Faculty of Electronics and Computer Technology and Engineering

DEVELOPMENT OF SHUTTLECOCK SPEED MONITORING SYSTEM FOR BADMINTON TRAINING USING IOT

AIRUL SHAZLI HIZAL BIN ABDULLAH SANI

Bachelor of Computer Engineering Technology (Computer Systems) with Honours

2024

DEVELOPMENT OF SHUTTLECOCK SPEED MONITORING SYSTEM FOR BADMINTON TRAINING USING IOT

AIRUL SHAZLI HIZAL BIN ABDULLAH SANI

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology with Honours

Faculty of Electronics and Computer Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UTeM

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN

PROJEK SARJANA MUDA II

Tajuk Projek	:	Development Of Shuttlecock Speed Monitoring System For Badminton Training Using Iot
Sesi Pengajian	:	2023/2024

Saya AIRUL SHAZLI HIZAL BIN ABDULLAH SANI mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

[/]	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
[ ]	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.
[ ]	**TIDAK TERHAD**	

Disahkan oleh:

(TANDATANGAN PENULIS)

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: Jalan Imam Tua
Kampung Morak
Palekbang 16040
Tumpat Kelantan

TS. NIZA BINTI MOHD IDRIS
PENSYARAH
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Tarikh : 14 Januari 2024

Tarikh :

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled “Development Of Shuttlecock Speed Monitoring System For Badminton Training Using IOT ” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Student Name	:	AIRUL SHAZLI HIZAL BIN ABDULLAH SANI
Date	:	14/1/2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering Technology (Conputer System) with Honours.

Signature :

Supervisor Name : TS. NIZA BINTI MOHD IDRIS

Date : 22/1/2024

Signature :

Co-Supervisor Name (if any) :

Date :

DEDICATION

I just wanted to take a moment to express my gratitude to three amazing people who have made a significant impact on my life.

Firstly, to my incredible parents, Che Siti Nor Binti Che Omar and Abdullah Sani Bin Che Ibrahim – your love and support mean the world to me. Mom, your kindness and resilience inspire me every day. Dad, your strength and encouragement have been my rock.

A big thank you to my supervisor, Ts Niza Binti Mohd Idris. Your guidance and support have been crucial, and I appreciate all you've done to help me succeed.

To my parents and supervisor – this achievement is as much yours as it is mine. Thank you for being awesome.

ABSTRACT

The main objective of this project is to develop a Shuttlecock Speed Monitoring System for Badminton Training Using IOT. This project uses Arduino Uno for the microcontroller, an infrared sensor for the detection of the shuttle, and a wifi module for connectivity to a software application that is being remotely monitored by any Android OS smartphone. As technology is advancing so it can ease the badminton player to train effectively without hiring a coach to train them. They also can monitor their power smash by calculating the speed of the shuttlecock crossing to the net. In order to achieve this, a wifi module is interfaced with the Arduino at the receiver end while on the transmitter end, an application on the cell phone receives calculated speed data to the phone to monitor the speed. By touching the specified button on the application, the loads can be be stored and delete data remotely through this technology.

ABSTRAK

Objektif utama projek ini adalah untuk membangunkan Sistem Pemantauan Kelajuan Shuttlecock untuk Latihan Badminton Menggunakan IOT. Projek ini menggunakan Arduino Uno untuk mikropengawal, sensor inframerah untuk pengesanan ulang-alik, dan modul wifi untuk penyambungan kepada aplikasi perisian yang dipantau dari jauh oleh mana-mana telefon pintar OS Android. Memandangkan teknologi semakin maju, ia dapat memudahkan pemain badminton berlatih dengan berkesan tanpa mengupah jurulatih untuk melatih mereka. Mereka juga boleh memantau rempuhan kuasa mereka dengan mengira kelajuan bulu tangkis melintas ke jaring. Untuk mencapai matlamat ini, modul wifi disambungkan dengan Arduino di hujung penerima manakala pada hujung pemancar, aplikasi pada telefon bimbit menerima data kelajuan yang dikira ke telefon untuk memantau kelajuan. Dengan menyentuh butang yang ditentukan pada aplikasi, beban data boleh disimpan dan dipadam dari jauh melalui teknologi ini.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts. Niza Binti Mohd Idris for their precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and Faculty of Technology and Engineering Electronic and Computer (FTKEK) for the financial support which enables me to accomplish the project. Not forgetting my fellow colleague, for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents, and family members for their love and prayer during the period of my study. I express my gratitude to my family and friends for their unwavering encouragement, understanding, and patience during this academic journey. Their constant support has been a source of strength and motivation.

Finally, I would like to thank all the staffs at the FTKEK, fellow colleagues and classmates, the faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

δ	-	Voltage angle
$	-	US Dollar
	-	

LIST OF ABBREVIATIONS

V	-	Voltage
Km/p	-	Kilometer per hour
g	-	gram
m/s	-	Meter per seconds
cm	-	centimeter

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Badminton is the one of top Three famous sports in Malaysia. Badminton is challenging sports because you need to get the points by hit the shuttlecock to area of opponent. In badminton, smash is the best way to get a point from opponent which required a good technique and strength. To know how hard and strong you are, you need to know the speed of shuttlecock. To measure that you need a device that can measure the speed of moving shuttlecock across the net.

This project revolves around utilizing an Arduino Uno and an IR sensor to accurately gauge the velocity of a shuttlecock. Subsequently, the gathered data is transmitted for analysis to determine the average speed of the shuttlecock via an IoT connection. The setup entails positioning two IR sensors, each comprising an IR Transmitter (IR LED) and an IR Receiver (Photo Diode), at a distance from one another on a particular side of the road (connected to the Arduino board). Whenever a shuttlecock passes through the sensors, both IR sensors are linked to the interrupt pin of the Arduino and they identify the descending waveform while calculating the duration between activating the Arduino's internal timer sensor. Once the shuttlecock advances and reaches the first sensor, the IR Sensor activates, marking the commencement of a timer that continues to record time until the shuttlecock arrives at the second IR Sensor.

Now speed is measured using a simple distance time relationship:

Speed = Distance ÷ Time

where , Speed= shuttlecock’s speed

Distance= Distance between sensors

Time= measured by Arduino

LCD is connected to Arduino and measured speed is shown on LCD Display . By simulating the distance between the two sensors to be 5 meters, we can calculate the speed at which the shuttlecock travelled from IR Sensor 1 to IR Sensor 2 as we already know the time of travel. All the calculations and data gathering are done by Arduino and the final result is displayed on a LCD Module after that it will send the data to application on device. The connection is done by ESP32.

1.2 Problem Statement

The problem addressed by the Shuttlecock Speed Monitoring System for Badminton Training using IoT is the limited availability of time and financial resources for individuals who aim to become professional or advanced badminton players. Training in badminton requires rigorous practice, with a focus on attack and defense techniques. The power smash, in particular, is a highly effective attack technique due to its speed and potential to challenge opponents' defensive abilities.

Professional players often hire coaches to receive proper and effective training. However, many individuals face constraints such as a lack of suitable training time or insufficient funds to afford coaching services. To overcome these challenges, the

Shuttlecock Speed Monitoring System for Badminton Training using IoT is designed to assist players who lack flexibility in their training schedules and financial resources.

This system offers a portable solution that can be utilized in any location with a badminton net. Its affordability and accessibility make it a viable option for players with limited financial means and time constraints. By using this system, players can engage in systematic training, enhance the quality of their practice sessions, and become more deeply engaged in the sport.

The goal of the Shuttlecock Speed Monitoring System for Badminton Training using IoT is to provide a cost-effective and versatile tool that empowers players to train effectively, regardless of their financial situation or availability of training resources. By utilizing this system, players can improve their skills, enhance their training experience, and advance their performance in the sport of badminton

1.3 Project Objective

1. To design a system that can measure the speed of moving shuttlecock.
2. To produce a system that can continuously monitor the movement speed of shuttlecock across the net.
3. To analyze the consistent of detection of the shuttlecock across the net.
4. To analyze the system performance of speed monitoring device and data transmitted by IOT.

1.4 Scope of Project

The scope of this project are as follows:

a) IR sensor to detect the movement of the shuttlecock across the net.

b) Arduino Uno to calculate the speed of the shuttlecock

c) Wi-Fi module to transmit the calculate data to database and phone apps.

d) Phone apps to receive the data that has been transmitted and displays the result.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In recent years, technology has transformed almost every aspect of our lives, from how we work and communicate to how we learn and entertain ourselves. Some examples of modern technologies include Artificial intelligence (AI) and machine learning (ML). These technologies enable machines to perform tasks that traditionally required human intelligence, such as image recognition, speech recognition, and decision-making. Then, Internet of Things (IoT) refers to the interconnected network of physical devices, vehicles, and appliances that can communicate with each other and exchange data over the internet. After that, Cloud computing allows users to store and access data and applications over the internet, rather than on local hardware. Next is Virtual and augmented reality. These technologies create immersive and interactive experiences by simulating real-world environments or overlaying digital content onto the real world. Lastly, Blockchain technology is used to securely store and share data and transactions across a network of computers, without the need for a central authority. These are just a few examples of the many technologies that are changing the way we live, work, and interact with each other.

2.2 Arduino Atmega-328 Microcontroller

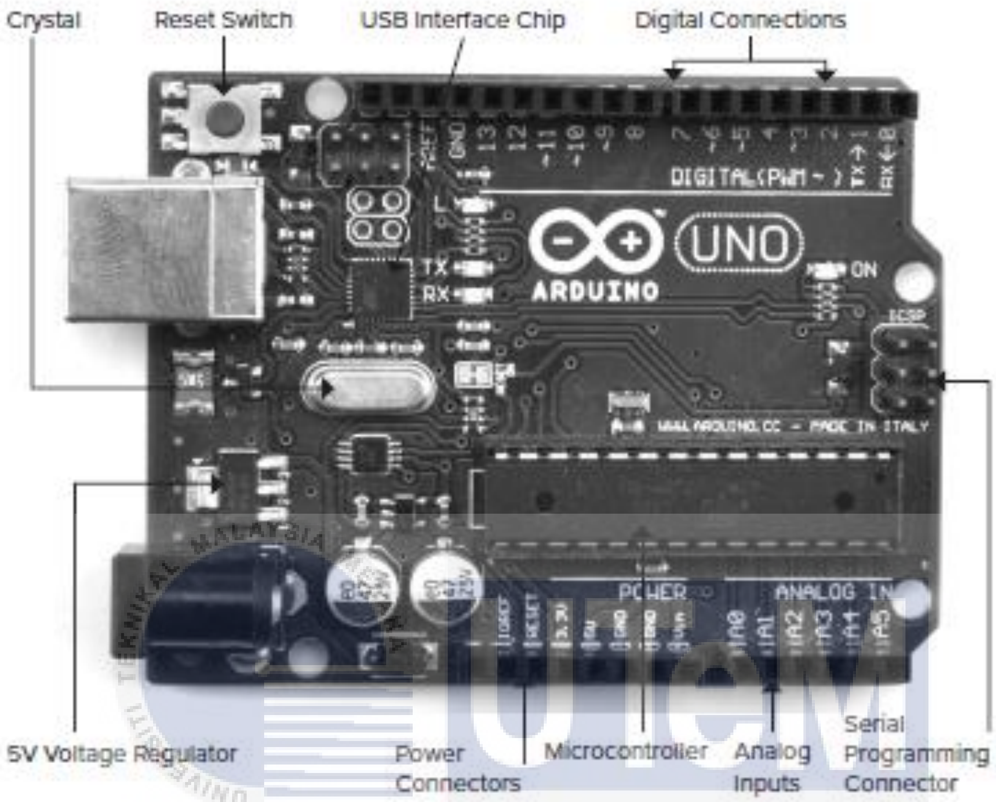


Figure 2.1 Arduino

The Arduino ATMEGA-328 microcontroller has gained significant popularity due to its versatility in various applications. To facilitate the execution of programs, the Arduino microcontroller is programmed using a power jack cable. The market offers a wide range of Arduino board options, each catering to different needs. However, this paper focuses on providing a comprehensive and detailed description of the Arduino UNO ATMEGA-328 microcontroller[1]. In order to program the Arduino, the Arduino software is installed on a computer, allowing users to edit and upload programs tailored to specific applications. Notably, the Arduino software supports both C and C++ programming languages, expanding its programming capabilities. With multiple input and output ports available on the Arduino board, users can simultaneously utilize up to eight ports for a variety of applications. Some examples of applications commonly implemented with Arduino boards include controlling the rotation of a general motor, operating a stepper motor, and managing valve openings, among others.[2]

The versatility of inputs in Arduino allows for the reading of both digital signals (indicating switch status) and analog signals (measuring voltage at a pin). This versatility enables the connection of various sensors for light, temperature, sound, and more. On the other hand, outputs can be either analog or digital. Digital outputs allow you to control pins to be either on or off, directly activating light-emitting diodes (LEDs) or driving higher power devices such as motors. Additionally, analog outputs grant the ability to control power output on a pin, providing fine control over motor speed or light brightness rather than simple on/off functionality. The central microcontroller on an Arduino Uno board is a 28-pin chip housed in a socket. This single chip encompasses the memory, processor, and all necessary electronics for input/output operations. Notably, the Arduino system offers a standardized approach to programming a wide range of microcontrollers, extending beyond official Arduino boards. This means that, with a few exceptions, you can program any microcontroller as an Arduino without the need to learn proprietary software tools specific to each manufacturer.[3]

2.3 IR Sensor Working Principal & Applications

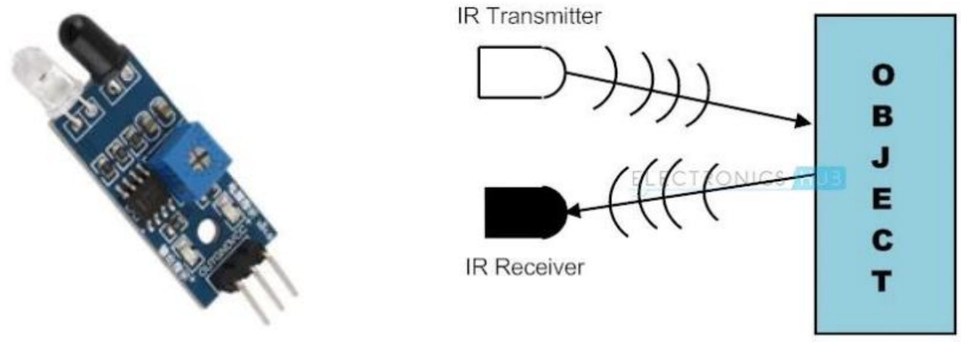


Figure 2.2 IR Sensor

The infrared sensor (IR), an essential electronic device, effectively detects and analyzes various attributes of its environment through the emission and reception of Infrared radiation. By precisely sensing specific light wavelengths within the infrared spectrum, ranging from 0.75 to 1000 micrometers, IR sensors enable accurate detection and measurement of relevant characteristics in their surroundings.[4]

The elucidation of an IR sensor's role as an Object Detection Sensor can be unveiled through the following description. As previously mentioned, this remarkable sensor comprises a fascinating duo: an IR LED and an IR Photodiode, which are lovingly referred to as a Photo-Coupler or Opto-Coupler. When an obstinate obstacle audaciously intrudes upon the sensor's path, a fraction of the splendid IR rays emanating from the LED gracefully pirouettes back to the receiver. The sensor's output dances and twirls in sync with the intensity of these reflected rays, which are gallantly embraced by the receiver. In this grand spectacle, an operational amplifier IC, splendidly performing as a voltage comparator, elegantly compares the radiant intensity of the reflected rays with the threshold value diligently set by the photodiode acting as the sensor's calibration maestro and gracefully orchestrates the production of a befitting output signal. Furthermore, the strategic placement of the IR transmitter and receiver becomes an exquisite choreography in the quest to detect radiant waves. When these noble components are poised directly opposite each other, engaging in a symphony of direct incidence, it ensures that nearly all the radiant waves emitted by the transmitter tenderly caress the receiver, as they embark on an ethereal journey of detection and discovery.[5]

2.4 Internet of Things (IoT)

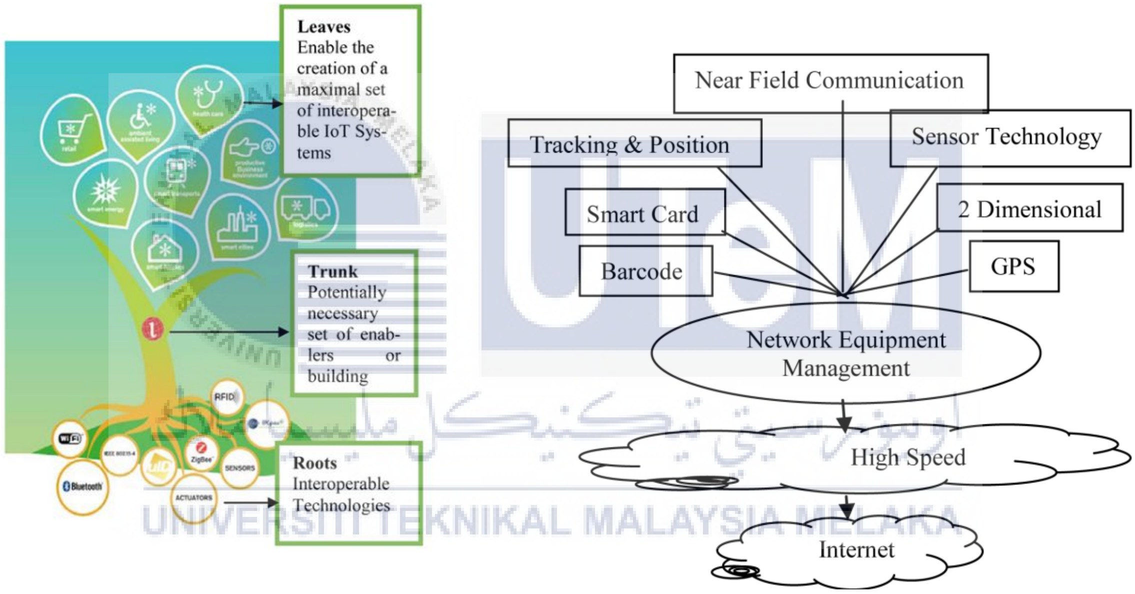


Figure 2.3 Internet of Things

The Internet of Things (IoT) creates a connected network where physical objects merge with digital technologies, enabling communication, data exchange, and smart functionality. It transforms everyday objects into intelligent devices through sensors and connectivity. IoT empowers devices to collect data, collaborate, and bring benefits to sectors like homes, healthcare, transportation, agriculture, and manufacturing. It enhances comfort, efficiency, and security in homes, enables remote patient monitoring and optimized transportation, and improves agricultural processes. However, privacy, security, and ethical concerns must be addressed. IoT holds the potential to revolutionize technology, industries, and our lives by seamlessly merging the physical and digital realms. All network equipment management such as GPS, 2 Dimensional, sensor technology, Near Field Communication (NFC), Tracking and Position, Smart Card and Barcode also can be used in IoT.[6]

2.5 Nodemcu ESP32 Working Principal & Applications

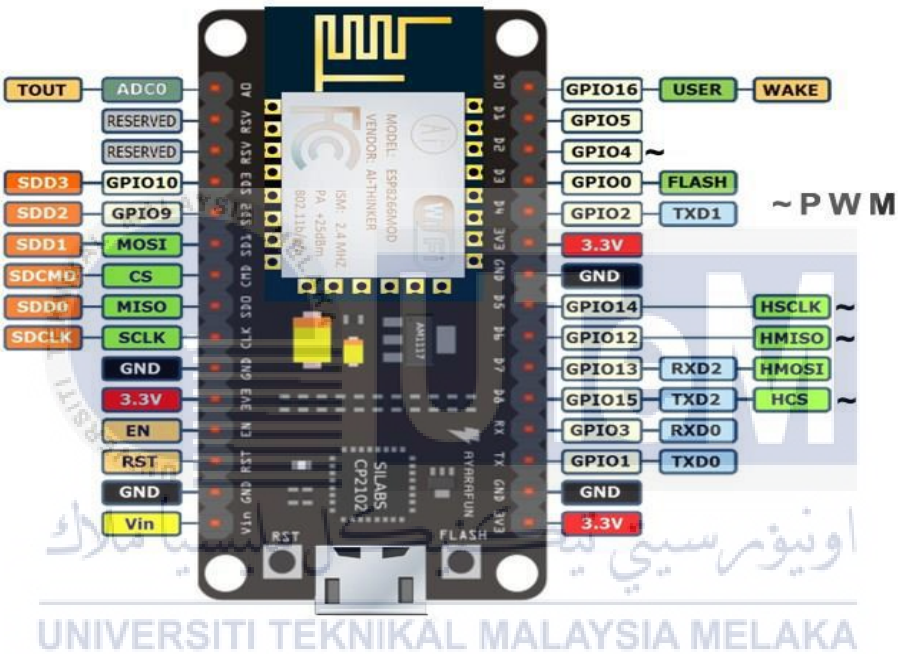


Figure 2.4 ESP32

Featuring the ESP32 chip, stands out as a highly integrated component tailored to meet the demands of our interconnected world. With its comprehensive and self-contained Wi-Fi networking solution, it excels in two key roles: acting as a standalone application host and offloading all Wi-Fi networking tasks from another application processor. One of the notable strengths of ESP32 lies in its impressive on-board processing power and storage capacity, facilitating seamless integration with various sensor-specific devices through its GPIOs. This characteristic minimizes the need for extensive upfront development efforts and reduces runtime overhead.[7]

The significance of this work extends to a multitude of applications and potential extensions. In the realm of Internet of Things (IoT), ESP32 empowers developers to effortlessly connect and control devices in a wireless network. This opens up avenues for smart home automation, where ESP32 can be used to monitor and manage lighting, temperature, security systems, and more. Furthermore, in industrial settings, ESP32 can enable efficient data collection from sensors, leading to enhanced monitoring, predictive maintenance, and process optimization.[8]

Additionally, ESP32's capabilities make it suitable for building robust and scalable IoT platforms, enabling cloud-based data processing, analytics, and visualization. This can be leveraged for environmental monitoring, asset tracking, and remote sensing applications. Moreover, ESP32 flexibility and ease of use encourage rapid prototyping, facilitating innovation in areas such as wearable technology, agricultural automation, and smart city initiatives. In essence, ESP32 provides a versatile platform that empowers developers and enthusiasts to explore and create a wide range of connected solutions, making significant strides towards a more interconnected and intelligent world.[9]

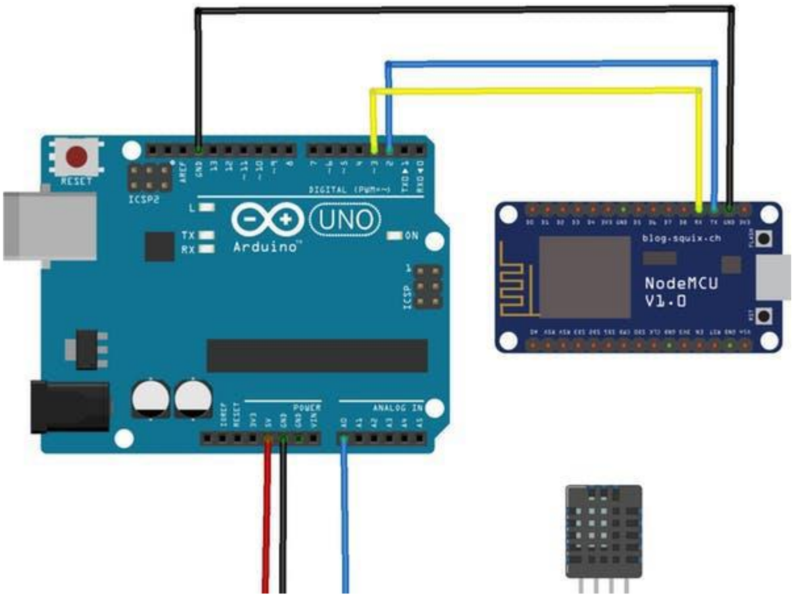


Figure 2.5 Serial Communication Between ESP32 and Arduino

Utilizing Arduino modules and microcontrollers has consistently proven to be a superb option for integrating automation into any given project. However, these modules possess a minor limitation in lacking built-in Wi-Fi capability, necessitating the addition of external Wi-Fi protocols to enable seamless compatibility with internet connectivity.. The Figure 2.5 above shows Serial Communication between ESP32 and Arduino. [10]

2.6 A Sample Project Distance Monitoring System with ESP32 for Industrial Automation Machines.

The presented system utilizing ESP32 is specifically designed for PLCs that lack internet connectivity options, providing owners with the capability to integrate a distance monitoring system seamlessly, without requiring any modifications. The research paper showcases a compact and user-friendly integrated system that boasts a remarkable 7 Mbps data transmission rate, ensuring high data security and real-time monitoring. Moreover, the project extensively explores the functionality and advantages offered by the ESP32, conducting a comprehensive comparison with alternative microcontrollers in the process. The comparison shows in figure below.

Table 1 Microchips for Monitoring Industrial Automation Systems

Microchips	Data Transmit Speed	Data Security	Processor Speed	Average Price
AVR	2.4Mb/s	-	50 MHz	$8.51
Raspberry Pi	55Mb/s to 178 Mb/s	WPA	1.2GHz	$35
ESP32	7Mb/s	WPA	80-160 MHz	$3

By referring to Table 1, it becomes evident that AVR microchips and Raspberry Pi possess commendable capabilities in constructing a novel monitoring system that incorporates sensors and actuators. However, ESP32 excels in fabricating a compact distance monitoring module that seamlessly integrates into existing machinery or systems, functioning as a diminutive gadget without necessitating any extensive reconstruction. This innovation results in the development of an efficient real-time distance monitoring system.

Table 2 Comparison Raspberry Pi, ESP32, ESP8266

Microchips	Number of GPIO'S	Communication	Data Security	Average Price
Raspberry Pi Zero W	26	802.11n wireless LAN and Bluetooth 4.0	WPA	$14
ESP32	15-16	802.11b/g/n wireless LAN and Bluetooth v4.2 BR/EDR	WPA	$4
ESP8266	15-17	802.11n b/g/n	WPA	$3

Upon consulting Table 2, it is apparent that three options exhibit a striking similarity. Raspberry Pi Zero W distinguishes itself by offering a greater quantity of GPIO pins compared to the others, albeit at a higher cost. Meanwhile, ESP32 and ESP8266 share numerous similarities in terms of GPIO pin count, communication capabilities, security features, as well as pricing. .[11]

2.7 A sample Project Vehicle Speed Detection System using IR Sensor.

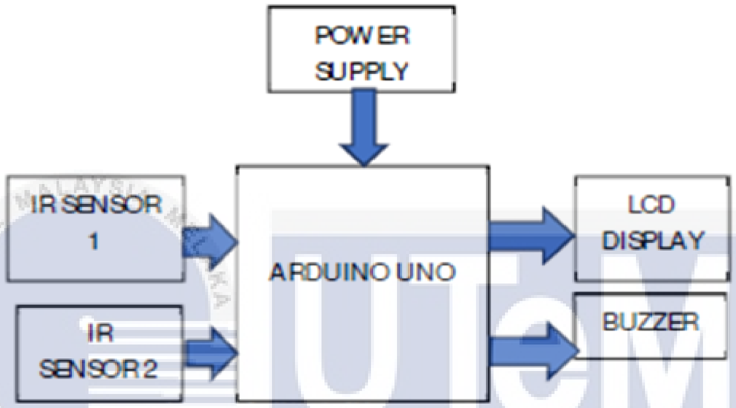


Figure 2.6 Block Diagram Vehicle Speed Detection

The primary objective of this system is to identify and track vehicles exceeding the speed limit by calculating their speed based on the time taken to traverse a fixed distance between two sensors. In the circuit design, the crucial component responsible for detecting the vehicle's speed is the implementation of IR Sensors. These sensors effectively gauge the duration it takes for the vehicle to pass through the specified distance by activating upon detection. Once the initial IR sensor is activated, a timer is initiated to accurately measure the time until the vehicle reaches the second IR sensor. Subsequently, the microcontroller engages in calculating the vehicle's speed in kilometers per hour (km/h), which is then displayed on a 16X2 LCD Module.

In the event that the vehicle's speed surpasses the predetermined speed limit, an integrated mechanism is triggered to alert the observer. This includes the activation of a buzzer and the blinking of an LED. Additionally, the LCD display will prominently exhibit the message "Over Speed Detected!!" to draw immediate attention to the violation. A comprehensive visualization of the system's architecture is presented in Figure 2.6, outlining the key components and their connections within the block diagram.[12]

2.8 The Effect of Two Different Weighted Badminton Rackets About Velocity and Torque When Outstanding Badminton Players Was Perfoming Smash Movement.

This research showcases the impact of varying racket weights on biomechanical variables during the smashing technique performed by elite players. Notably, when employing the heavier racket, the stability of the smash movement significantly increases to an impressive 92%. Conversely, when using a lighter racket, the stability measures only reach 88.7% during the identical action. With the shoulder joint serving as the pivotal point, the angular acceleration of the smash movement is considerably swifter when wielding the heavier racket. This substantial discrepancy in angular acceleration can be attributed to the closely matched variables of both moments of inertia. As a result, the heavier racket unquestionably demonstrates its superiority over the lighter counterpart in terms of torque.

Table 3 The Testing Result of the Velocity of Two Rackets

Weight Rackets	85g	100g
Velocity(m/s)	65.99 +- 7.48	69.42 +- 5.49
Stability	88.7%	92%

The data presented in Table 3 provides a comprehensive overview of the velocities attained by the two distinct rackets. It is evident that the velocity achieved with the heavier racket weighing 100g surpasses that of the lighter counterpart weighing 85g during the act of smashing. Furthermore, the T-test results indicate a statistically significant difference in velocities, with the T-values reaching a level of significance at 0.05. In terms of stability, it is determined by dividing the variable standard deviation by the velocity. Consequently, considering the stability of a smashing action, it becomes apparent that the heavier racket (92%) holds a clear advantage over the lighter alternative (88.7%).

Table 4 Kinematics Variables of Badminton Smash

Weight Rackets	85g	100g
Angular velocity(rad/s)	17 +- 4.48	19.4 +- 3.97
Angular acceleration(rad/s)	1032 +- 268.8	1164 +- 238.2
Moment of inertia(kg-m2)	0.37 +- 0.073	0.39 +- 0.078
Torque(kg-m2-rad/s)	381.84 +- 19.62	453.96 +- 18.58

The kinetics variables pertaining to the smashing motion of distinct badminton rackets with varying weights are presented in Table 4. Notably, the angular velocity observed in the weightier racket surpasses that of the other, alongside a similar trend observed in angular acceleration. By utilizing the torque formula, it becomes apparent that torque is proportional to the product of angular acceleration and the moment of inertia associated with the hand and racket. As both moment of inertia values are closely aligned, the heavier racket consequently achieves a higher velocity during the badminton smash action. .[13]

2.9 Simulation of a Badminton Racket

1. Badminton Strokes

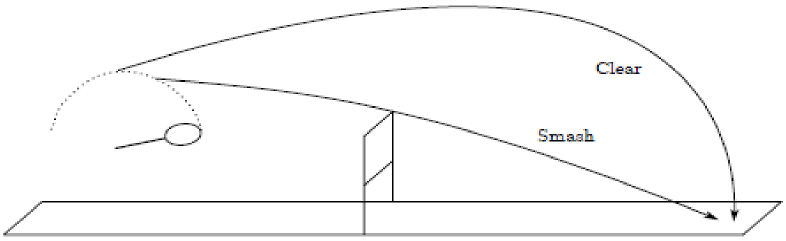


Figure 2.7 Badminton Strokes

In the realm of badminton, numerous distinct strokes serve specific purposes, with two of the most prevalent ones being the clear and the smash stroke, as illustrated in Figure 2.7. The clear stroke is commonly employed as a defensive maneuver, propelling the shuttle with a substantial upward velocity, causing it to arc over the court. The opposing player can only reach it as it descends near the end of the court, farthest from the net. While it is relatively easy for the opponent to hit the shuttle in this position, executing an aggressive counter-attack becomes arduous. Additionally, the player executing the clear stroke benefits from the extra time it takes for the shuttle to descend, allowing them to reposition themselves effectively. This combination of factors renders the clear stroke an excellent defensive shot. On the other hand, the smash stroke serves as an offensive move, involving striking the ball with considerable velocity at a steep angle. This tactic leaves the opponent with minimal time to react and return the shuttle. While there are various other types of badminton strokes, this study focuses on the aforementioned clear and smash strokes due to their frequent usage and the considerable power required from the racket to execute them effectively..[14]

2. **Racket Mass and Swing-Weight**

Through careful examination within a conducted study, it becomes evident that when comparing two rackets with distinct masses, specifically 100g and 85g, the heavier racket exhibits a swifter swinging motion. This observation suggests that weight reduction in rackets may yield advantageous outcomes, but only up to a certain threshold or limit..[15]

2.10 An Introduction to Firebase

Figure 2.8 Firebase Pillars

Firebase is a versatile web application platform that assists developers in building high-quality apps. It utilizes the JavaScript Object Notation (JSON) format for data storage, eliminating the need for complex queries when inserting, updating, deleting, or retrieving data. Serving as a backend system and database, Firebase offers several services. Firebase Analytics is a paid app measurement solution that provides valuable insights into app usage, user engagement, and behavior. Developers can capture pre-defined events and properties, as well as collect custom data. Firebase Cloud Messaging (FCM), formerly known as Google Cloud Messaging (GCM), is a paid cross-platform service that enables messaging and notifications for Android, Web Applications, and iOS. Firebase Auth supports social login providers such as Facebook, Google, GitHub, and Twitter. It offers client-side authentication capabilities and includes a user management system, allowing developers to enable email and password logins stored with Firebase.

Figure 2.9 Firebase Authentication

Firebase provides a real-time database and backend services. Developers can use the provided API to synchronize application data across clients and store it in Firebase's cloud. Client libraries facilitate seamless integration with Android, iOS, and JavaScript applications. Firebase Storage enables easy and secure file transfers for Firebase apps, regardless of network quality. Backed by Google Cloud Storage, it offers a cost-effective solution for storing images, audio, video, and other user-generated content. Firebase Test Lab for Android provides a cloud-based infrastructure for testing Android apps. With a single operation, developers can test their apps across a wide range of devices and configurations. Test results, including screenshots, videos, and logs, are conveniently available in the Firebase console. Test Lab can even automatically exercise the app, even if no test code has been written, to identify crashes. Firebase Crash Reporting generates detailed reports of app errors, grouping them into clusters based on similar stack traces and triaging them by severity. Additionally, developers can log custom events to capture the steps leading up to a crash.[16]

The storage of databases can be customized according to your needs, either by keeping them locally on your computer or by leveraging the capabilities of cloud storage solutions. Each application, be it on Android, iOS, or the web, possesses its unique database structure. When developing an Android application, developers have the flexibility to create databases utilizing various tools such as SQLite, shared preferences, websites, or even leverage cloud-based storage platforms[17]. The server infrastructure utilized for Android applications comprises Oracle SQL, Microsoft SQL Server, and MySQL, all interconnected with PHP files. However, a novel addition to this landscape is Firebase, an innovative platform specifically designed for Android apps, which employs JSON as its data storage format. Unlike the traditional servers, Firebase stores data in a JavaScript Object Notation (JSON) format, eliminating the need for query-based operations such as insertion, updating, deletion, or data addition. It serves as the backend system, functioning as a database for storing data. The diverse services and components of this infrastructure are depicted in Figure 2.8.[18]

2.11 MIT App Inventor

Figure 2.10 Graphical interface editor of MIT App Inventor

MIT App Inventor is an online platform aimed at teaching computational thinking through mobile app development. It enables students to create applications by dragging and dropping components in a design view and using a visual blocks language for programming. MIT App Inventor has two main editors: the design editor and the blocks editor. The design editor, also known as the designer, lets users drag and drop elements to create the app's user interface. On the other hand, the blocks editor provides a space for app inventors to visually arrange the logic of their apps. In this editor, users work with color-coded blocks that fit together like puzzle pieces, allowing them to easily describe the program's functionality.[19]

MIT App Inventor is a visual programming tool with a user-friendly interface similar to Scratch. However, it's specifically designed for creating fully functional mobile applications for phones and tablets. In App Inventor, users can visually define the app's graphical interface, making it easy for students to develop programs. What sets it apart is that students end up with a tangible product—a mobile app they can showcase, install on friends' and family members' phones, or even publish on Google Play. This focus on mobile app development and the creation of a real product serves as a strong motivator, often making MIT App Inventor a preferred choice over Scratch for introductory programming in undergraduate settings.[20]

2.12 Summary

In my project on Shuttlecock Speed Monitoring System, I have explored various components that are necessary to achieve the desired outcome. Through studying different sample projects, I have gained insights into different approaches for integrating these components effectively. Many of these projects utilize Arduino as the microcontroller, which is closely aligned with my own project's use of Arduino Uno as the microcontroller and IoT for connectivity. By examining these sample projects, I have learned how to seamlessly integrate Arduino Uno with other components such as infrared sensors, ESP8266, and more. This knowledge has equipped me with the ability to combine various sensors in the appropriate manner. Moreover, these sample projects have not only provided me with valuable ideas but have also instilled confidence in my ability to successfully complete my project, as I now have several references that can assist me along the way.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology section of a research project provides a detailed description of the methods and procedures employed to conduct the study. It outlines the steps taken to collect, analyze, and interpret data, as well as any experimental or observational techniques utilized. It also used is to help produce a creative and innovative project to achieve production objectives in this semester. The design of this project is development based on the suggestions as well as the views of project supervisor and agreement.

3.2 Methodology

The methodology encompasses the overall approach taken to design and construct the system, as well as the specific methods used for data collection and analysis. To begin, the development of the Shuttlecock Speed Monitoring System involved a systematic process that consisted of several key stages. These stages included requirements gathering, system design, component selection, prototyping, and final implementation. Each stage was carefully planned and executed to ensure the successful creation of a functional and efficient system.

Regarding data collection, our primary method involved the use of an infrared sensor positioned strategically within the system setup. This sensor was calibrated to detect the passage of the shuttlecock and measure its speed accurately. Additionally, we incorporated an Arduino Uno microcontroller to facilitate the sensor data acquisition and processing. The collected data was stored in a suitable format for further analysis.

To analyze the collected data, we employed various statistical techniques. These techniques included calculating the average speed, determining standard deviation, and assessing the consistency of shuttlecock speed across multiple trials. Furthermore, we performed a comparative analysis of the measured speeds against a reference standard to validate the accuracy of our system. In this section we going to see the process flow and equipment to build this project, including the testing process as well.

3.3 Elaboration Process Flow

3.3.1 Flowchart

Prior to doing a project or work, initial planning is essential to ensure that the planned project can be completed and can operate as best as possible. This is aimed at leveraging on the use of time and work done continuously according to the standards set. On the contrary, bad job planning will cause various problems when doing a job especially when the job requires full attention and focus. Among the problems arising from the irregular planning of work is that such projects are not completed at a set time, failing to meet standards, foggy and abandoned. It would be unpleasant for some to do the work done. For that reason, every project work is planned for the successful implementation of this project.

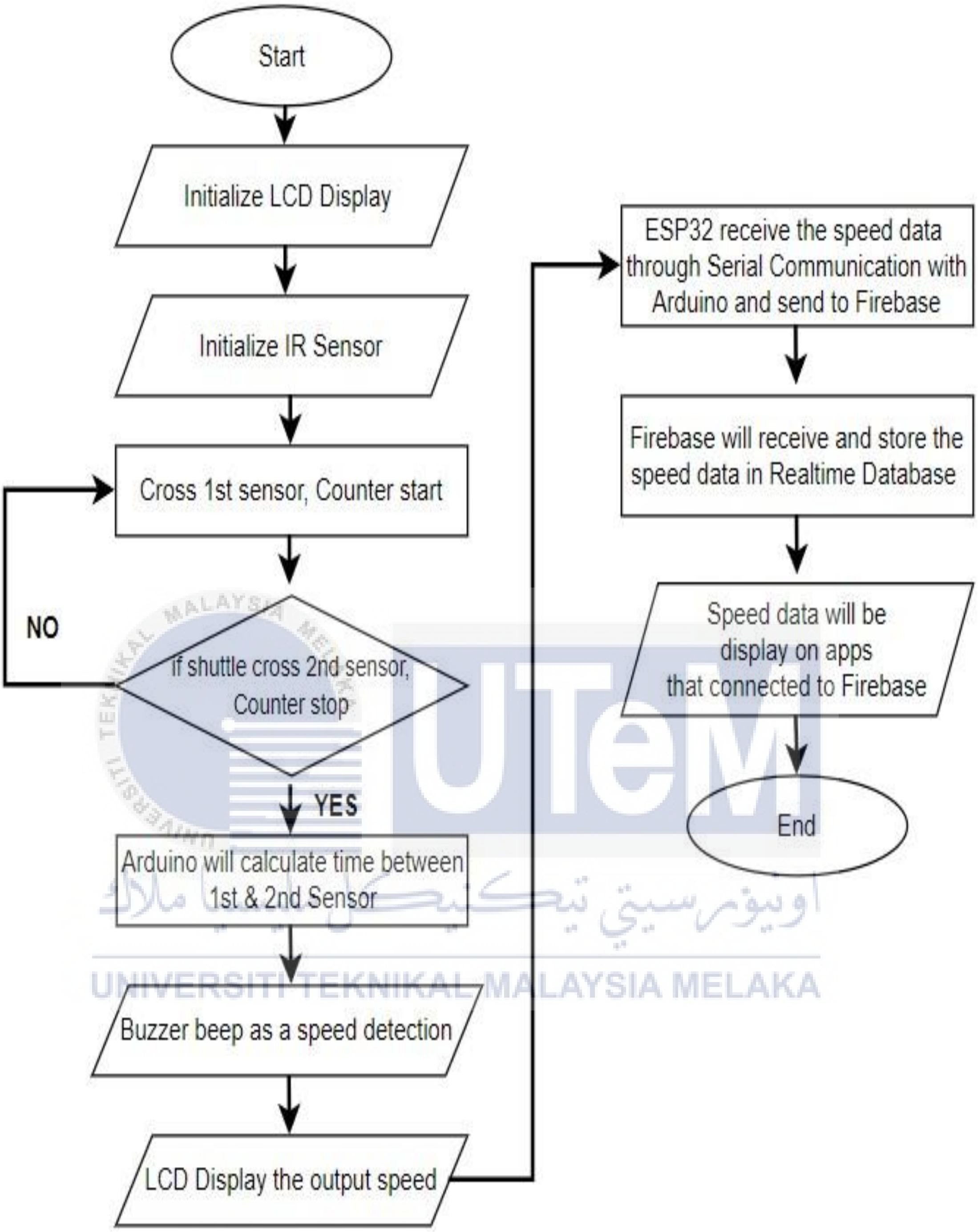


Figure 3.1 Flowchart Shuttlecock Speed Monitoring System

3.3.2 Block Diagram

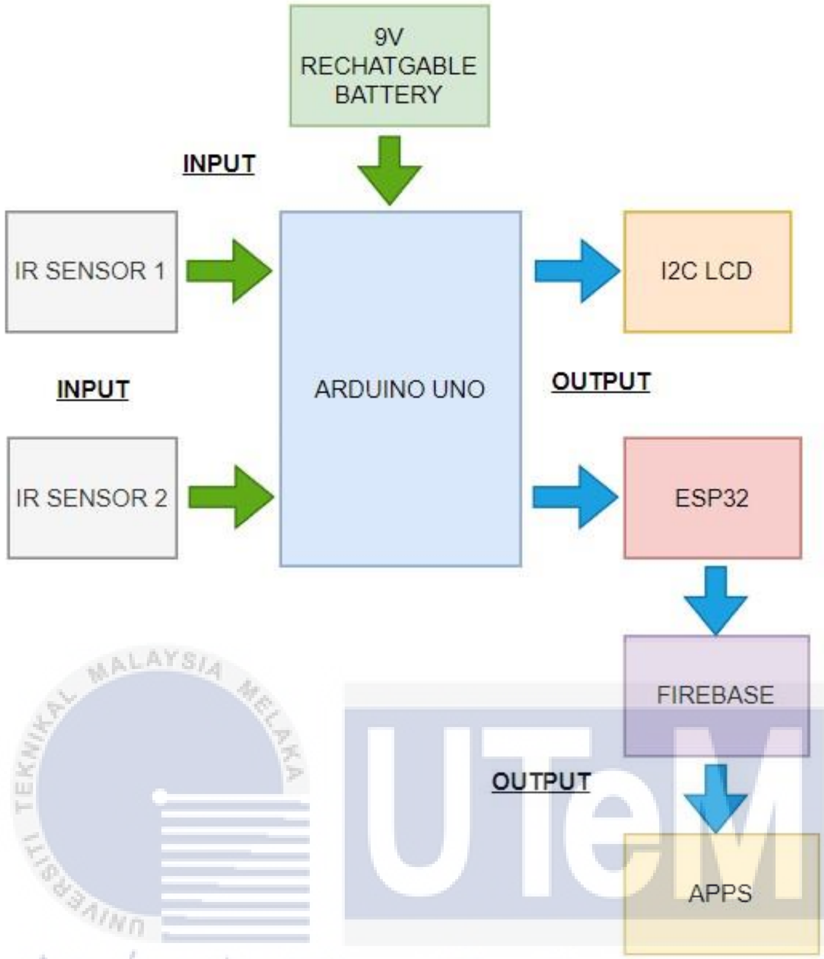


Figure 3.2 Block Diagram Shuttlecock Speed Monitoring System

Based on figure 3.2, The Shuttlecock Speed Monitoring System operates through the following procedure: Firstly, a power supply is required for the Arduino Uno board, which operates within a voltage range of 7 to 12 volts. In this system, a 9V rechargeable battery is utilized not only to power the Arduino Uno board but also to provide current to other components. The Arduino Uno R3 microcontroller plays a crucial role in the system as it is responsible for sensor integration, data collection, data processing, connectivity, actuator control, communication, and IoT prototyping. An IR sensor is employed to detect interruptions or reflections caused by the object in question. By measuring the time between these interruptions, the speed can be accurately calculated. Additionally, an I2C LCD module is incorporated, which offers various functions for controlling and manipulating the display. Furthermore, the ESP32 can be interfaced with the IR sensor. It is capable of collecting data from the IR sensor and transmitting it over Wi-Fi to either a cloud platform or a local server for further analysis and monitoring.

3.3.3 Calculation of Speed By Arduino Uno

```
void loop() {
  while (digitalRead(sensorA)); // Read IR SensorA
  while (digitalRead(sensorA) == 0);
  t1 = millis(); // Record Time1
  while (digitalRead(sensorB)); // Read IR Sensor B
  t2 = millis(); // Record Time2
  speed = t2 - t1;
  speed = speed / 1000; // Convert milliseconds to seconds
  speed = (1.0 / speed); // v = d / t
  speed = speed * 3600; // Multiply by seconds per hour
  speed = speed / 1000; // Divide by meters per kilometer
```

Figure 3.3 Coding of Calculation Speed by Arduino

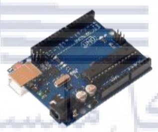
1. The **loop** function is the main function that runs repeatedly in an Arduino program.
2. The first **while** loop waits for the signal from sensorA to become HIGH (or non-zero). This means it waits until the sensorA detects an object.
3. The second **while** loop waits for the signal from sensorA to become LOW (or zero), essentially waiting for the object to leave the sensorA's detection range.
4. **t1** is then set to the current time using the **millis** function, which records the time when the object passes sensorA.
5. The third **while** loop waits for the signal from sensorB to become HIGH, detecting when the object reaches sensorB.
6. **t2** is then set to the current time using the **millis** function, recording the time when the object passes sensorB.

7. The time difference (**t2 - t1**) is calculated and stored in the variable **speed**. This difference represents the time it took for the object to travel between sensorA and sensorB.
8. The subsequent lines of code convert the time difference into speed in kilometers per hour. The calculations involve converting milliseconds to seconds, then applying the formula v = d / t to get the speed in meters per second, and finally converting it to kilometers per hour.

3.4 Equipment Requirements

3.4.1 Hardware Requirements

Table 5 List of Component

No.	Components	Description	Quantity
1.	Arduino Uno R3 	Arduino is an open-source hardware and software company, that designs and manufactures single-board microcontrollers for building digital devices.	1
2.	E18-D80NK IR Sensor 	The E18-D80NK adjustable IR Infrared Proximity Sensor Switch features an IR sensor that can measure distance between 3 to 80 cm.	2
3.	I2C LCD 	I2C LCD" refers to a type of liquid crystal display that can be connected to a microcontroller using the I2C communication protocol. The I2C protocol uses only two wires for communication. This makes it easy to connect the LCD to a microcontroller without using many pins	1

4.	ESP32 	An open-source software and hardware development environment built and contains crucial elements of a CPU, RAM, WiFi, and even a modern operating system and SDK.	1
5.	Battery 9V 	A 9V battery provides a constant voltage of 9 volts, which is suitable for powering many low-voltage electronic devices.	1

3.4.2 Software Requirment

For the software requirements of my project, I have chosen to utilize ISIS Proteus simulation and Arduino IDE. ISIS Proteus simulation is a user-friendly software tool that allows me to design circuit schematics and simulate their functionality. This software enables me to create and verify the circuit design before implementing it physically, ensuring the completeness and accuracy of the project. The combination of ISIS Proteus simulation and Arduino IDE allows for a streamlined workflow. Once I have successfully compiled the code in Arduino IDE, I can use ISIS Proteus simulation to visualize and test the functionality of the circuit. This integration ensures that the project is thoroughly tested and functioning as intended.

Figure 3.4 Arduino IDE

3.5 Bill of Materials (BOM)

Table 6 Bill of Materials (BOM)

NO	ITEM ID	DESCRIPTION	QUANTITY	COST/ITEM	COST
1.	Arduino UNO R3	Arduino UNO R3 Compatible with USB Cable, ATmega328 16U2	1	RM39.80	RM39.80
2.	E18-D80NK IR Sensor	Adjustable IR Infrared Range Finder Obstacles Avoid Distance Digital Sensor Module for Arduino	2	RM12.90	RM25.80
3.	Battery 9V	Rechargeable Battery 9V	1	RM25.00	RM25.00
4.	Jumper	Male to Female (FF) 40pcs Dupont Jumper Wire DIY Experiment Breadboard Rainbow Cable 40P 10cm 20cm 30cm for Arduino	1	RM4.00	RM4.00
5.	Breadboard	MB102 Solderless Large Breadboard 170 400 830 Holes Dupont Jumper Wire EE Components Arduino UNO Donut Board	1	RM5.00	RM5.00
6.	ESP32	ESP32 is a chip that provides Wi-Fi and (in some models) Bluetooth connectivity for embedded devices	1	RM32.90	RM32.90
7.	I2C LCD	I2C Address: 0x27 Backlight (White character on Blue background) Supply voltage: 5V Size: 82x35x18 mm Come with I2C interface	1	RM10.90	RM10.90
				TOTAL COST	**RM143.40**

3.6 Experimental / Study Design

3.6.1 Simulation Hardware

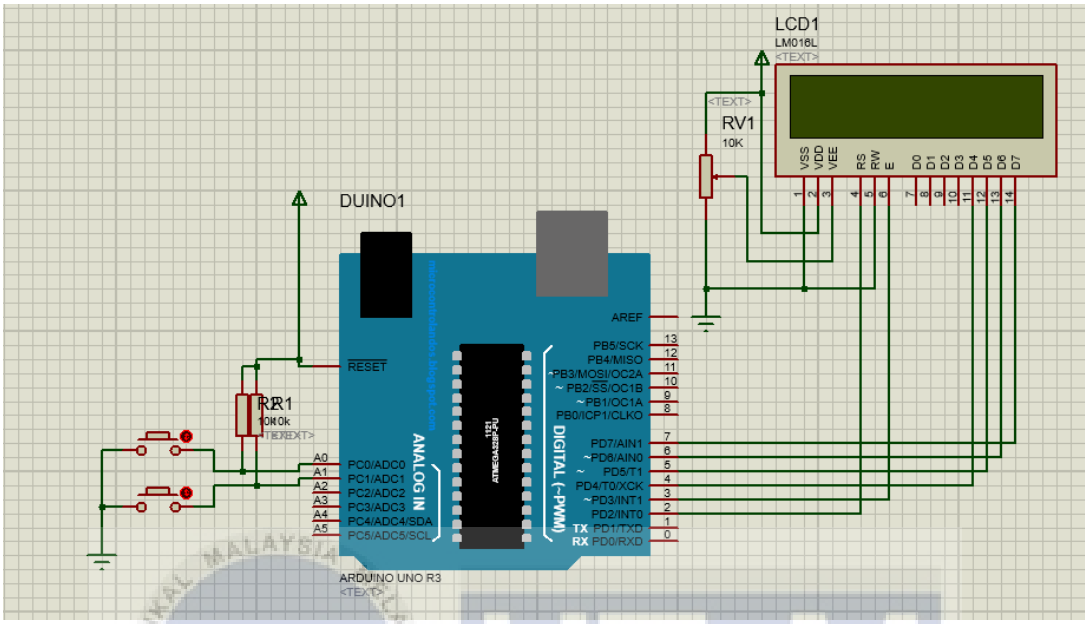


Figure 3.5 Circuit in Proteus

shuttle_simulation

```
void loop() {
if(digitalRead (ir_s1) == LOW && flag1==0){timer1 = millis(); flag1=1;}

if(digitalRead (ir_s2) == LOW && flag2==0){timer2 = millis(); flag2=1;}

if (flag1==1 && flag2==1){
     if(timer1 > timer2){Time = timer1 - timer2;}
else if(timer2 > timer1){Time = timer2 - timer1;}
 Time=Time/1000;//convert millisecond to second
 speed=(distance/Time);//v=d/t
 speed=speed*3600;//multiply by seconds per hr
 speed=speed/1000;//division by meters per Km
}

if(speed==0){
lcd.setCursor(0, 1);
if(flag1==0 && flag2==0){lcd.print("No Shuttle detected");}
                    else{lcd.print("Detecting...    ");}
}
else{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Speed:");
    lcd.print(speed,1);
    lcd.print("Km/Hr  ");
    lcd.setCursor(0, 1);
  if(speed > 50){lcd.print("  Impresive  ");}
            else{lcd.print("  Normal    "); }
    delay(3000);
    speed = 0;
    flag1 = 0;
    flag2 = 0;
 }
}
```

Figure 3.6 Coding Hardware in Arduino IDE

3.7 Testing Simulation

Based on coding that has been uploaded to Arduino. The two button will represent IR sensor. That button will give input 0 or 1. For 0 it will represent not detect and for 1 will represent detected. Then, The microcontroller will measure the time taken between the detection of the shuttlecock entering two predetermined points. Using this time and the known distance between the two points, it can calculate the speed of the shuttlecock. There are two conditions based on calculations, when the speed move over 50km/h , LCD will display “Impressive”. If not over 50km/h, it will display normal. Figure 3.7 and Figure 3.8 shows the display of the output.

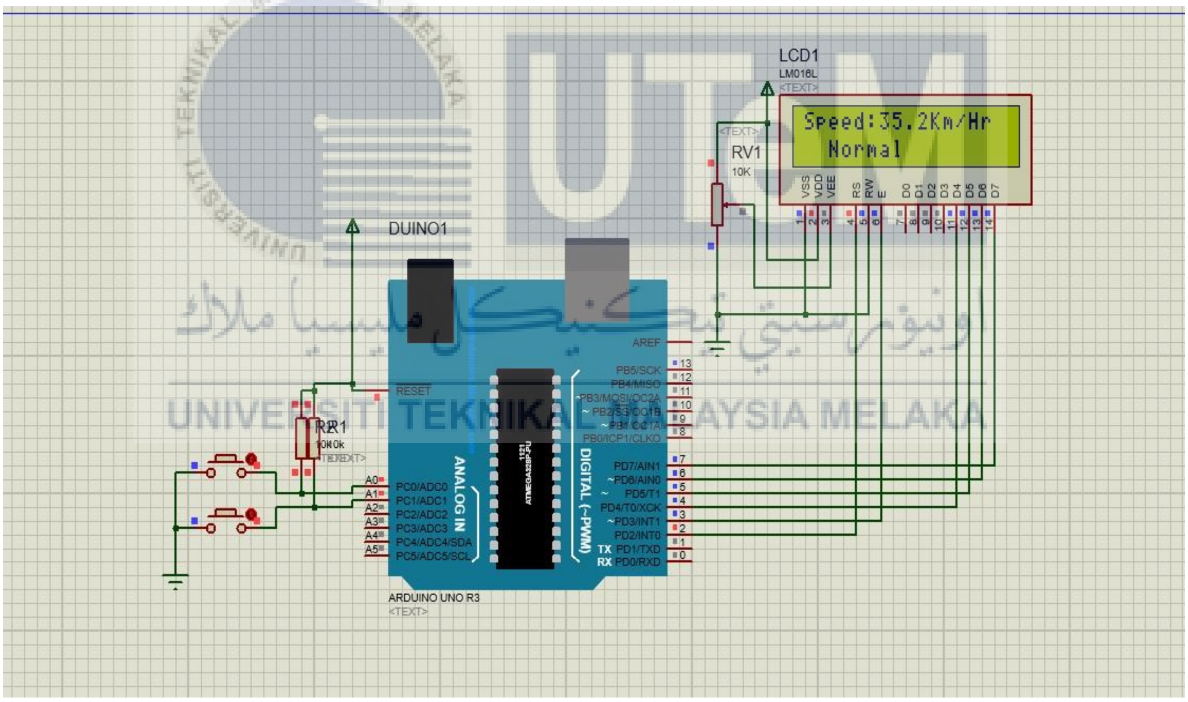


Figure 3.7 Not Over 50km/h

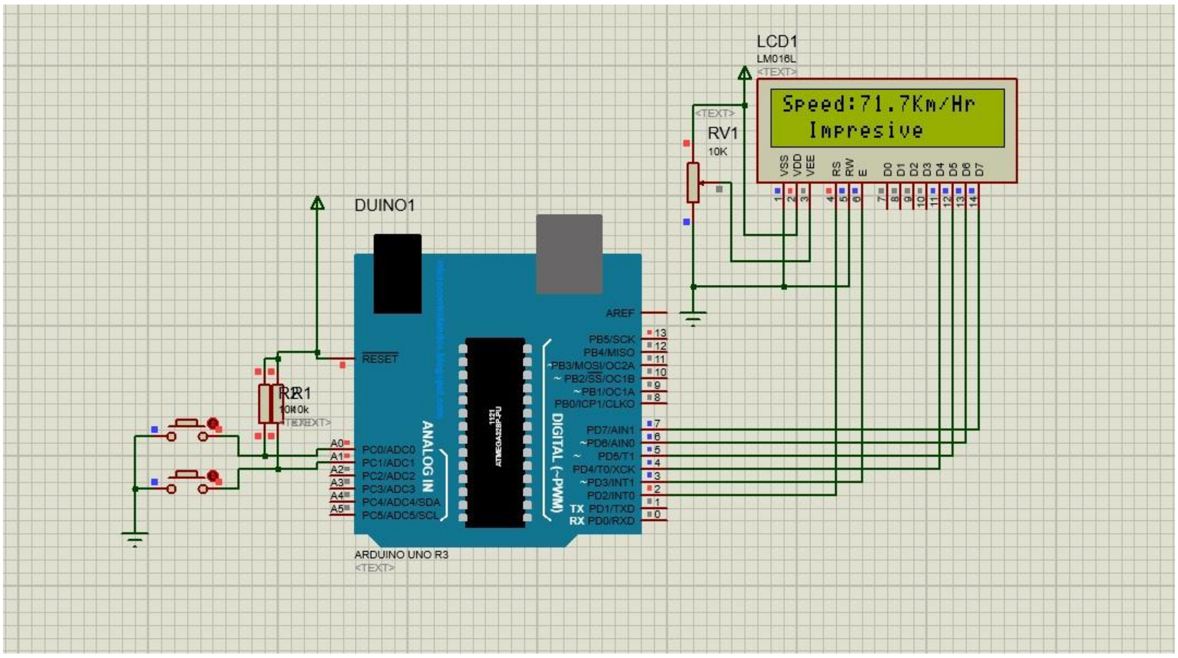


Figure 3.8 Over 50km/h

3.8 Coding Prototype

3.8.1 Serial Communication Between Arduino Uno and ESP32

This Arduino code uses two IR sensors to measure a badminton shuttlecock's speed. It waits for the shuttlecock to pass each sensor, then calculates its travel time and converts it to km/h. The speed is displayed on an LCD and sent to an ESP32 for further processing.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX
LiquidCrystal_I2C lcd(0x27, 16, 2);
int sensorA = 9; // Initialize Pin 3 for IR sensor A
int sensorB = 8; // initialize pin 2 for IR sensor B
int buzzerPin = 7; // Pin to which the buzzer is connected
unsigned long t1 = 0; // Define Time1 Variable
unsigned long t2 = 0; // Define Time2 Variable
float speed; // Define Speed variable
int i = 0;
```

```
void setup() {
  mySerial.begin(9600);
  lcd.begin(16, 2);
  lcd.backlight();
  lcd.setCursor(2, 0);
  lcd.print("Hi");
  lcd.setCursor(0, 1);
  delay(1000);
  lcd.clear();
  pinMode(sensorA, INPUT);
  pinMode(sensorB, INPUT);
  pinMode(buzzerPin, OUTPUT); // Set the buzzer pin as output
  lcd.setCursor(0, 0);
  lcd.print("Speed Measurement");
}

void loop() {
  while (digitalRead(sensorA)); // Read IR SensorA
  while (digitalRead(sensorA) == 0);
  t1 = millis(); // Record Time1
  while (digitalRead(sensorB)); // Read IR Sensor B
  t2 = millis(); // Record Time2
  speed = t2 - t1;
  speed = speed / 1000; // Convert milliseconds to seconds
  speed = (0.06 / speed); // v = d / t
  speed = speed * 3600; // Multiply by seconds per hour
  speed = speed / 1000; // Divide by meters per kilometer

  // Display speed on LCD
  lcd.setCursor(0, 0);
  lcd.print("Shuttle speed = ");
  lcd.setCursor(3, 1);
  lcd.print(speed);
  lcd.print(" Km/hr   ");

  // Send speed to ESP32 using Serial communication
  mySerial.print("Speed: ");
  mySerial.println(speed);

  // Display speed on serial monitor for debugging
  Serial.print("Speed: ");
  Serial.println(speed);

  // Sound the buzzer
  if (speed > 10.0) { // Adjust the threshold speed as needed
    digitalWrite(buzzerPin, HIGH);
    delay(100); // Adjust the delay as needed
    digitalWrite(buzzerPin, LOW);
  }
```

```
  delay(1000); // Adjust the delay as needed
}
```

3.8.2 Connection Between ESP32 to Firebase

This code grabs badminton shuttlecock speed data from another Arduino via serial, connects to Wi-Fi, and uploads the speed to your Firebase Realtime Database.

```
#include <SoftwareSerial.h>
#include <Arduino.h>
#if defined(ESP32)
  #include <WiFi.h>
#elif defined(ESP8266)
  #include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"

// Insert your network credentials
#define WIFI_SSID "No Connection"
#define WIFI_PASSWORD "airulshazli12"

// Insert Firebase project API Key
#define API_KEY "AIzaSyAxtHMQPGVYPh6-xuTy1b1HvV9l5AGZtL8"

// Insert RTDB URLefine the RTDB URL */
#define DATABASE_URL "https://speed-monitoring-df77a-default-rtdb.asia-
southeast1.firebasedatabase.app/"

//Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

SoftwareSerial mySerial(35, 34); // RX, TX, connect to Arduino
```

```
unsigned long sendDataPrevMillis = 0;
int count = 0;
bool signupOK = false;

void setup(){
  Serial.begin(9600); // Start the serial communication
  mySerial.begin(9600);

  //wifi connection
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  /* Assign the api key (required) */
  config.api_key = API_KEY;

  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;

  /* Sign up */
  if (Firebase.signUp(&config, &auth, "", "")){
    Serial.println("ok");
    signupOK = true;
  }
  else{
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }

  /* Assign the callback function for the long running token generation task */
  config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}

String prevDataSpeed = ""; // Variable to store the previous data received

void loop(){
  String data_speed = "";
  while (mySerial.available() > 0) {
    data_speed += char(mySerial.read());
  }
  data_speed.trim();
```

```
  Serial.println("Data Sensor Dari Arduino: " + data_speed);
  delay(2000);

  // Check if the received data is different from the previous data
  if (data_speed != prevDataSpeed) {
    prevDataSpeed = data_speed; // Update the previous data

    // Upload data to Firebase
    if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 15000 ||
sendDataPrevMillis == 0)){
      sendDataPrevMillis = millis();

      // Write a String to the database path Data/Speed
      if (Firebase.RTDB.setString(&fbdo, "Data/Speed", data_speed.c_str())) {
        Serial.println("PASSED");
        Serial.println("PATH: " + fbdo.dataPath());
        Serial.println("TYPE: " + fbdo.dataType());
      } else {
        Serial.println("FAILED");
        Serial.println("REASON: " + fbdo.errorReason());
      }
    }
  }
}
```

3.9 Summary

This chapter presented the process flow of Shuttlecock Speed Monitoring System flowchart and the block digram of the component how will there operate. We also show the equipment description of the hardware and the software that used for build the project. In this chapter we put the arduino code and demonstrate in simulation. I also use Proteus and Arduino IDE to simulation the hardware component. There also coding for prototype which divide for two part. First is coding for Arduino Uno that calculate data speed from IR sensors and sent the data speed to ESP32 by serial communication. After that the coding for ESP32 that recieved the data speed from Arduino Uno and send to Firebase Real-time Database to store the data for monitoring using apps.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the results and analysis on the development of The Shuttlecock Speed Monitoring System is a technological solution designed to measure and display the speed at which a shuttlecock travels during a badminton game. This system utilizes an IR (Infrared) sensor to detect the shuttlecock's presence as it passes through its field of view. By calculating the time taken for the shuttlecock to travel between two predetermined points and using the known distance between them, the system can accurately determine the speed of the shuttlecock. The purpose of this section is to present the results obtained from the Shuttlecock Speed Monitoring System and provide a comprehensive discussion of the findings.

4.2 Pre-Results

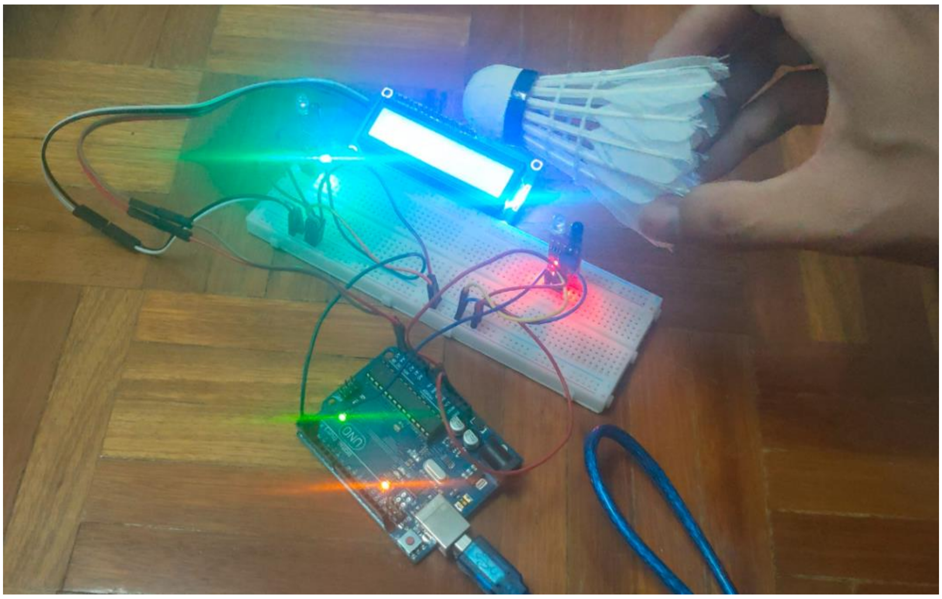


Figure 4.1 Shuttlecock Across First And Second Sensor

In Figure 4.1, when shuttlecock moving from first IR sensor (red) to second IR sensor (green), microcontroller will calculate the shuttlecock's speed based on the time measurements and the known distance between first and second IR sensors .

Figure 4.2 Display of the Speed

In Figure 4.2, after calculation done, you will see the speed of the shuttlecock moving across first and second IR sensor on LCD display is 13.08 km/h.

4.3 Results and Data Analysis

4.3.1 Analysis Hardware Prototype Only Without Iot Connection

This analysis is done in 2 sessions, which is normal swing and smash. Data will be recorded in the form of attempt, speed and detection. This hardware system only use Arduino uno to calculate the data speed shuttlecock from two IR sensors and display the value data to the LCD installed on the prototype.

4.3.1.1 Session 1 (Normal Swing)

Table 7 Data Of Normal Swing

ATTEMPT	SPEED KM/H	DETECTION
1	192.37	YES
2	219.98	YES
3	205.71	YES
4	381.19	YES
5	199.74	YES
6	220.12	YES
7	294.03	YES
8	300.61	YES
9	276.33	YES
10	239.68	YES

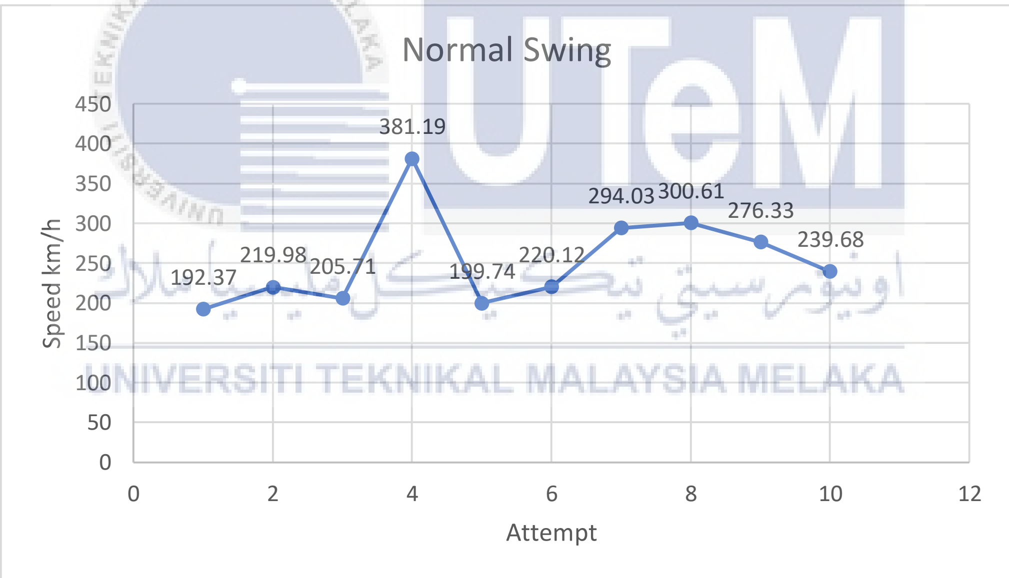


Figure 4.3 Graph Speed of Normal Swing

The data figure above shows the table of the first session which is the normal swing and the graph of the data. As referred to the data, the first session was performed with 10 normal swing attempts. It is shown that the sensor can detect all and the speed of the shuttle. Based on the graph it is seen that the speed range of the 10 attempts is between 190 km/h to 390 km/h.

4.3.1.2 Session 2 (Smash Swing)

Table 8 Data Of Smash Swing

ATTEMPT	SPEED KM/H	DETECTION
1	556.13	YES
2	0	NO
3	0	NO
4	769.91	YES
5	0	NO
6	598.04	YES
7	622.75	YES
8	0	NO
9	0	NO
10	733.31	YES

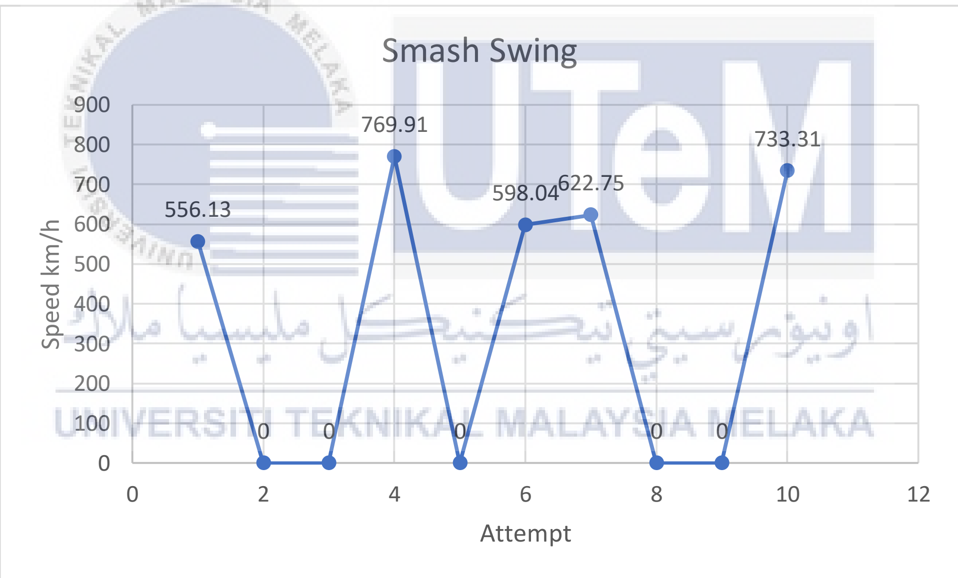


Figure 4.4 Graph Speed of Smash Swing

The data above shows the table of the second session which is the smash swing and the graph of the data. As referred to the data, the second session was also performed with 10 normal swing attempts. It is shown that the sensor cannot detect all the speed of the shuttle. Based on the table it is seen that there are 5 attempts that cannot be detected. Based on the graph it can be seen that the speed range of the detected 5 attempts is between 550 km/h to 800 km/h.

4.3.2 Analysis With Full System With Iot Connection

This analysis is done in 3 sessions, which is net play, drive and smash. Data will be recorded in the form of attempt, speed and detection. This system combined with Arduino Uno will get the data speed from IR sensors. Then Data will be sent to ESP32 through a serial connection. After that ESP32 will receive the data and then send the data to Firebase through Wi-Fi connection. The data will be stored in Firebase in a real-time database. Android application will get data from Firebase and display it in a smartphone for monitoring.

4.3.2.1 Session 1 (Net Play)

Net play in badminton refers to the actions and strategies employed by players near the net, particularly around the center of the court where the net is located. This area is crucial for both offensive and defensive purposes. Net play involves various techniques and skills aimed at gaining an advantage over the opponent.

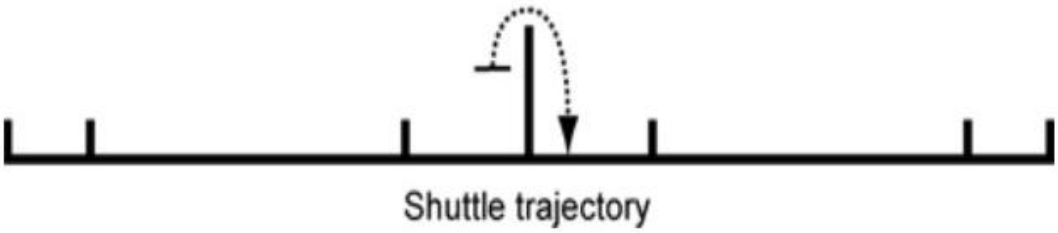


Figure 4.5 Net Play

Table 9 Net Play Data

ATTEMPT	SPEED KM/H	DETECTION
1	102.86	YES
2	133.33	YES
3	138.46	YES
4	90.00	YES
5	124.14	YES

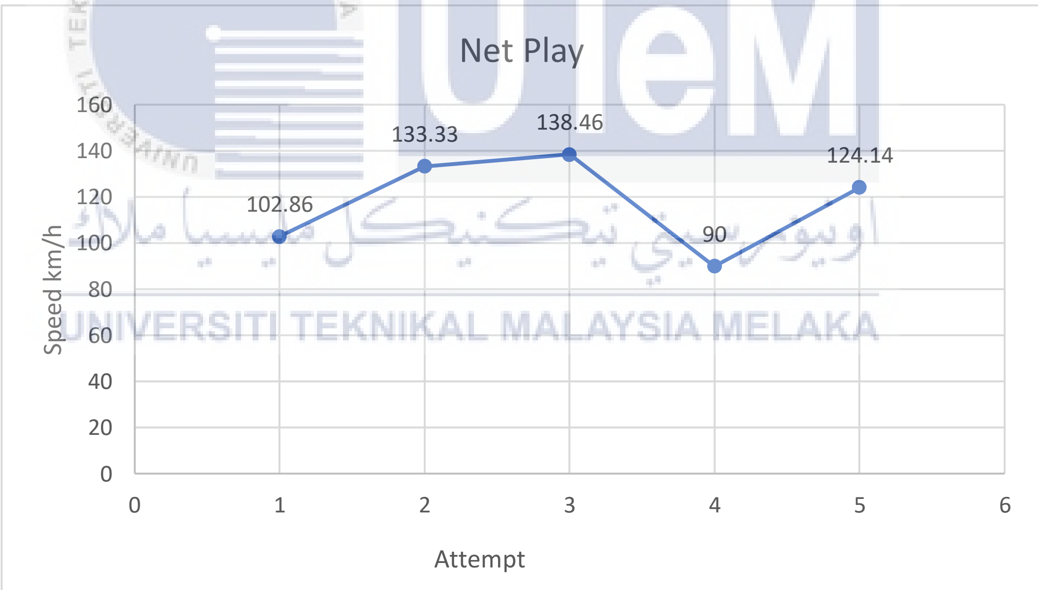


Figure 4.6 Net Play Speed Graph

The data in Figure 4.6 shows the table of the session 1 which is the Net Play and the graph of the data. As referred to the data, the first session was performed with 5 attempts. It is shown that the sensor can detect all and the speed of the shuttle. Based on the graph it is seen that the speed range of the 5 attempts is between 90 km/h to 140 km/h.

4.3.2.2 Session 2 (Drive)

In badminton, a drive refers to a fast, flat shot hit sharply and horizontally over the net, usually aimed at the opponent's body or into open spaces on the court. The drive is a quick and aggressive shot that is used both offensively and defensively.

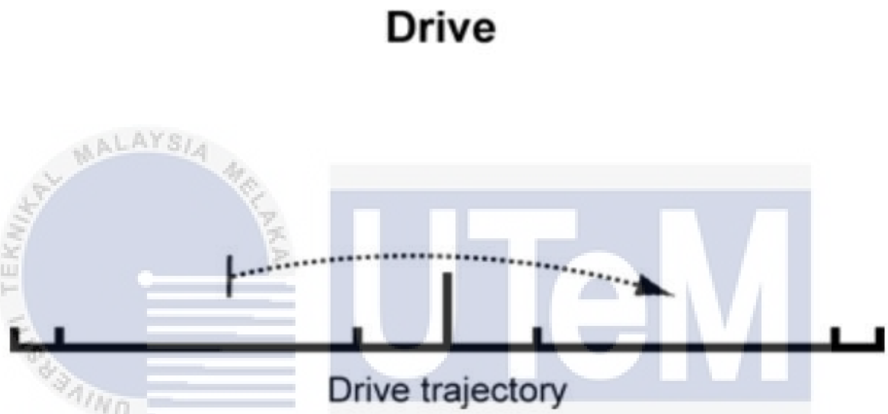


Figure 4.7 Drive

Table 10 Drive Data

ATTEMPT	SPEED KM/H	DETECTION
1	124.14	YES
2	276.92	YES
3	200.00	YES
4	144.00	YES
5	225.00	YES

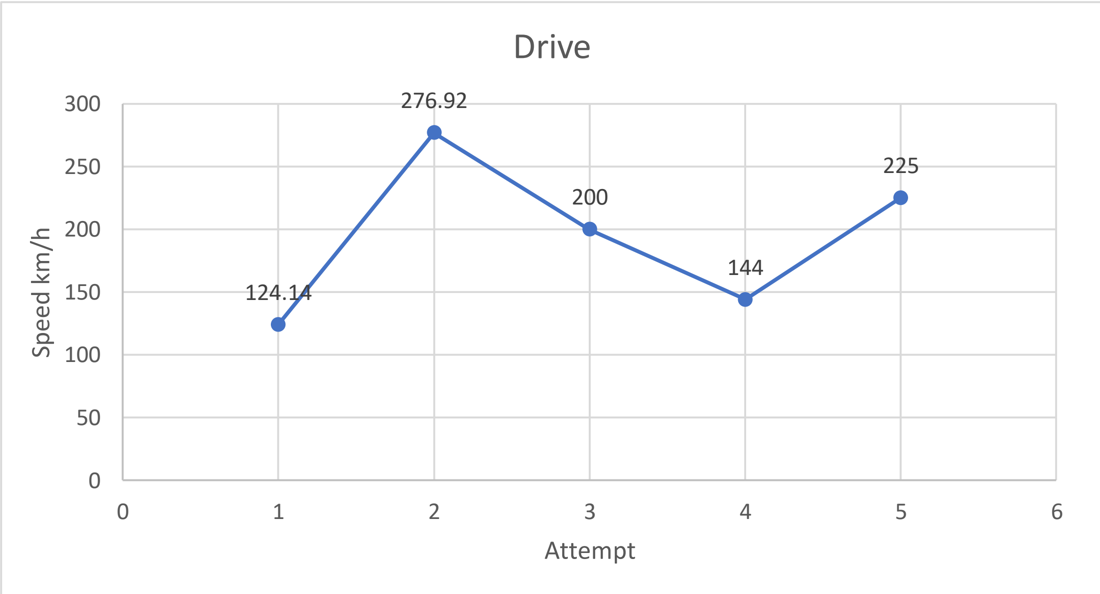


Figure 4.8 Drive Speed Graph

The data in Figure 4.8 shows the table of the session 2 which is the Drive and the graph of the data. As referred to the data, the first session was performed with 5 attempts. It is shown that the sensor can detect all and the speed of the shuttle. Based on the graph it is seen that the speed range of the 5 attempts is between 120 km/h to 280 km/h.

4.3.2.3 Session 3 (Smash)

A smash in badminton is a powerful and aggressive overhead shot where the player forcefully hits the shuttlecock downwards and steeply towards the opponent's side of the court. The aim of a smash is to generate high speed and steep angle, making it challenging for the opponent to return the shuttle effectively. The smash is one of the most potent attacking shots in badminton and is often used as a finishing move to score points.

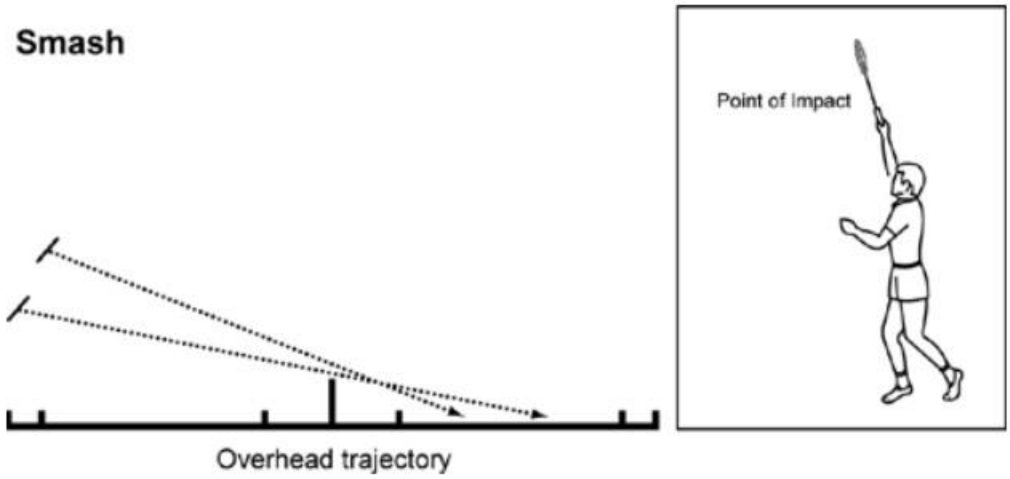


Figure 4.9 Smash

Table 11 Smash Data

ATTEMPT	SPEED KM/H	DETECTION
1	450.00	YES
2	400.00	YES
3	514.29	YES
4	327.27	YES
5	450.00	YES

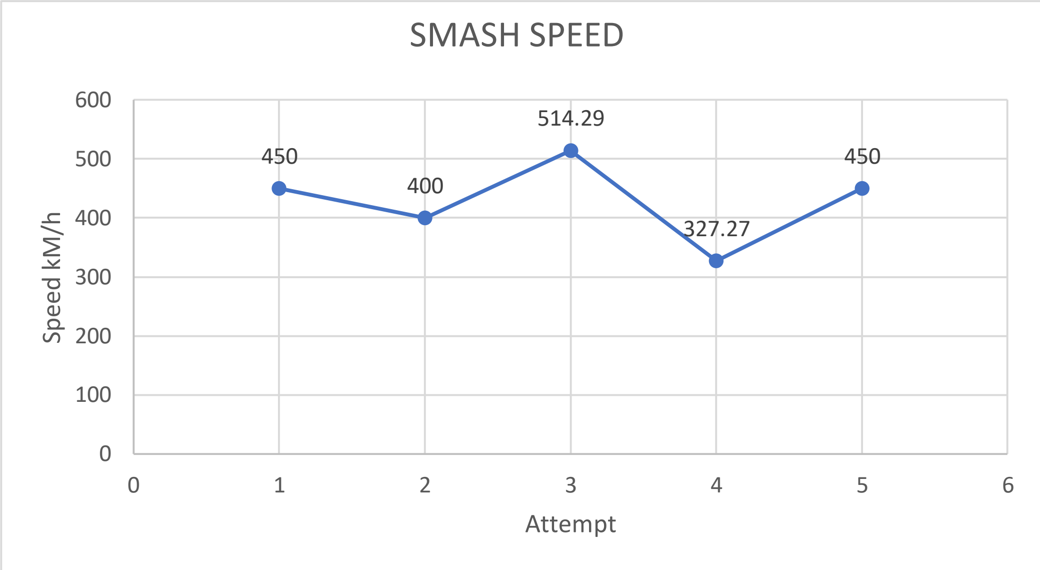


Figure 4.10 Smash Speed Graph

The data in Figure 4.10 shows the table of the session 3 which is the Net Play and the graph of the data. As referred to the data, the first session was performed with 5 attempts. It is shown that the sensor can detect all and the speed of the shuttle. Based on the graph it is seen that the speed range of the 5 attempts is between 325 km/h to 550 km/h.

4.3.2.4 Net Play vs Drive vs Smash graph

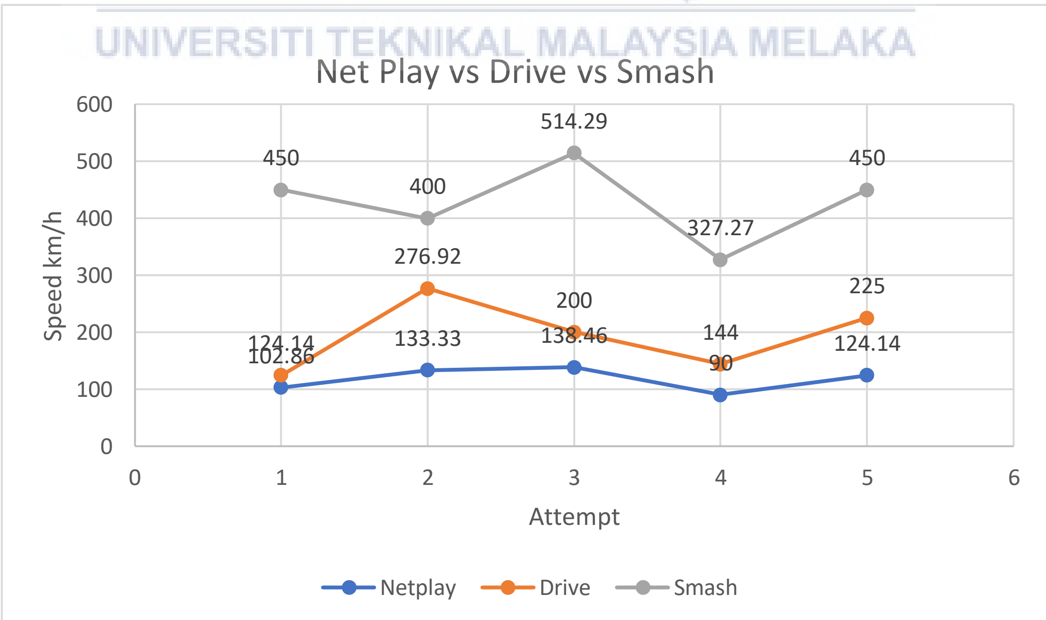


Figure 4.11 Net Play vs Drive vs Smash Graph

In Figure 4.11, smashes reign supreme as the fastest stroke, starting at a blistering 450 km/h but gradually slowing down, while net play and drive steadily pick up speed from around 130 km/h to reach roughly similar peak speeds by the fifth attempt. Overall, players seem to optimize their speed within a handful of tries and maintain consistency for net play and drive, while demanding smashes show quicker fatigue.

4.3.3 Monitoring Data From Firebase and Android Application Based on 3 Session

4.3.3.1 Session 1 (Net Play)

Figure 4.12 Speed 102.86 km/h

Figure 4.13 Speed 133.33 km/h

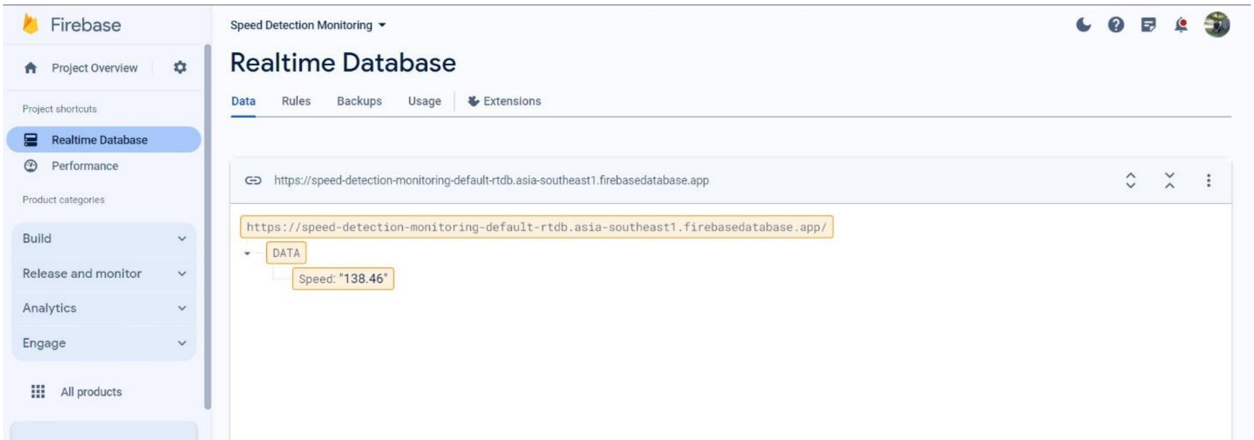


Figure 4.14 Speed 138.46 km/h

Figure 4.15 Speed 90.00 km/h

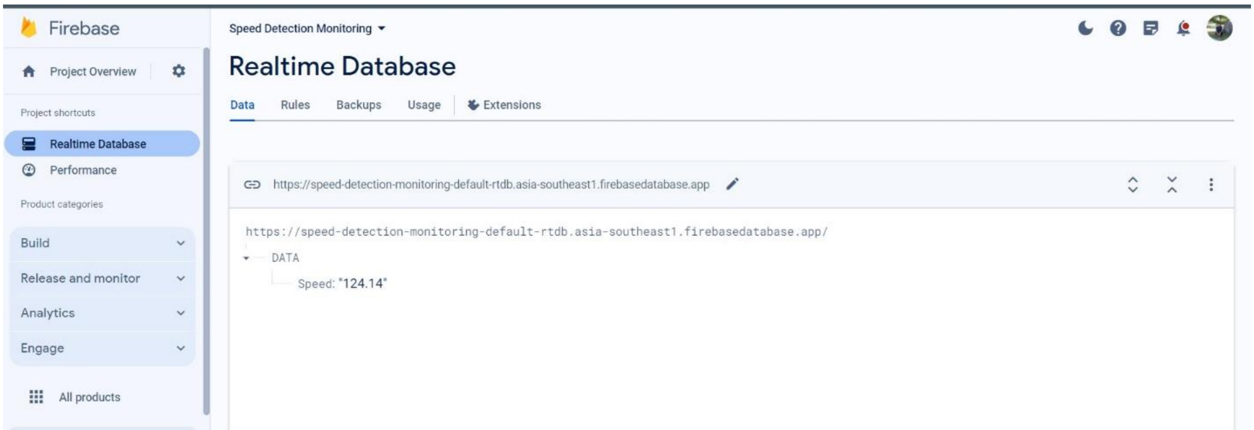


Figure 4.16 Speed 124.14 km/h

Display On Android Application

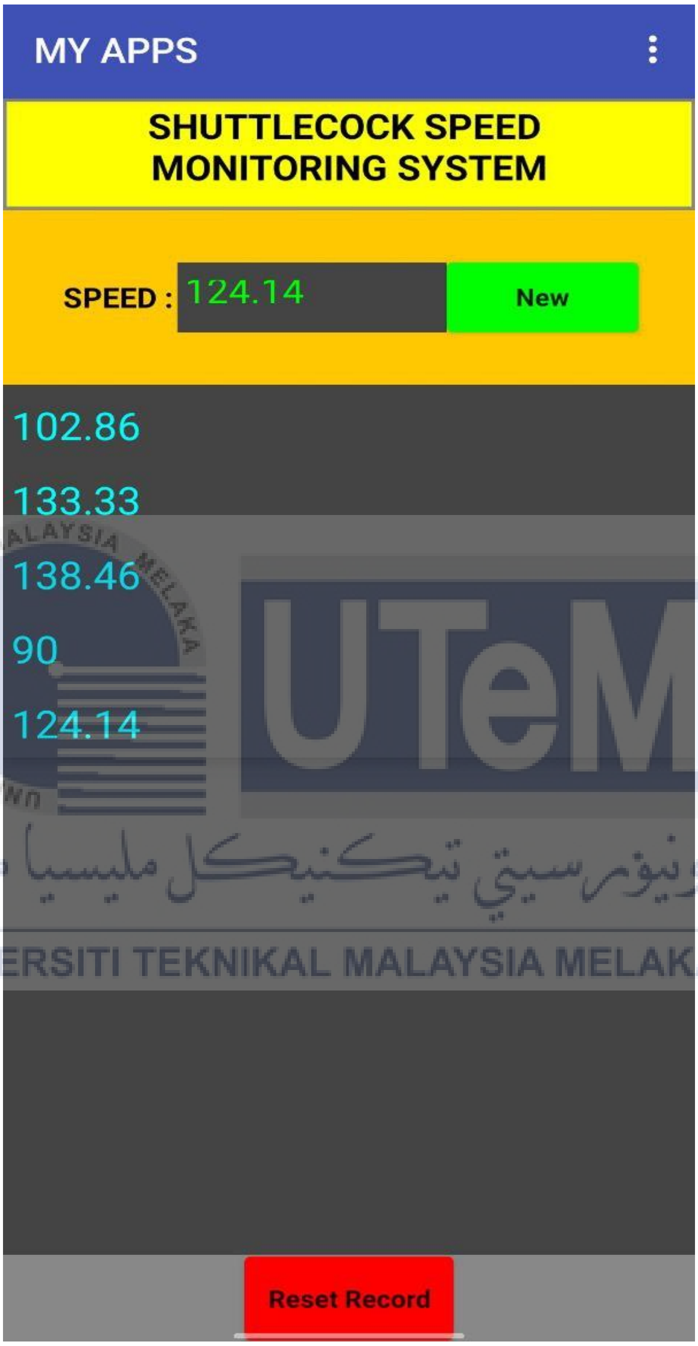


Figure 4.17 Application Display for Net Play

4.3.3.2 Session 2 (Drive)

Figure 4.18 Speed 124.14 km/h

Figure 4.19 Speed 276.92 km/h

Figure 4.20 Speed 200.00 km/h

Figure 4.21 Speed 144.00 km/h

Figure 4.22 Speed 225.00 km/h

Display On Android Application

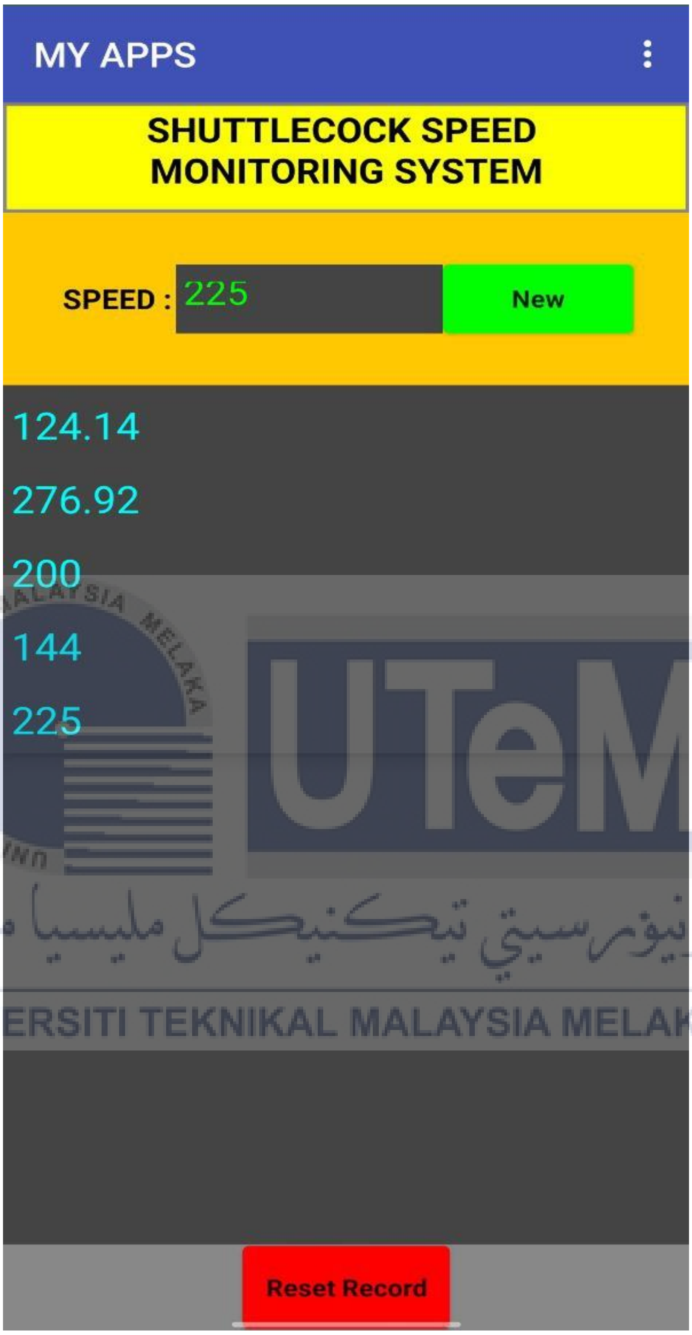


Figure 4.23 Application Display for Drive

4.3.3.3 Session 3 (Smash)

Figure 4.24 Speed 450.00 km/h

Figure 4.25 Speed 400.00 km/h

Figure 4.26 Speed 514.29 km/h

Figure 4.27 Speed 327.27 km/h

Figure 4.28 Speed 450.00 km/h

Display On Android Application

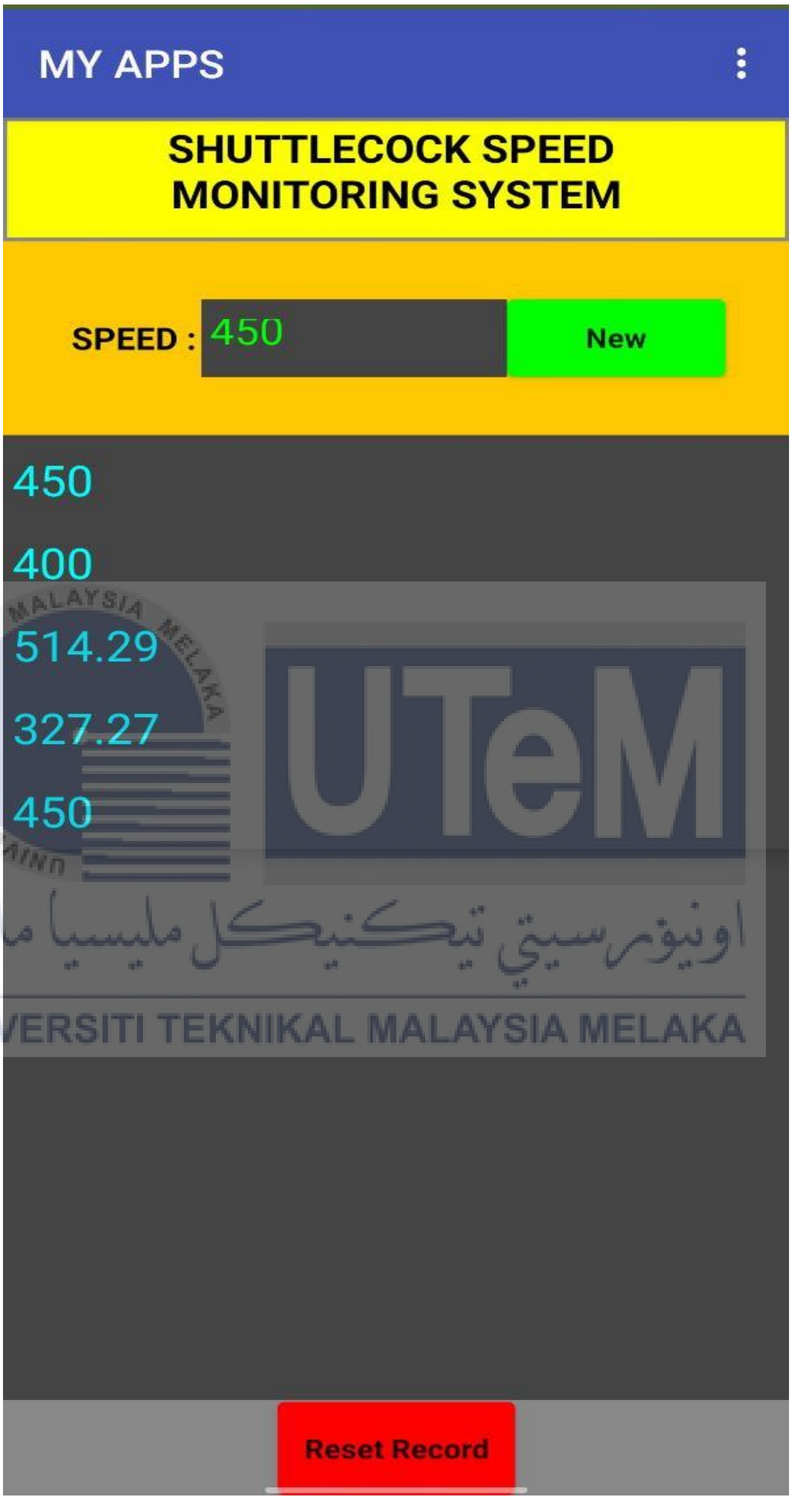


Figure 4.29 Application Display for Smash

4.4 Prototype Design

4.4.1 Skecth Design

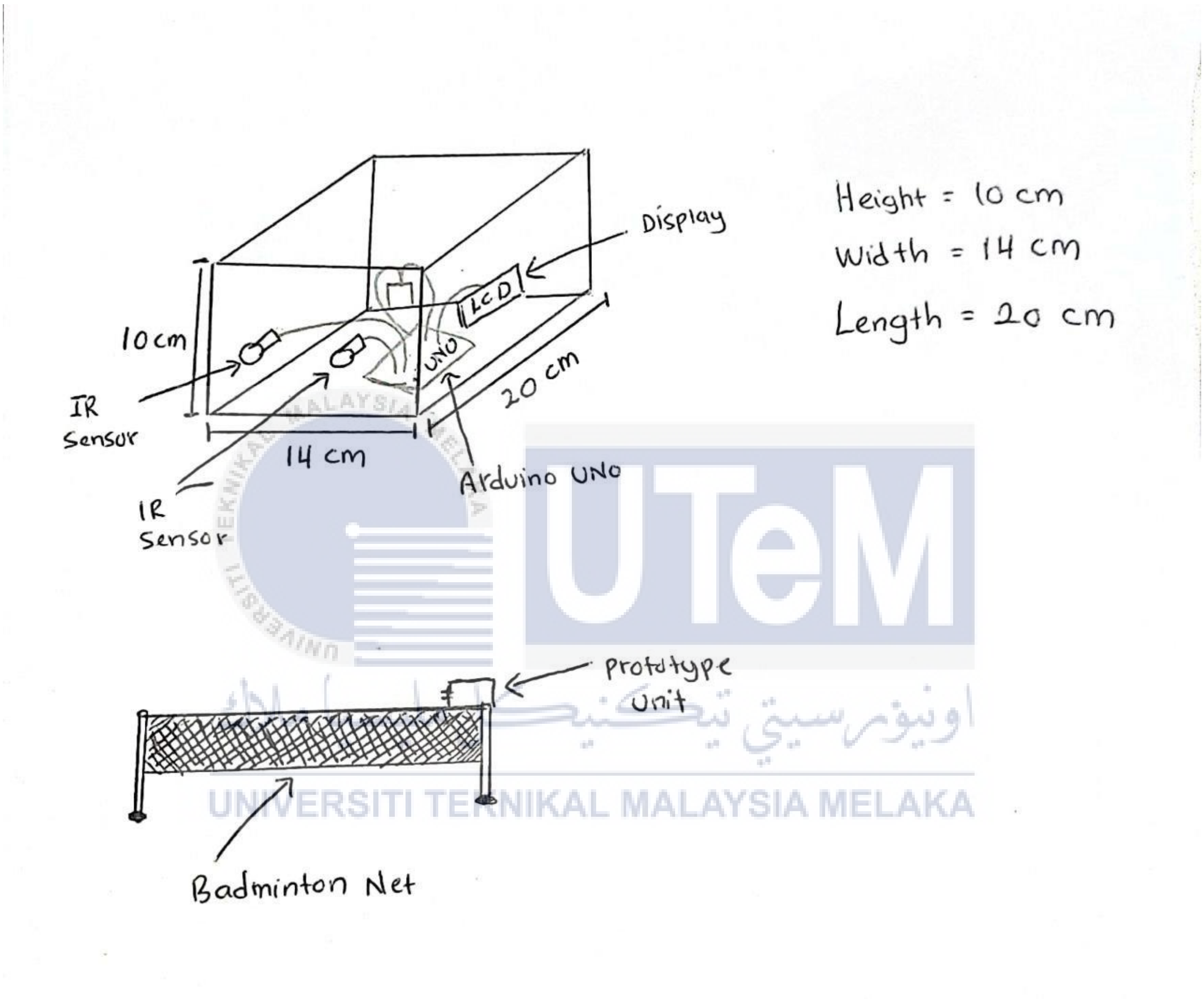


Figure 4.30 Prototype Design

4.4.2 Simulation Design Using Thinkercad

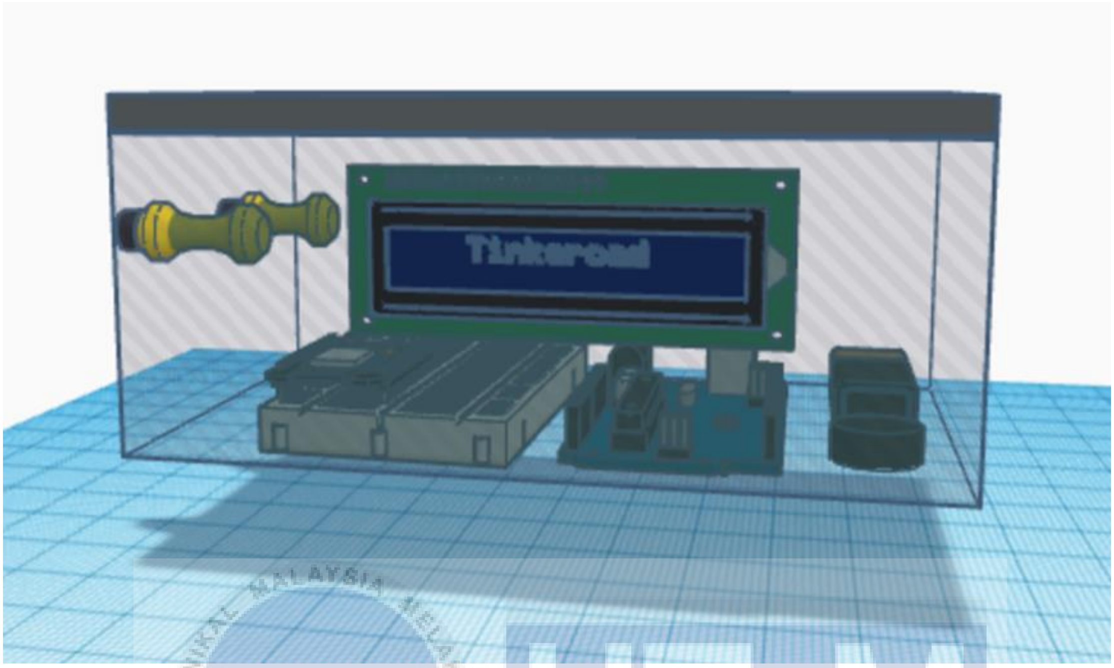


Figure 4.31 Prototype View

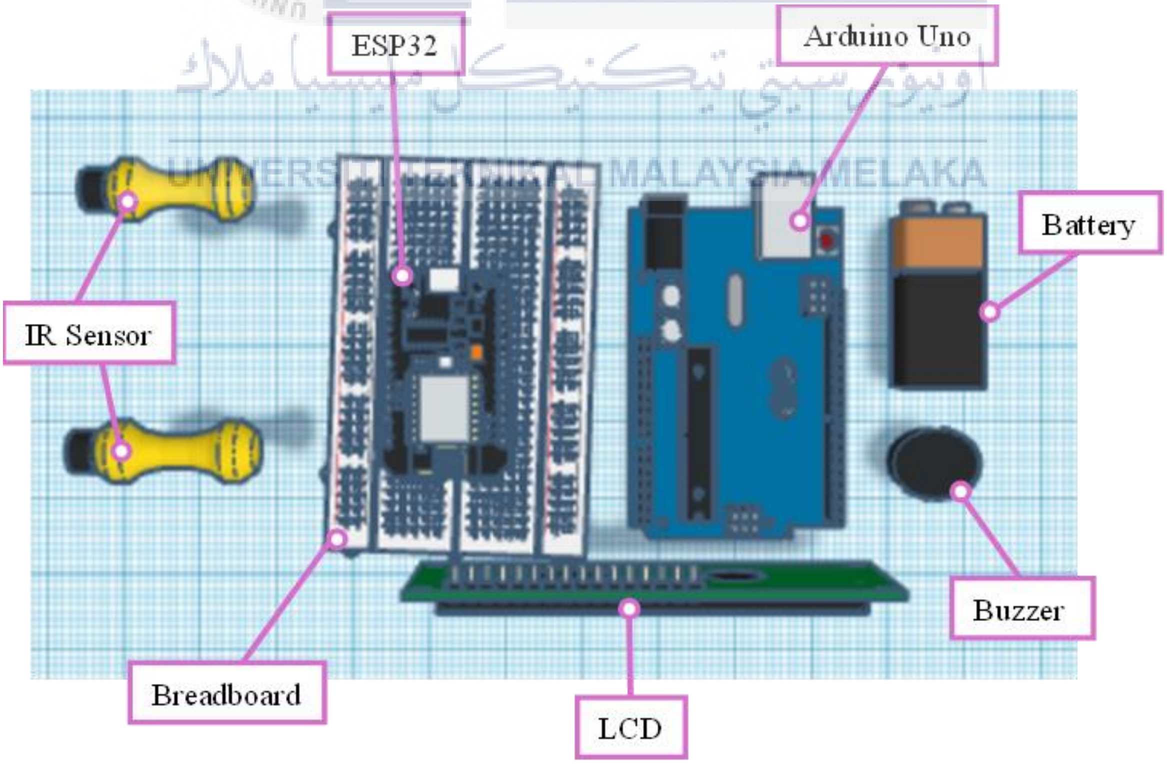


Figure 4.32 Component Inside Prototype

4.4.3 Prototype

Figure 4.33 Angle View

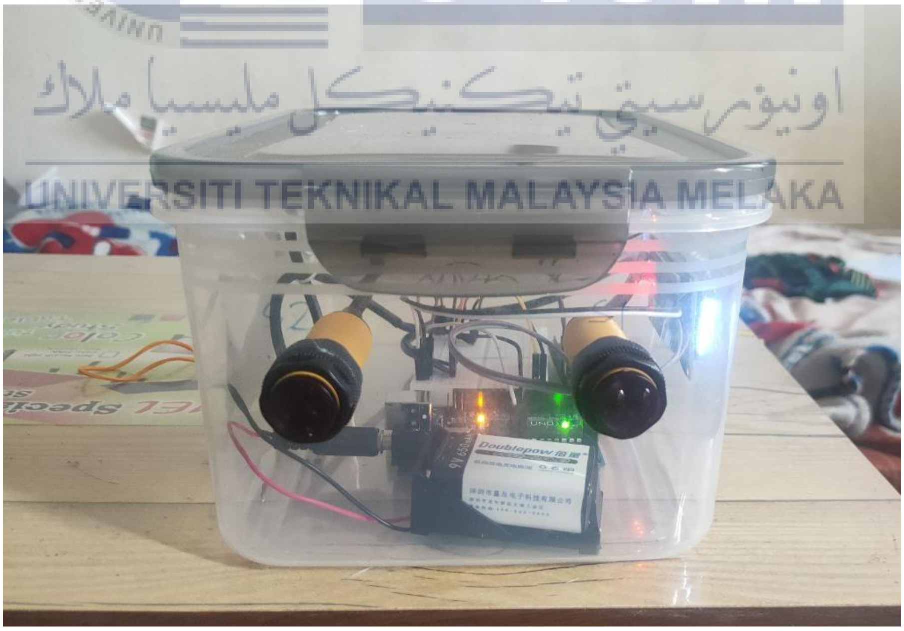


Figure 4.34 Front View

Figure 4.35 Side View

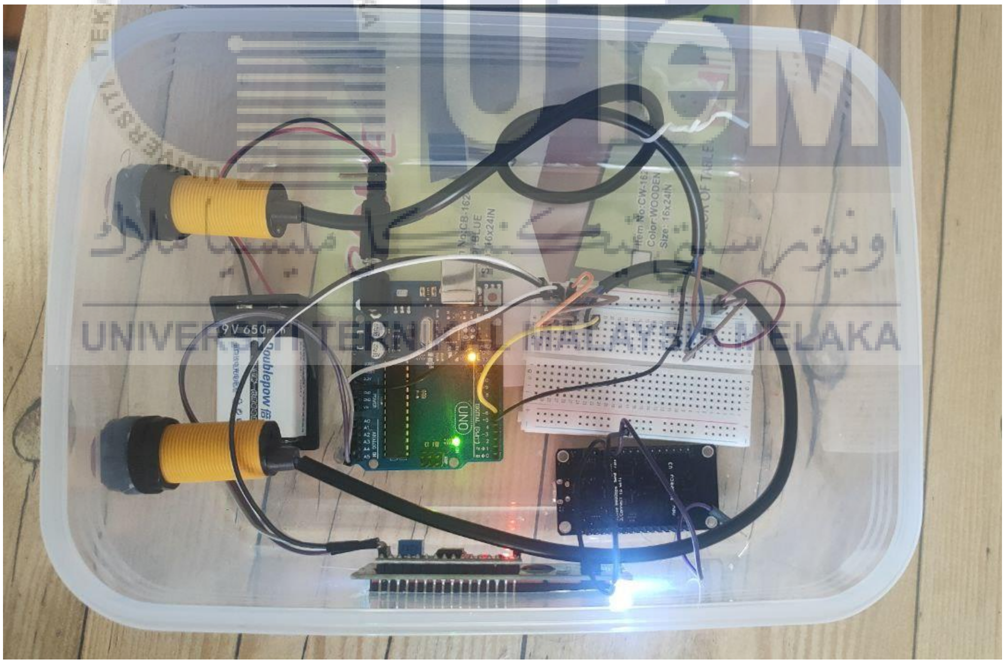


Figure 4.36 Upside and Inside View

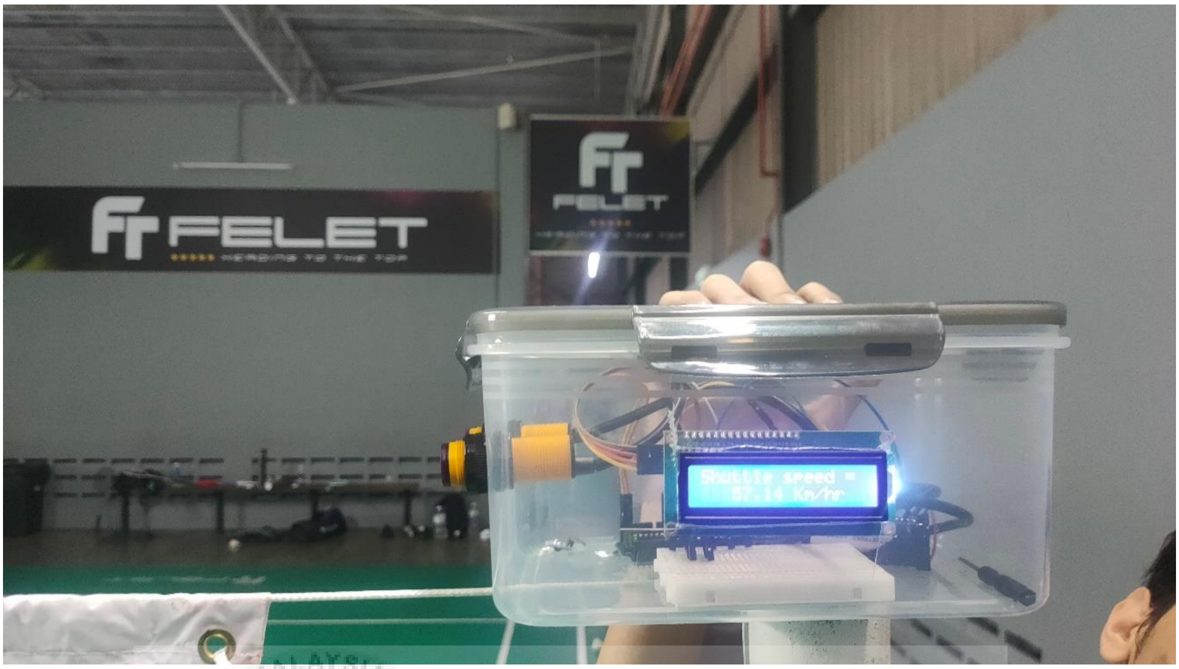


Figure 4.37 Position on Court

Figure 4.38 Position Near Net

4.5 Summary

The results obtained from the Shuttlecock Speed Monitoring System revealed accurate and real-time measurements of shuttlecock speed during badminton training. The data analysis from testing the prototype based on various type of session. The speed data then will be convert to graph for analysis. The system successfully detected the shuttlecock's presence using the IR sensor and calculated the time taken for it to travel between two specific points. These measurements were then used to calculate the shuttlecock's speed, which was displayed on a visual display unit. During the testing phase, the Shuttlecock Speed Monitoring System was used to capture and record the speed of the shuttlecocks. The system displayed the speed in units of kilometers per hour (km/h), providing immediate feedback to players in training.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The development and implementation of the Shuttlecock Speed Monitoring System using IoT (Internet of Things) technology have resulted in a highly advanced and versatile solution for measuring and analyzing the speed of shuttlecocks during badminton games. The integration of IoT capabilities has enhanced the system's functionality, connectivity, and data management, offering valuable insights for players, coaches, and spectators. The IoT-enabled Shuttlecock Speed Monitoring System successfully captures speed measurements using the IR sensor and transmits the data wirelessly to a central hub or cloud-based platform for further analysis and visualization. The system allows for real-time monitoring of shuttlecock speed, enabling immediate feedback and performance evaluation.

5.2 Future Works

Future work for the Shuttlecock Speed Monitoring System using IoT can encompass several areas of improvement and expansion. Here are some potential avenues for further development:

1. **Multi-Sensor Integration**: Incorporate additional sensors or technologies to gather more comprehensive data about shuttlecock movement and performance. For example, integrating accelerometers or gyroscopes can provide insights into shuttlecock spin, trajectory, and stability during flight. This would enhance the

overall analysis of shuttlecock behavior and offer a more holistic understanding of gameplay dynamics.

2. **Data Integration with External Systems**: Explore opportunities for integrating the Shuttlecock Speed Monitoring System with external systems or platforms. For instance, integrating with sports analytics software or badminton coaching apps can provide a seamless experience for players, allowing them to incorporate speed data into a broader performance analysis framework.
3. **Integration with Wearable Technology**: Explore integration possibilities with wearable devices such as smartwatches or fitness trackers. This would enable players to monitor their shuttlecock speed and receive real-time feedback directly on their wearable devices, promoting convenience and ease of use.
4. **Collaboration with Badminton Associations and Professionals**: Collaborate with badminton associations, professional players, and coaches to gather valuable insights and feedback for system improvement. Engaging with the badminton community will help ensure that the Shuttlecock Speed Monitoring System meets the specific needs and requirements of players and coaches.
5. **Performance Comparison and Benchmarking**: Develop features that allow players to compare their shuttlecock speed performance against their own historical data or against other players' data. This can facilitate healthy competition, skill improvement, and motivation among players.

REFERENCES

[1] L. Louis, “Working Principle of Arduino and Using it as a Tool for Study and Research,” *International Journal of Control, Automation, Communication and Systems*, vol. 1, no. 2, pp. 21–29, Apr. 2016, doi: 10.5121/ijcacs.2016.1203.

[2] R. H. Sudhan, M. G. Kumar, A. U. Prakash, S. A. R. Devi, and S. P., “ARDUINO ATMEGA-328 MICROCONTROLLER,” *IJIREEICE*, vol. 3, no. 4, pp. 27–29, Apr. 2015, doi: 10.17148/ijireeice.2015.3406.

[3] S. Monk, “Programming Arduino® Getting Started with Sketches,” 2023.

[4] B. Shukla *et al.*, *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions) : September 7-9, 2016, venue, Amity University Uttar Pradesh, Noida, India*. 2016.

[5] A. T. S, P. Hari babu, and R. Divya Sree, “IR Sensor Based Obstacle Detection and Avoiding Robot PJAEE, 17 (9) (2020) IR Sensor Based Obstacle Detection and Avoiding Robot,” 2020.

[6] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A Literature Review,” *Journal of Computer and Communications*, vol. 03, no. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.

[7] Y. Singh Parihar and Y. S. Parihar, “Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products IoT based Controlled Soilless vertical farming with hydroponics NFT system using microcontroller View project Learning Management system View project Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products,” JETIR, 2019. [Online]. Available: www.jetir.org

[8] Prakhar Srivastava, Mohit Bajaj, and Ankur Singh Rana, *2018 IEEMA Engineer Infinite Conference (eTechNxT)*. IEEE, 2018.

[9] T. Thamaraimanalan, S. P. Vivekk, G. Satheeshkumar, and P. Saravanan, “Smart Garden Monitoring System Using IOT,” 2018. [Online]. Available: www.ajast.net

[10] A. Al Dahoud and M. Fezari, “NodeMCU V3 For Fast IoT Application Development,” 2018.

[11] I. Mosashvili and S. Oniani, “Distance Monitoring System with ESP8266 for Industrial Automation Machines,” in *IDAACS-SWS 2020 - 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, Proceedings*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020. doi: 10.1109/IDAACS-SWS50031.2020.9297094.

[12] Prof. L. Kumar, “Vehicle Speed Detection System using IR Sensor,” *Int J Res Appl Sci Eng Technol*, vol. 8, no. 5, pp. 1563–1567, May 2020, doi: 10.22214/ijraset.2020.5253.

[13] C.-C. Hsieh, T.-L. Wong, J.-E. Wang, and M.-J. Chung, “THE EFFECT OF TWO DIFFERENT WEIGHTED BADMINTON RACKETS ABOUT VELOCITY AND TORQUE WHEN OUTSTANDING BADMINTON PLAYERS WAS PERFORMING SMASH MOVEMENT,” 2018.

[14] E. Blomstrand and M. Demant, “Simulation of a Badminton Racket,” 2019.

[15] M. King, H. Towler, R. Dillon, and S. McErlain-Naylor, “A correlational analysis of shuttlecock speed kinematic determinants in the badminton jump smash,” *Applied Sciences (Switzerland)*, vol. 10, no. 4, Feb. 2020, doi: 10.3390/app10041248.

[16] C. Khawas and P. Shah, “Application of Firebase in Android App Development-A Study,” *Int J Comput Appl*, vol. 179, no. 46, pp. 49–53, Jun. 2018, doi: 10.5120/ijca2018917200.
[17] S. Khedkar and S. Thube, “Real Time Databases for Applications,” *International Research Journal of Engineering and Technology*, 2017, [Online]. Available: www.irjet.net
[18] L. Moroney, “An Introduction to Firebase,” in *The Definitive Guide to Firebase*, Apress, 2017, pp. 1–24. doi: 10.1007/978-1-4842-2943-9_1.
[19] S.-C. Kong and H. Abelson, “Computational Thinking Education.”
[20] S. B. Mir and G. F. Llueca, “Introduction to Programming Using Mobile Phones and MIT App Inventor,” *Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 15, no. 3, pp. 192–201, Aug. 2020, doi: 10.1109/RITA.2020.3008110.

APPENDICES

Appendix A Gantt Chart for BDP1

NO.	Project Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
1.	Intro & Project planning														
2.	Literature review														
3.	Methodology														
4.	Preliminary testing/investigation														
5.	Report writing & presentation														

Appendix B Gantt Chart for BDP2

ACTIVITIES PLANNING WORK GANTT CHART FOR PSM2

No.	Activities	week 1	week 2	week 3	week 4	week 5	week 6	week 7		week 8	week 9	Week 10	week 11	week 12	week 13	week 14
1	Do correction on the prototype based of Panel comment								S							
2	Continue build the prototype								E							
3	Meet Supervisor and discuss about the project								M							
4	Discuss about the Data Analysis with Supervisor								E							
5	Submit Claim form for PSM								S							
6	Present to Supervisor								T							
7	Collect the Data Analysis 1								E							
8	Collect the Data Analysis 2								R							
9	Collect the Data Analysis 3															
10	Do Final test on the prototype								B							
11	Add the Data Analysis and Result in PSM2 Report								R							
12	Submit PSM2 Report to Supervisor								E							
13	Preperation for Presentation PSM2								A							
14	Submit PSM2 Report and Logbook								K							