the system. It greatly enhances the ability of the machine to handle multiple marbles at ~2 seconds interval, providing a smooth and consistent operation.

## 8.6 Wiring:

A circuit board will be used to interface the connection between the electrical components and the Arduino. This will allow for an organised and permanent wiring solution, without requiring any soldering directly to the Arduino board. Stranded wire will be used throughout the wiring, as this ensures more reliable connections at points of soldering.
Below is an electrical diagram of the design.

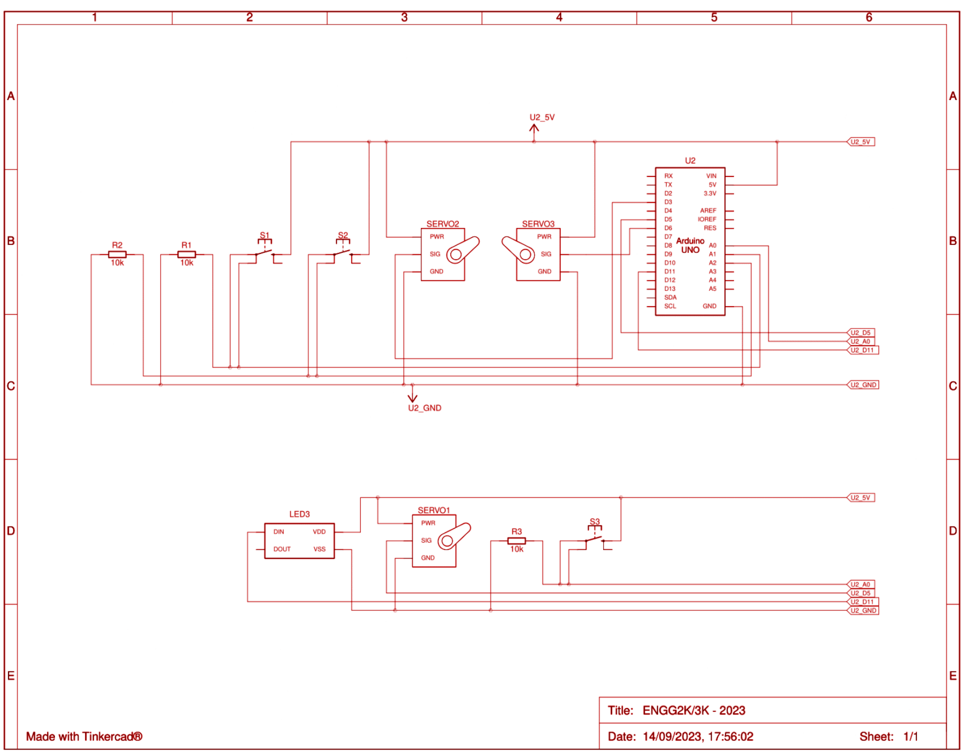


*Figure 5 - Electrical Diagram*

# 9.Design traceability

| Requirement ID | Description | Design Component | Linkage |
|---|---|---|---|
| RC-3 | A cube MUST provide some visual indication using LEDs or other lighting systems approved by the customer as to the function being performed. | LED strip, Arduino, sensor | The LEDs enable visual indication, while the Arduino and the sensor allow this indication to be reactive. |
| RC-4 | A cube MUST access external services, such as power or telecommunications, only through the service port shown in figure 6* | Arduino | The Arduino will receive its power through the provided service port. |
| RC-10 | The only forces provided to a cube are gravity and electrical power. A cube MUST NOT use any liquid in its operation, and MUST NOT use compressed air or other gases. | Motors, Arduino | The motors will enact a force on the marble, enabled by the electrical power of the Arduino. |

* Figure 6 is provided in the Requirements Document, see Fig. 1 in Appendix

# 10. Test cases

| ID | Description | Expected Results | Prerequisite Tests |
|---|---|---|---|
| T-1 | Power the Arduino with its main loop continuously printing “Hello World” | The arduino will print “Hello World” continuously in the STDOUT | N/A |
| T-2 | Power the motor | The motor spins | T-1 |
| T-3 | Power the LED strip, with static lighting as input | The entirety of the LED strip illuminates at full brightness | T-1 |
| T-4 | Power the sensor and move an object in and out of its effective range. | The sensor will provide a HIGH signal in the presence of an object and a LOW signal in the absence of an object | T-1 |
| T-5 | Set the LED strip to light up with a HIGH signal from the sensor | The LED strip will light up in the presence of an object | T-3, T-4 |
| T-6 | Set the motor to rotate 45°, pausing for a second after each increment | The motor will rotate within 1° of 45°, with the specified pause | T-2 |
| T-7 | Set each addressable LED in the strip to light up sequentially, at a certain speed | The LEDs will present with a “trace” effect, indicating a speed calculable by the distance between each LED over the rate in which they light up | T-3 |
| T-8 | Sed the entire LED strip to light up, | The LED strip will light up in the colour | T-3 |

| | | | |
|---|---|---|---|
| | provided with an RGB value to determine the colour. | corresponding with the RGB value provided | |
| T9 | Set the motor to spin as is configured in test T6, with a weight attached to its radius | The motor will rotate within 1° of 45°, with the specified pause | T6 |
| T10 | | | |

# References

**[1]** Core Electronics (n.d.). *Infrared IR Proximity Sensor for Arduino (10±5mm~80±20mm)*. [online] core-electronics.com.au. Available at: https://core-electronics.com.au/infrared-ir-proximity-sensor-for-arduino-105mm8020mm.html [Accessed 15 Aug. 2023].

**[2]** www.jaycar.com.au. (n.d.). *8mm White LED Solid Strip Light, 12V | Jaycar Electronics*. [online] Available at: https://www.jaycar.com.au/8mm-white-led-solid-strip-light-12v/p/ZD0461?pos=3&queryId=ef18917e5d9aa678768c6a06baabdcff&sort=relevance&searchText=led%20strip [Accessed 15 Aug. 2023]

# Appendix

## Terms, Definitions, and Abbreviated Terms

*Table 1 - Terms and abbreviations used in this document and their definitions*

| **Abbreviation** | **Definition** |
| --- | --- |
| Massive Marvellous Marble Machine | The product that is being developed. It is a system designed to guide a marble through a cube through complex subsystems. |
| Comms 1 | Communications Team 1 - The team this documentation is created by |
| LED | Light-emitting diode - the lighting system used in this project |
| Sub-system | Smaller components of systems that make up the Marble Machine System. |
| Arduino Microcontroller | Programmable circuit board controlling different electrical components including LED lights, sensors and servos motors. |
| Variable Resistor | A component used to control the resistance - used to control the speed in this machine |
| MDF | Medium Density Fibreboard |
| Modularity | Division of a system into smaller modules |
| Signal Control | The system regulating the signals being sent to control different components |
| Prototyping | Making a preliminary model of the system |
| Position Accuracy | Precision with which the servo motors are controlled |
| Energy Efficiency | How well the energy is used in the system to operate the system |

*Table 1 - Bill of Materials*

| Macquarie University Bill of Materials | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Project: | Marble Machine | | | | | | | | | | | | |
| Team: | T4S2, T4C1, T4M1 | | | | | | | | | | | | |
| Group: | T4S2, T4C1, T4M1 | | | | | | | | | | | | |
| Order Date: | 8-Sep-23 | | | | | | | | | | | | |
| Notes: | | | | | | | | | | | | | |
| Group Name (ilearn) | Submitter (Name) | Total Quantity Required | Description | Purchase Quantity (Supplier Units) | Price (AUD) (excluding GST) | Price (AUD) (including GST) | Lead time (Days) | Delivery Date | Shipping Price | Extended Price: | Alternative Acceptable (Yes/No): | Supplier URL |
| T4S2 | Luke | 1 | 330mm sanding Belt | 3 | $9.99 | $10.99 | 7 | | | $10.99 | yes | Link |
| T4M1 | Rhys | 1 | Arduino servo motor | 1 | $6.32 | $6.95 | 6 | 25/9/22 | $7.16 | $14.11 | No | Link |
| T4M1 | Rhys | 1 | 70RPM 12VDC Reversible Gearhead Motor | 1 | $15.41 | $16.95 | 7 | 25/9/23 | 8 | $24.95 | No | Link |
| T4M1 | Rhys | 3 | Arduinos | 0 | $0.00 | $0.00 | 0 | 23/8/23 | 0 | 0 | yes | |
| T4C1 | Flynn | 3 | Infrared | 3 | $6.32 | $6.95 | 7 | 25/9/23 | | $20.85 | No | Link[1] |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | sensor | | | | | | | | | |
| T4C1 | Flynn | 1 | LED Strip 1m | 1 | $7.27 | $8.00 | 7 | 25/9/23 | | $8.00 | yes | Link[2] |
| T4S2 | Luke | 1 | Twin Wire roll | 1 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |
| | Uni | 1 | Box - mdf | 1 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |
| | Uni | 1 | Front screen - polycarbonate | 1 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |
| T4S2 | Luke | 1 | Self vulcanising rubber tape | 1 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |
| T4S2 | Luke | 1 | Glue/fasteners | 1 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |
| T4S2 | Luke | 20 | Steel Rods | 4 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |
| T4S2 | Luke | 2 | Plastic sheet/plate for shrouds | 2 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |
| | Uni | 2 | metal sheet to make mounting tabs | 2 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |
| T4S2 | Luke | 1 | Pvc pipe for belt roller | 1 | $0.00 | $0.00 | 0 | 23/8/23 | $0.00 | $0.00 | yes | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T4S2 | Sean | 1 | **19mm x 90cm Clear Vinyl Tubing** | 1 | $0.00 | $0.00 | 0 | 14/09/2023 | $0.00 | $0.00 | Yes | |
| T4S2 | Sean | 1 | Propeller 3D Printing | 1 | $0.00 | $0.00 | 0 | 14/09/2023 | $0.00 | $0.00 | No | |

*Figure 6 - Figure from requirements doc*