



AN-NAJAH NATIONAL UNIVERSITY  
FACULTY OF ENGINEERING & INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER ENGINEERING

GRADUATION PROJECT II

# Bike Emulator



PREPARED BY:

**Masa Lubbadeh**

**Raya Hindi**

SUPERVISED BY:

**Dr. Anas Toma**

Presented in partial fulfillment of the requirements for  
Bachelor's Degree in Computer Engineering

September 7, 2024

---

## Acknowledgment

We thank all who contributed to the completion of this project. We are very grateful for our supervisor Dr. Anas Toma for guiding us through this project with continuous follow-up, support, and constructive feedback. We also want to thank our professors in computer engineering department for all their work and support through the past 4 years. We are very grateful for the support of our family, friends. At last, a special thanks to engineer Abdullah Hennawi for his constant support and help during this project, and for Mr Alaa Zorba for his help and contribution in the project.

---

## **Disclaimer**

This report was written by Masa Lubbadah and Raya Hindi at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the student(s). An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

---

## Abstract

Biking is a widely popular sport that people enjoy. Yet outdoor cycling can be challenging especially in Palestine due to many obstacles like bad weather, dangerous roads, safety concerns triggered by occupation, or simply not feeling like going outside. BikeEmulator is an enhanced bike that combines sport and entertainment, it provides the user with a complete outdoor cycling experience, but from the comfort of their home or a preferred location, featuring integrated drive emulation technology. Our objectives for this project include enhancing the enjoyment of biking activities, enabling indoor cycling experiences, and delivering a complete outdoor riding experience. We began the process with a normal bike that we fixed using a base we designed, connecting the bike to sensors, actuators and a controller. Using the readings from them we managed to make the experience more realistic. Laser and LDR sensors are used to determine the speed of the bike, another sensor is used to detect the direction of the steering, stepper motors are used to control break and Shift cables, and for more realistic experience, a fan on the bike will blow air on the biker manually or based on the surrounding temperature.

The user will ride it like a normal bike, the actions performed will be mirrored into the drive simulation, data will be displayed on LCD screen for the user to view, it includes heart rate and the time spent playing. From a keypad the user can select the preferred play mode, there's also a flashlight feature if the bicycle detects dim lighting.

Combining drive simulation with stationary gym bikes already exists, but it doesn't provide a full experience like ours. In our bike, the user has the ability to steer left and right in the simulation, in addition to controlling braking actions.

# Contents

<b>List of Figures</b>	<b>6</b>
<b>1 Introduction</b>	<b>9</b>
1.1 Problem Statement . . . . .	9
1.2 General Background . . . . .	9
1.3 Objectives (Purpose or Aims) of the Work . . . . .	10
1.4 Significance or importance of your work . . . . .	10
1.5 Organization of the report . . . . .	10
<b>2 Theoretical Background and Previous Work</b>	<b>11</b>
<b>3 Methodology</b>	<b>12</b>
3.1 System Structure . . . . .	12
3.1.1 Bicycle . . . . .	12
3.1.2 Hardware Storage box . . . . .	12
3.1.3 Bicycle Support Stand . . . . .	13
3.1.4 Light Shield Box . . . . .	13
3.1.5 Speed Measurement Triangle . . . . .	13
3.1.6 Interactive Display Panel . . . . .	14
3.2 Hardware Components . . . . .	14
3.2.1 Arduino Due . . . . .	14
3.2.2 Power Supply . . . . .	15
3.2.3 LDR Sensor Module . . . . .	16
3.2.4 Laser Module . . . . .	16
3.2.5 Rotary Encoder . . . . .	16
3.2.6 4-Digit 7-Segment Display . . . . .	17
3.2.7 Liquid Crystal Display (LCD) . . . . .	17
3.2.8 4X4 Keypad . . . . .	17
3.2.9 Nema 23 Stepper Motor . . . . .	17
3.2.10 TB6600 Stepper Motor Driver . . . . .	18

3.2.11	RGB LED strip . . . . .	19
3.2.12	Relays . . . . .	19
3.2.13	MAX30102 Heart-Rate Sensor . . . . .	20
3.2.14	ESP-WROOM-32 . . . . .	20
3.2.15	Bicycle Lights . . . . .	20
3.2.16	Push Button . . . . .	20
3.2.17	RFID RC522 . . . . .	22
3.2.18	DHT22 Temperature sensor . . . . .	22
3.3	Mobile Application . . . . .	22
3.3.1	Start Page . . . . .	23
3.3.2	User section . . . . .	23
3.3.3	Admin section . . . . .	24
3.4	How the system works . . . . .	24
3.4.1	Brake system . . . . .	25
3.4.2	Gearing System And Mode Changing . . . . .	25
3.4.3	Controlling Fan . . . . .	28
3.4.4	Controlling Speed and Direction Of The Bicycle . . . . .	28
3.5	Constraints and Limitations . . . . .	29
3.5.1	Lack of Mechanical Experience . . . . .	29
3.5.2	Lack Of Time . . . . .	29
3.5.3	Cost . . . . .	29
3.5.4	Measurement Challenges . . . . .	29
3.6	Standards / Codes . . . . .	29
<b>4</b>	<b>Results and Discussion</b>	<b>31</b>
<b>5</b>	<b>Conclusion and Recommendations</b>	<b>33</b>
5.1	Conclusion . . . . .	33
5.2	Recommendation . . . . .	33
	<b>Bibliographic</b>	<b>34</b>
	<b>Appendix</b>	<b>35</b>
A	A . . . . .	35

# List of Figures

3.1	Bicycle Used In Our Project . . . . .	12
3.2	Hardware Storage box . . . . .	13
3.3	Bicycle Support Stand . . . . .	13
3.4	Light Shield Box . . . . .	14
3.5	Speed Measurement Triangle . . . . .	14
3.6	Interactive Display Panel . . . . .	14
3.7	Bicycle's Front End Anatomy . . . . .	14
3.8	Arduino Due Pinout . . . . .	15
3.9	Computer Power Supply . . . . .	15
3.10	LDR Module . . . . .	16
3.11	Laser Module . . . . .	16
3.12	Rotary Encoder Base . . . . .	17
3.13	Rotary Encoder Base . . . . .	17
3.14	Rotary Encoder Setup . . . . .	17
3.15	Time Display . . . . .	17
3.16	LCD with I2C module . . . . .	18
3.17	Keypad . . . . .	18
3.18	Stepper Motor . . . . .	18
3.19	Coupler Used With Stepper Motor . . . . .	18
3.20	Stepper Motor Driver . . . . .	19
3.21	RGB LED Strip . . . . .	19
3.22	Relay . . . . .	19
3.23	MAX30102 . . . . .	20
3.24	ESP-WROOM-32 . . . . .	21
3.25	Bicycle Lights . . . . .	21
3.26	Push Button . . . . .	21
3.27	RFID RC522 . . . . .	22
3.28	DHT22 Temperature Sensor . . . . .	22

3.29 Start Page . . . . .	23
3.30 Book Bicycle Screen . . . . .	23
3.31 User Screen1 . . . . .	24
3.32 View Readings Screen . . . . .	24
3.33 Change Mode Screen . . . . .	25
3.34 Admin Screen . . . . .	25
3.35 System Flow Chart . . . . .	26
3.36 System Flow Chart . . . . .	27
3.37 Brake Flow Chart . . . . .	27
3.38 Parts of the Bike Drivetrain . . . . .	28
3.39 FAN Flow Chart . . . . .	28

# List of Tables

# Chapter 1

## Introduction

### 1.1 Problem Statement

Traditional workout routines have changed in recent years due to the integration of technology, which has made them more data-driven and interactive. Our project aims to convert a normal bike into an advanced gym bike with drive emulation, allowing the user to live the full experience as if he's riding his bike outside.

### 1.2 General Background

In recent years, more and more people are paying attention to their health, and the popularity of home gyms has increased. This switch brought with itself a demand for fitness equipment that didn't just help people exercise but also all those devices which would tell them something about their health, keeping track of their progress, and bringing them closer to reaching their goals. As a result, the market has been flooded with new gym tools, each one trying to outshine the others with the latest features and technology. However, these advanced tools often come with a high price tag, putting them out of reach for many people.

Cycling is great for fitness because it's easy on the joints but gives one a good cardiovascular workout. It would thus be ideal for upgrades, especially by the addition of sensors and displays that turn an ordinary bicycle into a smart exercise machine showing you, in real time, your speed, distance, calories burnt, heart rate, and time. This way, you'll get to track your performance, set goals, and see your progress as you ride.

This Project targets this highly advanced feature to be made available for the masses and puts an attempt toward creating an affordable alternative. Instead of investing in expensive gym equipment, you can use a bike you already own and turn it into a sophisticated fitness tool. This approach will not only save money, but will also make high-tech fitness available to one and all.

### 1.3 Objectives (Purpose or Aims) of the Work

The primary objective of this project is to transform a standard bicycle into an advanced exercise machine that provides real-time feedback on various fitness metrics. We seek the ability to offer users a motivating and interactive workout environment, similar to high-end gym equipment, but at a fraction of the cost, simply by incorporating sensors and displays. Our goals are to:

- **Measure and Display Key Metrics in Real-Time:** Speed, distance, calories burned, heart rate, and workout duration.
- **Enhance User Experience:** Implement a drive emulation system to simulate outdoor cycling, making indoor workouts more engaging and realistic.
- **Provide an Affordable Solution:** Offer a cost-effective alternative to expensive gym equipment, making advanced fitness technologies accessible to a wider market.

By achieving these objectives, we believe that more people will have the drive to be regularly physically fit and monitor their levels of fitness with efficiency.

### 1.4 Significance or importance of your work

This project fills a specific current need in the market for fitness solutions by providing a reasonably priced solution that will bring advanced exercise features into the regular consumer's home. With the increasing trend of personalized fitness tools and home gyms, many individuals are on the lookout for cost-effective measures to enhance their workloads without having to spend excessively.

Our solution saves money by democratizing advanced fitness devices. With the ability to recycle an existing bicycle and add modern features based on what we have to offer, we increase end-user groups for advanced fitness tracking and simulation and allow users to maintain motivation and set and achieve fitness goals more economically.

### 1.5 Organization of the report

This report is well organized to explain how our system works, starting with the introduction and stating the problem that made us consider doing this project and then giving a brief background about our project. Then in the literature review chapter we talked about some researches that were done about some parts of our project. From there, we go to the methodology chapter in which we give a detailed explanation about the hardware components in our project and what are they used for and briefly explained how we implemented a our software. Then in the result chapter we stated out results that we got from testing our project. The results were followed by discussion and analysis and finally in the conclusion section we stated our conclusions and gave suggestions for the future work

## Chapter 2

# Theoretical Background and Previous Work

The integration of smart technologies into bicycles has become increasingly common. For example, Chou et al. [1] developed bicycle motion analysis model, they used a test bicycle equipped with multiple sensors so they could measure the vertical acceleration while the user rides the bike,

Bike simulator and emulators provide an environment that closely mimics real cycling conditions for research purposes [2], We discovered many studies that explore this area. A recent study by Dian et al. [3] presented virtual reality bike simulator, combined with arcade game that uses only the bicycle handlebar as a full control, they used an Arduino connected to CMPS12 sensor and Optocoupler sensor, the authors wanted to show that they could combine virtual cycling simulator with an arcade-style game to reduce cybersickness. Another study, conducted by Liu et al.[4], employed advanced software for bicycle simulation. This study found that the knowledge gained from these simulations might greatly improve cycling techniques, and help in developing injury prevention Measures in sports.Also Wang et al .designed simulation for sports bicycle through three development processes, [5].

Bike emulators aim at replicating the pedaling experience even when the user is pedaling on a stand, [6], Radrizzani et al 's research focused on bike emulation, very similar to what we did in our project. Their major challenge was reproduce the effect of the freewheel.

Ranky et al. claim that there are no such modular exercise bicycle systems with sensors integrated into the application, which can measure physiological data, such as heart rate and biomechanical data in kinetics and kinematics, [7]. In this study, physiological and biomechanical parameters were integrated by using hardware components with an embedded sensor, applied on a stationary exercise bicycle, which is integrated with virtual environments for rehabilitation purposes.

# Chapter 3

## Methodology

### 3.1 System Structure

#### 3.1.1 Bicycle

The bicycle is the primary component of the project, into which We have integrated various hardware elements to enhance its functionality and performance, adapting it to meet the specific requirements of our project.



Figure 3.1: Bicycle Used In Our Project

#### 3.1.2 Hardware Storage box

This 20x18 cm wooden box is designed to store most of the hardware component that we used, we chose its size to provide enough space while ensuring it does not obstruct the steering or add excessive weight



Figure 3.2: Hardware Storage box

### 3.1.3 Bicycle Support Stand

The main concept of our project involves converting a regular bicycle into a stationary exercise bike for interactive use. In order to ensure the bicycle remains in place during operation, we used a wooden stand to securely elevate the back wheel.



Figure 3.3: Bicycle Support Stand

### 3.1.4 Light Shield Box

LDRs (Light Dependent Resistors) can be easily influenced by surrounding light, which can interfere with the accuracy of our speed measurements, therefore we designed a shield that blocks external light, as Shown in figure 3.4. Next to it we can also see a small wooden stand, we used it to position the second motor on it.

### 3.1.5 Speed Measurement Triangle

To ensure accurate detection by the LDR positioned across from the laser as the wheel spins, we needed a more effective way to track changes. The lines on the wheel were too close together for precise detection. Therefore, we crafted a cardboard triangle and attached it to the wheel. It's shown in 3.5. This larger surface covers about 12 lines on the wheel, allowing

the system to detect changes more effectively and significantly enhancing the accuracy of our speed measurements.

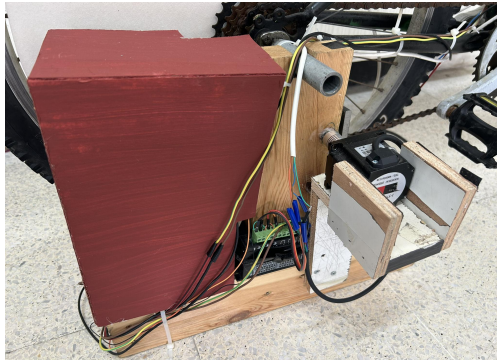


Figure 3.4: Light Shield Box



Figure 3.5: Speed Measurement Triangle

### 3.1.6 Interactive Display Panel

We positioned several components in this panel, LCD, 4-digit 7-segment display, fan and the keypad. As shown in figure 3.6 We positioned it on the stem and the handlebar which shown in 3.7.

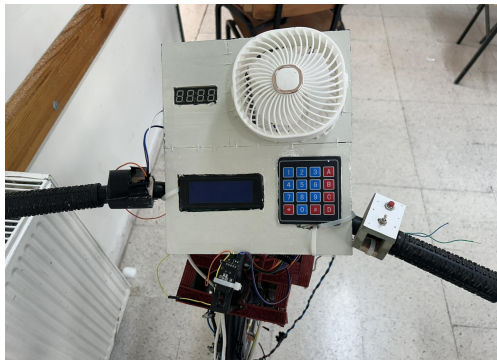


Figure 3.6: Interactive Display Panel

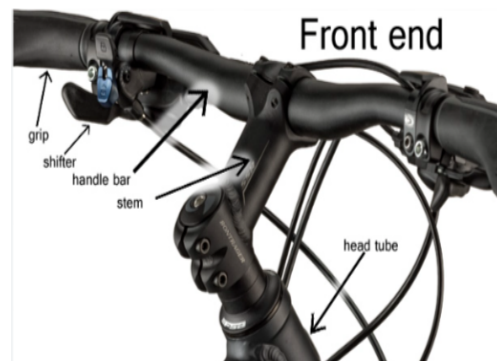


Figure 3.7: Bicycle's Front End Anatomy

## 3.2 Hardware Components

### 3.2.1 Arduino Due

The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU. It is the first Arduino board based on a 32-bit ARM core microcontroller. It has 54 digital input/output pins (of which 12 can be used as PWM outputs), 12 analog inputs, 4 UARTs (hardware serial ports), a 84 MHz clock, an USB OTG capable connection, 2 DAC (digital to analog), 2 TWI, a power jack, an SPI header, a JTAG header, a reset button and an erase button.[8]

We mainly used Arduino Due because it has the ability to act as a USB host for peripherals such as a keyboard connected to the SerialUSB port. Therefore, we used the Native USB port on it to send commands from the arduino to the keyboard to control the game that the user chose.

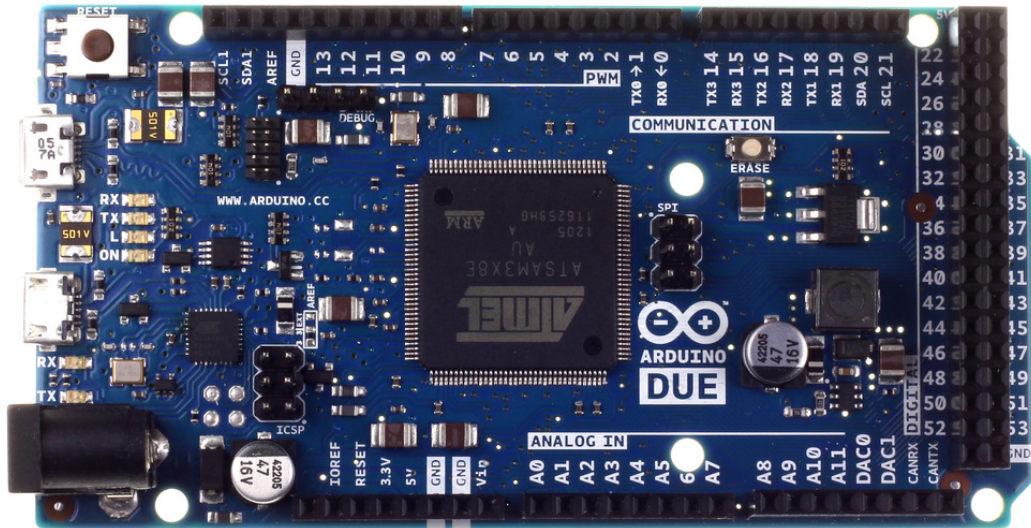


Figure 3.8: Arduino Due Pinout

### 3.2.2 Power Supply

ISO-450 ATX computer power supply provides stable power for various components in the system. It delivers 5V at 32A and 12V at 16A, it's well-suited for providing the necessary power for the components in our project.



Figure 3.9: Computer Power Supply

### 3.2.3 LDR Sensor Module

The LDR Sensor Module is used to detect the presence of light / measuring the intensity of light. The output of the module goes high in the presence of light and it becomes low in the absence of light. The sensitivity of the signal detection can be adjusted using a potentiometer if you are using the analog signal but we controlled it using the digital signal.

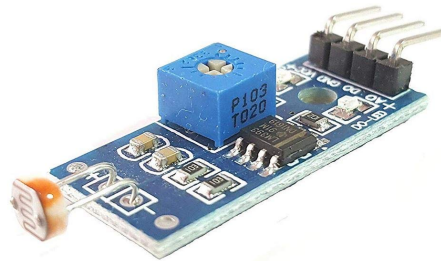


Figure 3.10: LDR Module

We used two LDRs for the following tasks: The first task is to measure the speed of the wheel, with a laser pointing to it and with the help of the Speed Measurement Triangle. The Second task is for light detection, if the light around it has changed and got darker, a notification on the LCD will appear to ask the user if he wants to open the flash light, and when the lighting around the bicycle gets higher the flash will turn off on its own.

### 3.2.4 Laser Module

A device that emits a single static laser beam of single or multiple wavelengths. Shown in [3.11](#)

### 3.2.5 Rotary Encoder

The Rotary Encoder is a device that measures the angular position of rotation. We 3D printed a base for it to set it up on the steering of the bicycle as shown in [3.12](#) [3.13](#) [3.14](#)

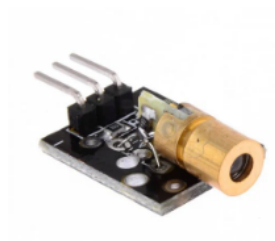


Figure 3.11: Laser Module

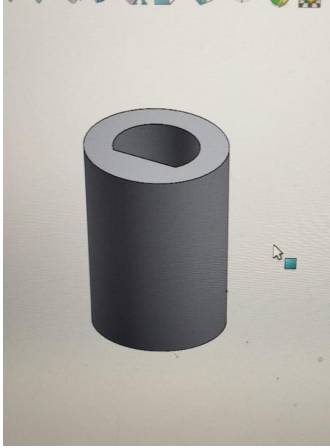


Figure 3.12: Rotary Encoder Base



Figure 3.13: Rotary Encoder Base



Figure 3.14: Rotary Encoder Setup



Figure 3.15: Time Display

### 3.2.6 4-Digit 7-Segment Display

4-Digit 7-Segment Display is used to display numbers or simple alphanumeric characters, we used it to display the session time for the user [3.15](#) .

### 3.2.7 Liquid Crystal Display (LCD)

The LCD is like a flat panel display which uses liquid crystals in its primary form of operation, we used a 20x4 LCD with I2C module which displays 20 characters per row, with 4 rows and we used it to display to the user multiple screens. [3.16](#)

### 3.2.8 4X4 Keypad

A 4x4 keypad is a matrix keypad which contains 16 buttons arranged in 4 columns and 4 rows. Its used for the user input in a project. We used it to control the LCD and other hardware components. [3.17](#)

### 3.2.9 Nema 23 Stepper Motor

The J-5718HB2401 is a NEMA 23 stepper motor, It operates at a current of 3.5A with a resistance of 0.8, providing a holding torque of 1.3 N.m. The motor offers high resolution and

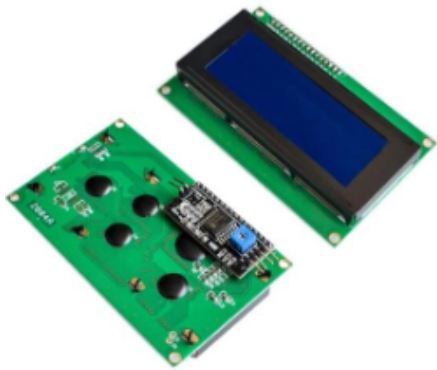


Figure 3.16: LCD with I2C module



Figure 3.17: Keypad

accurate control in positioning tasks with a step angle of  $1.8^\circ$ , It's also known for its precision and reliability in applications requiring accurate motion control.



Figure 3.18: Stepper Motor



Figure 3.19: Coupler Used With Stepper Motor

We used two stepper motors for the following tasks: The first motor is used to control the brake cable of the bicycle. The second motor is used to control the gearing mechanism, allowing the player to select the desired mode of the bicycle, according to the level of difficulty that they prefer.

### 3.2.10 TB6600 Stepper Motor Driver

The TB6600 is an easy-to-use stepper motor driver that supports both speed and direction control. It is compatible with Arduino and other microcontrollers that can output a 5V digital pulse signal. The driver supports a wide input voltage range, making it adaptable for different power supplies. In addition, the motor can handle motor currents of up to 4.5A, which is useful for driving larger stepper motors.

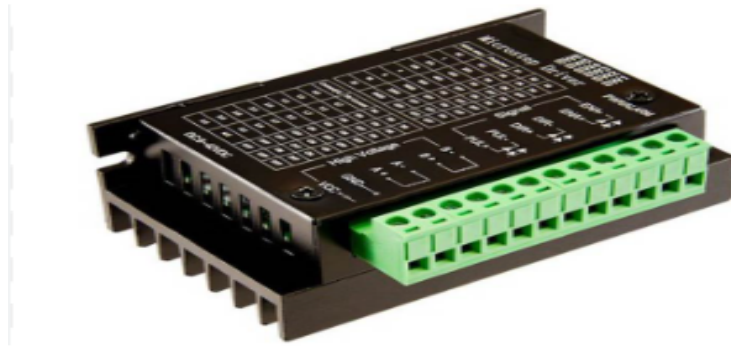


Figure 3.20: Stepper Motor Driver

### 3.2.11 RGB LED strip

RGB LED strips produce three hues of light: red, green, and blue. We used it to indicate the state of the bicycle, whether the bicycle is currently in use or available.



Figure 3.21: RGB LED Strip

### 3.2.12 Relays

A relay is an electrically controlled device that opens and closes electrical contacts, or activates and deactivates operation of other devices, in our project we used 2 relays each 2 channels to control the activation of the fan, bicycle front light, and to control colors in RGB cable.

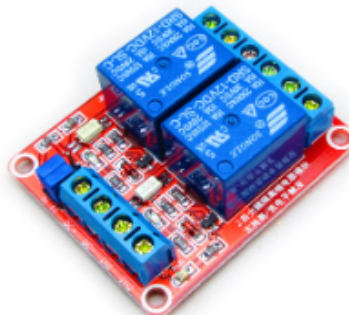


Figure 3.22: Relay

### 3.2.13 MAX30102 Heart-Rate Sensor

The MAX30102 is a heart-rate biosensor module, designed to monitor oxygen saturation and heart rate values, it operates on 3.3v or 5v and it communicates through an I2C interface. While playing, users can monitor their heart rate and oxygen saturation values either through LCD or the application.

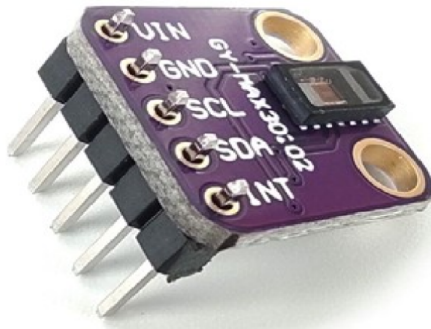


Figure 3.23: MAX30102

### 3.2.14 ESP-WROOM-32

The ESP-WROOM-32 is a powerful little module that combines Wi-Fi, Bluetooth, and Bluetooth Low Energy into one compact package, it's compatible with the 802.11 b/g/n standard at 2.4 GHz. This module can handle complex tasks while staying energy-efficient. It includes multiple interfaces such as SPI, I2C, UART, and ADC, therefore, we can easily connect it to sensors, motors, and other devices due to the variety of inputs and outputs. It's shown in [3.24](#); In our project, we used the ESP-WROOM-32 to connect the system wirelessly to the App Inventor application we developed. Also, we connected the Heart Rate sensor to the ESP-WROOM-32 since it required an I2C connection, and the Arduino only had one available, which we already used for the LCD.

### 3.2.15 Bicycle Lights

Bicycle Lights is a small module that consists of 2 LEDs, as shown in [3.25](#), we positioned the light on the handlebar. When the LDR senses low light levels, notification appears on the LCD screen, asking the user to decide whether to turn on the light or keep it off.

### 3.2.16 Push Button

A push button connects two points in a circuit when pressed resulting in triggering an action. We positioned it on a small stand near the grips, to serve as a brake so the user press it to activate the brake as needed. It's shown in [3.26](#)

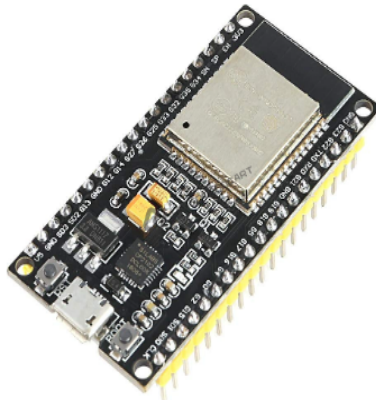


Figure 3.24: ESP-WROOM-32



Figure 3.25: Bicycle Lights



Figure 3.26: Push Button

### 3.2.17 RFID RC522

The module is designed to create an electromagnetic field and can be used for almost all kinds of 13.56MHz passive ID cards and tags. It is low power, low cost and easy to interface. Each user receives a card from the coach in the gym, which contains the time period and chosen mode that the player will be playing, When the user scans their card with the module, the LCD screen start a personalized session with them.

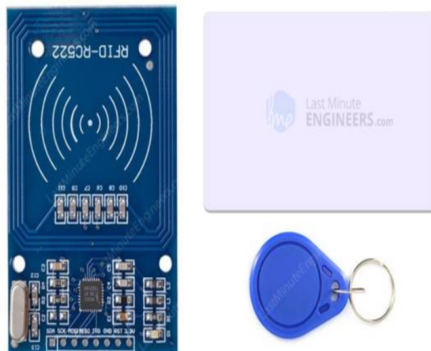


Figure 3.27: RFID RC522

### 3.2.18 DHT22 Temperature sensor

Temperature and humidity sensor that provides accurate temperature readings. We used it to control the fan so when the temperature rises the fan turns on. [3.28](#)

## 3.3 Mobile Application

We created this application with app inventor, to allow the coach and the player to control some features remotely. We have 6 interfaces as follow:

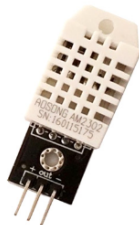


Figure 3.28: DHT22 Temperature Sensor

### 3.3.1 Start Page

This is the first page in the application, it includes 2 buttons that lead to 2 different screens.

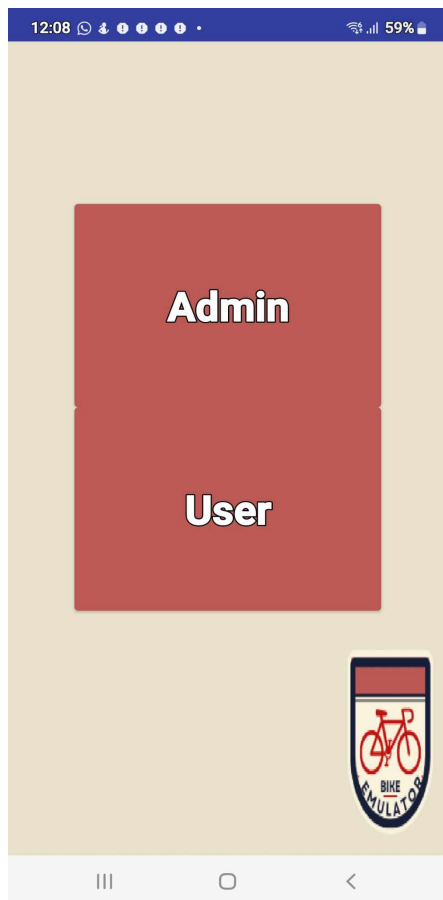


Figure 3.29: Start Page

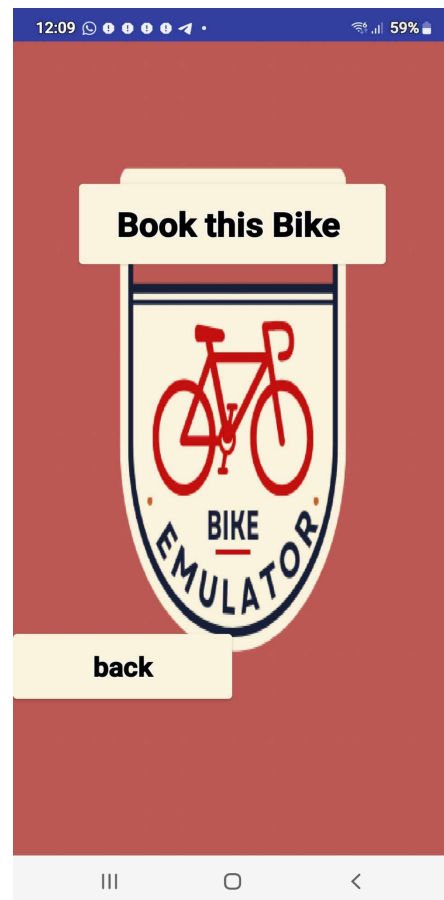


Figure 3.30: Book Bicycle Screen

### 3.3.2 User section

When "User" button is clicked in start page, this screen is opened. The user book the bicycle through this screen, as a result, the color of the RGB turns to green indicating that a player is using this bicycle. Then it opens the next screen shown in 3.31.

Screen 3.31 gives the user 2 options to choose from, whether to view readings, or change bicycle's mode.

If the user click on "View Readings", that will lead to 3.32 screen, which displays heart rate and Oxygen Saturation Level readings. The screen also includes a "refresh" button that updates the displayed values when clicked.

However, if the user click on "Change Mode", the user will be directed to the screen shown in figure 3.33. The user has the ability to change the mode of the bicycle through this screen. (The process of changing the mode will be explained later).

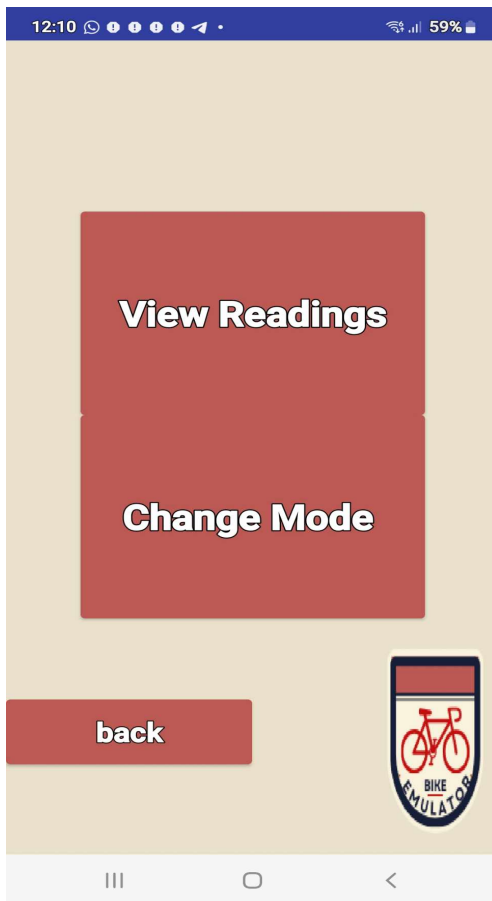


Figure 3.31: User Screen1

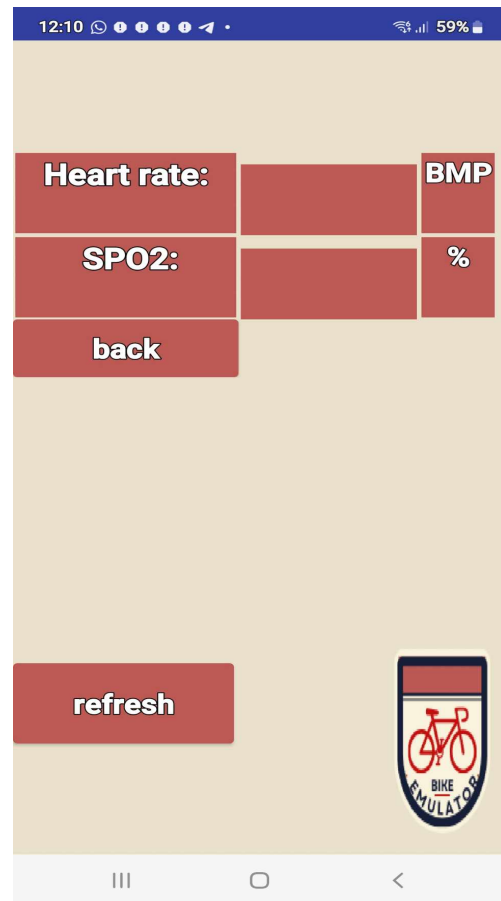


Figure 3.32: View Readings Screen

### 3.3.3 Admin section

Admin can remotely unbook the bicycle in case the player don't finish their session, when button in screen 3.34 is clicked, as a result, the color of the RGB turns to red indicating that this bicycle is now available for others to use.

## 3.4 How the system works

Arduino and esp32 are connected to power, the application is opened on a phone which is connected to the network provided by esp, the system is ready to be used by the user. In the beginning the RGB LED strip color is set to red, indicating that the bicycle is available to use by any user. The flowchart shown in 3.36 describes how the system works in general.

There are many processes and actions that the user can choose from during the session, including:

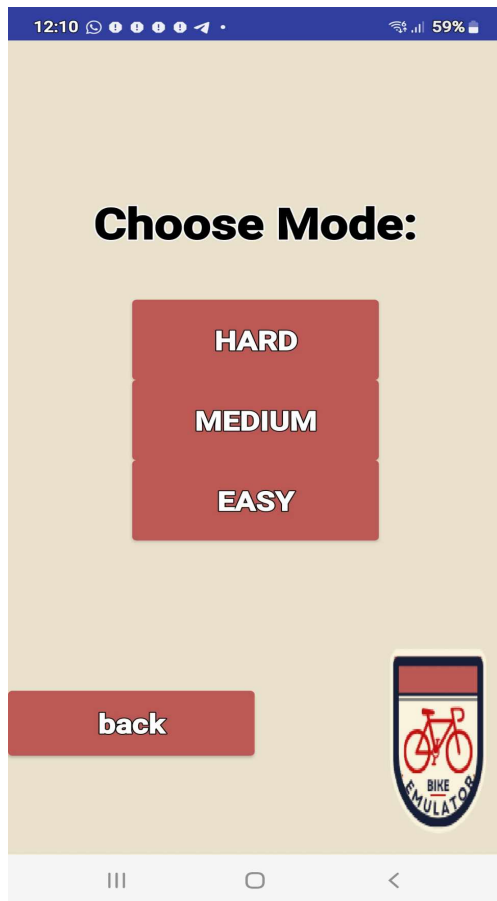


Figure 3.33: Change Mode Screen



Figure 3.34: Admin Screen

### 3.4.1 Brake system

The user can press the push button designed for the brake anytime, it's easy to access because we positioned it next to the grips. This will cause both the real bicycle and the one in the game to immediately stop. The flowchart is shown in 3.37.

To achieve this, we passed the brake cable through the coupler shown in Figure 3.19 and securely fixed the coupler around the motor shaft. With this setup, by rotating the motor in clockwise direction, brake cable is now pulled tight, and that will force the bicycle to stop, and by rotating the motor in counter clockwise direction, the cable is released, so the bicycle can move again. As shown in figure 3.37, first we pulled the cable for a second then released it.

### 3.4.2 Gearing System And Mode Changing

The user has the ability to change bicycle's mode anytime. Either through keypad or through the application.

This process was quite challenging, and it took us several attempts to get it right. We used a coupler, as shown in Figure 3.19, to securely attach the motor shaft. On the other end of the coupler, we connected a long threaded screw.

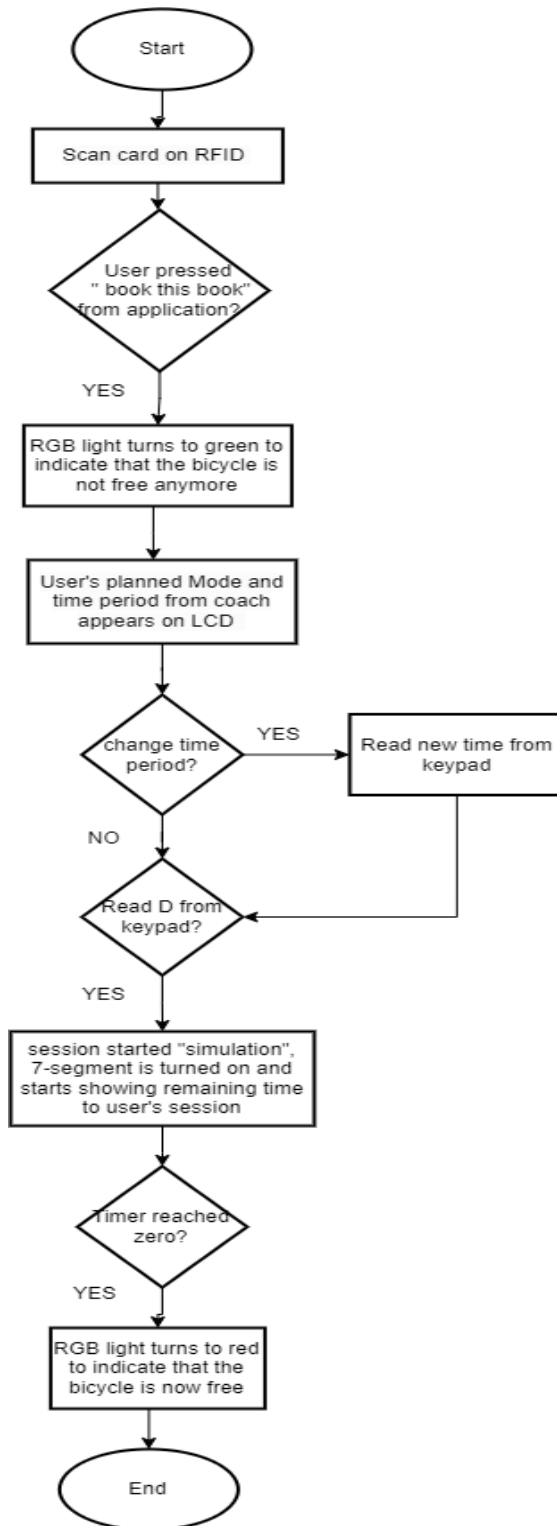


Figure 3.35: System Flow Chart

We fixed the motor onto a base positioned next to the rear wheel. To complete the setup, we fixed a nut onto the bicycle's support stand. When the motor rotates clockwise, the screw threads into the nut, pushing the rear derailleur and causing the chain to shift to a larger cassette. This results in easier pedaling, effectively changing the gear ratio. When the motor

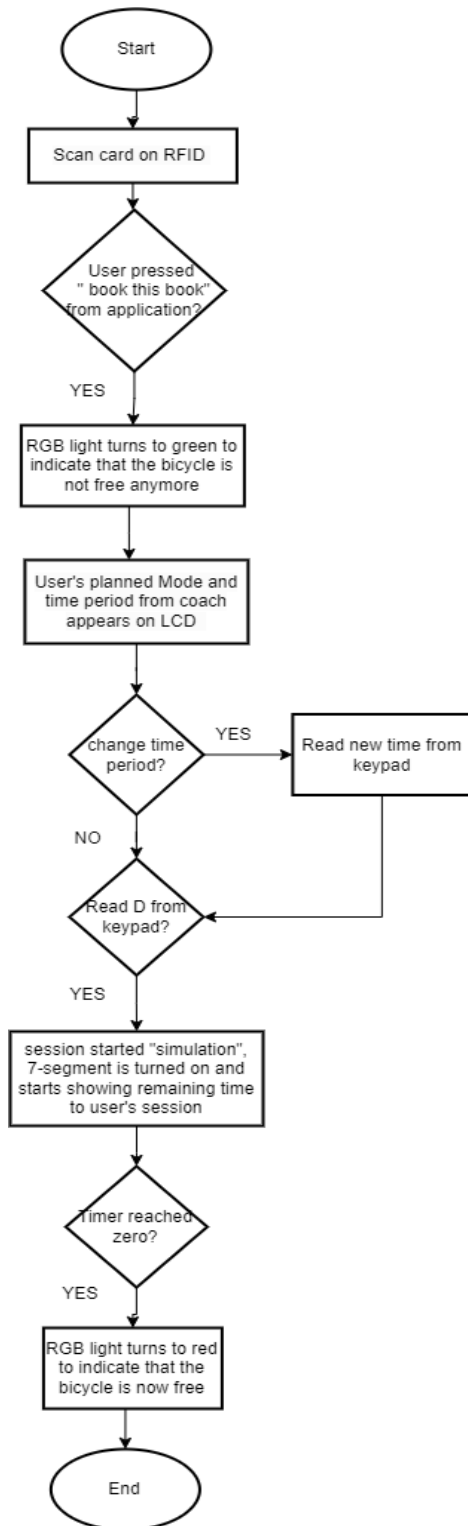


Figure 3.36: System Flow Chart

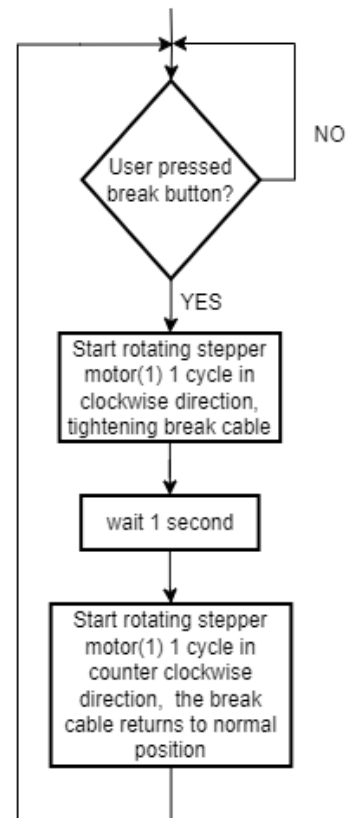


Figure 3.37: Brake Flow Chart

rotates in the opposite direction, the screw retracts, causing the derailleur to move the chain to smaller cassettes. This adjustment increases the gear ratio, making pedaling harder. As shown in figure 3.38 the rear components of the bicycle are illustrated.



Figure 3.38: Parts of the Bike Drivetrain

### 3.4.3 Controlling Fan

As we mentioned before we used temperature sensor, when the temperature of the room rises above  $28^{\circ}\text{C}$ , the fan is automatically activated, to encourage the user to continue playing comfortably, even in hot conditions. The user always has the ability to control the fan manually through keypad. Two keys are designated for this purpose: pressing '8' will turn the fan on, while pressing '9' will turn it off. Flowchart is shown in figure 3.39.

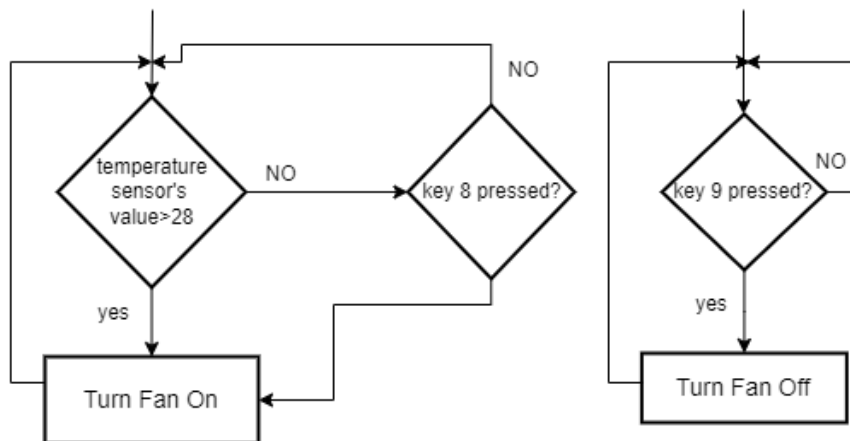


Figure 3.39: FAN Flow Chart

### 3.4.4 Controlling Speed and Direction Of The Bicycle

For measuring the speed of the bicycle we used LDR and Laser. When the laser gets interrupted by the speed measurement triangle, the LDR reading value will change, and we already have the circumference of the bicycle wheel so we did some calculations to measure the speed of the wheel, at the same time it also read values from the rotary encoder.

## 3.5 Constraints and Limitations

### 3.5.1 Lack of Mechanical Experience

Our primary focus before had been on software and electronics, so our limited experience with mechanical systems made it difficult at first to fully understand how the bicycle works, especially how the shifting and braking mechanisms operate. This had cost us a lot of time learning how to deal with the components of our project, as we continuously experimented with different methods to determine the most effective solution.

### 3.5.2 Lack Of Time

The timeline of the project was very short, less than 2 months, and much of that was used to develop the idea of the project. Additionally, many days were spent testing each component separately, and then integrating them with one another. The time spent on merging codes of all the components used was significant, especially that our project requires multiple modules to operate simultaneously.

Moreover, working under occupation further caused further delays, as strikes and difficult conditions caused us to lose many valuable days.

Due to limited time, we were unable to add all the features we were planning,

### 3.5.3 Cost

The cost of many components was relatively high, and there were additional expenses because some of the components got damaged or burned out after testing, so we had to replace them.

### 3.5.4 Measurement Challenges

Accurately measuring the speed of the bicycle using LDR and Laser required precise alignment, which was difficult to maintain due to the need to read LDR values rapidly and continuously, while also handling other components. After many attempts we found that implementing task management in our code was effective. We assigned each task a time slot to ensure all the tasks were processed effectively, but the problem wasn't resolved completely.

## 3.6 Standards / Codes

- We developed our code using Arduino IDE, enabling us to control the hardware components.
- We designed our mobile application using App Inventor platform, providing the users with the ability to control the system remotely.

- For communication, Message Queuing Telemetry Transport was our protocol, ensuring reliability and efficiency in data transmission.

## Chapter 4

# Results and Discussion

In this chapter, we will present the results of our project and discuss the outcome of each feature. Our focus will be on how the various functionalities—speed calculation, wheel rotation, heart rate measurement, and user interface—were integrated into our project to transform a regular bicycle into a smart gym bike. We will also discuss the problems that faced us and what we did to mitigate their impact.

The primary goal that we wanted to achieve in our project is that every component had to work simultaneously with other components and from here appeared the main challenge that faced us in the project. Knowing that the Arduino doesn't support multi-threading, we needed to find a similar solution. At first, we considered using interrupts, but it wasn't a convenient solution since we are using a lot of sensors if an interrupt will occur everytime a sensor reading changes it would be a disaster. Therefore, we found something similar to multi-threading which is using tasks. With the help of the TaskScheduler library we were able to get an almost accurate results like the results we wanted, but surely nothing is perfect so there were some other problems.

One of our main results achieved is that we wanted to get accurate readings so the emulation part will work very smoothly. At first, we weren't planning on using the laptop and we wanted the main server of our project to be a Raspberry-pi, but after dealing with it, we noticed that the raspberry pi couldn't handle the simple game that we wanted to do the emulation part on, keeping in mind that we used Raspberry Pi 4 Model B which is considered very good. After trying many solutions to increase the accuracy of the raspberry pi, we decided that we wanted to use the laptop or computer as our server. Then, we discovered that this is the ideal solution to give the user the ability to select any game he/she wants, and if we had used the Raspberry Pi instead of the laptop, we wouldn't have been able to implement this feature.

The results we got were fairly satisfactory, there were slight delays in performance but they

are bearable, knowing that they are caused from the software part of the project not the hardware. For the emulation part, there were two main results we wanted to achieve and to be very accurate, which are the speed and rotation measurements, we faced many problems and tried so many solutions to overcome them, and the final result was that we got an almost perfect results for the speed, and for the rotary encoder part, we got a very good results but not with the accuracy we wanted, this is due to the speed and rotations having to work at exactly the same time. One of the solutions we tried is that we got very accurate results from the rotary encoder but less accurate for the speed, but considering that the speed plays a vital part in our project, we decided to stick to the other solution that gave us more accurate results for the speed.

Another result we got that led us to make slight changes, is that at first we were displaying the session time for the user on the LCD, but since it needs to make an interrupt every second, it made system lags so much because it gets stuck in the update function. Therefore, we decided to display the session time for the user on a 4-digit 7-segment display and gave its own task so it won't affect the other system functionalities.

We also faced a problem that the Arduino only has one I2C module, at first it wasn't a problem and we connected both the LCD and the heart rate sensor in parallel with these two pins, but then this method stopped working so we connected the heart rate sensor with the esp instead. Another problem we faced is that for measuring the rotation of the steering of the bicycle, we first started by using MPU Sensor, but the readings weren't as accurate as we wanted. Therefore, we used Rotary Encoder.

And from there, another problem occurred, which is how to install the rotary encoder on the bike. Therefore, we 3D printed a base for it.

We also had a difficulty in finding a way to control the gearing system, we positioned the motor in 2 different locations before we settled on the current one. At first, our idea was to pull the shift cable similar to what we did with the brake cable, but the motor couldn't pull the cable as needed, so we fixed the motor as we explained before.

For the brake system, we first faced a problem of not finding the appropriate motor that is able to pull brake cable so we had to try several motors and that took a lot of time.

We have successfully implemented the features of controlling the brakes of the bicycle and its gears with very appealing results. Also, the fan and flash control.

## Chapter 5

# Conclusion and Recommendations

### 5.1 Conclusion

In conclusion, This project was designed to make everyday exercises easier and more entertaining. We managed to developed a bicycle emulation system, interfaced with a commercial stationary gym bike, which was basically a regular bicycle. The system provides the users with many features that enhance user comfort, and give him the

### 5.2 Recommendation

Due to the time constraints, we were unable to add all the features that we planned to have in our project, in addition to some features that need further enhancements. Here are some recommendations for improving the project:

- Implementing reverse playing feature for the user, to provide the user with a more realistic experience.
- Making some adjustments to the method used to measure the speed, to enhance the accuracy.
- Enhancing the bicycle's design. For example the hardware storage box has almost all the components inside, and that led to wires getting tangled, which makes it hard to track a component's wiring in case something went wrong.
- Adjusting the fan's speed based on based on pedaling speed , so with faster pedaling, the speed of the fan increases.
- Making wireless connection between the Arduino and the Laptop without the need for usb cable.

# Bibliographic

- [1] Chia-Pei Chou et al. “Simulation of bicycle-riding smoothness by bicycle motion analysis model”. In: *Journal of Transportation Engineering* 141.12 (2015), p. 04015031.
- [2] Alireza Ansariyar, Eazaz Vaziri, and Mansoureh Jeihani. “Systematic Review of Bike Simulator Studies”. In: *Ansariyar, A., Sadeghvaziri, E., & Jeihani, M.(2021). Systematic Review of Bike Simulator Studies. Journal of Science and Cycling* (2021).
- [3] Hendrayana Dian et al. “Simulation Bicycle Arcade Game with VR Bike”. In: *2022 International Symposium on Electrical, Electronics and Information Engineering (ISEEIE)*. 2022, pp. 183–186. DOI: 10.1109/ISEEIE55684.2022.00039.
- [4] Yung-Sheng Liu, Tswn-Syau Tsay, and Tsai-Chu Wang. “Muscles force and joints load simulation of bicycle riding using multibody models”. In: *Procedia Engineering* 13 (2011). 5th Asia-Pacific Congress on Sports Technology (APCST), pp. 81–87. ISSN: 1877-7058. DOI: <https://doi.org/10.1016/j.proeng.2011.05.055>. URL: <https://www.sciencedirect.com/science/article/pii/S1877705811009696>.
- [5] Jian Hua Wang et al. “Simulation about sports bicycle frame based on the experiments”. In: *Applied Mechanics and Materials* 37 (2010), pp. 1142–1147.
- [6] Stefano Radrizzani et al. “A concurrent controllers integration strategy to enhance the riding experience in bicycle driveline emulators”. In: *2021 IEEE Conference on Control Technology and Applications (CCTA)*. 2021, pp. 1043–1048. DOI: 10.1109/CCTA48906.2021.9659131.
- [7] Richard G Ranky et al. “Modular mechatronic system for stationary bicycles interfaced with virtual environment for rehabilitation”. In: *Journal of neuroengineering and rehabilitation* 11 (2014), pp. 1–16.
- [8] Arduino. *Arduino Due Documentation*. <https://docs.arduino.cc/hardware/du/>. Accessed: 2024-09-07. 2024.

# Appendix

A A

Text .....