REAL TIME SYSTEM AND INTERNET OF THINGS FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

Bluetooth RC (BRC)

Group B6

Aldrian Raffi Wicaksono	**2106653256**
Ivan Indrastata Ramadhan	**2106706981**
Naufal Febriyanto	**2106702674**
Raden Bagus S. K. R. G. R.	**2106702610**

PREFACE

In the ever-evolving world of technology, miniaturization and automation are taking center stage. This project delves into the realm of these advancements by exploring the creation of a Bluetooth RC (BRC) using an ESP32 microcontroller and FreeRTOS operating system. Inspired by the burgeoning field of Deskbots, these miniature robots aim to inject a playful and interactive element into our daily lives.

This report will document the entire journey of building the BRC, from concept and component selection to programming and testing. We will explore the technical intricacies of the ESP32 microcontroller, its functionalities, and its integration with FreeRTOS for real-time multitasking. We will delve into the world of motor control and OLED displays, equipping the BRC with the ability to move and express itself. Moreover, we will explore the possibilities of wireless control and facial expressions, further enhancing the interactive nature of the robot.

Throughout the report, we will strive to provide a clear and concise explanation of the technical aspects involved, making it accessible to readers with diverse technical backgrounds. We will also share our experiences and challenges encountered during the project, offering valuable insights for aspiring roboticists and enthusiasts.

This project aims to encourage exploration and innovation within the field of miniaturized robots. We believe that the BRC serves as a stepping stone for further advancements in Deskbots, contributing to a future where playful robots seamlessly integrate into our daily lives.

Depok, December 06, 2023

Group B6

TABLE OF CONTENTS

CHAPTER 1 4

INTRODUCTION 4

1.1 PROBLEM STATEMENT 4

1.3 ACCEPTANCE CRITERIA 5

1.4 ROLES AND RESPONSIBILITIES 5

1.5 TIMELINE AND MILESTONES 5

CHAPTER 2 7

IMPLEMENTATION 7

2.1 HARDWARE DESIGN AND SCHEMATIC 7

2.2 SOFTWARE DEVELOPMENT 7

2.3 HARDWARE AND SOFTWARE INTEGRATION 8

CHAPTER 3 9

TESTING AND EVALUATION 9

3.1 TESTING 9

3.2 RESULT 9

3.3 EVALUATION 10

CHAPTER 4 11

CONCLUSION 11

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

The modern workspace often lacks a playful and engaging element. While technology has revolutionized many aspects of our lives, it has also introduced a certain degree of monotony and isolation into our daily routines. Traditional desktop accessories, while functional, often fail to provide a sense of interactivity or amusement.

This project seeks to address this problem by introducing the Bluetooth RC (BRC). This miniature robot aims to inject a sense of fun and interaction into the workspace, providing users with a companion that can respond to their presence and engage in playful activities.

The BRC addresses the following key issues:

- **Lack of Playfulness**: Traditional desktop accessories often focus solely on functionality, neglecting the potential for playfulness and amusement. The BRC aims to bridge this gap by incorporating playful movements, animated expressions, and interactive features.
- **Monotony and Isolation**: The modern workspace can be a monotonous and isolating environment. The BRC seeks to combat this by providing a companion that can respond to user presence, offer simple interactions, and inject a sense of liveliness into the workspace.
- **Limited Functionality**: Existing miniature robots often lack sufficient functionality to be truly engaging. The BRC aims to overcome this limitation by incorporating features such as motor control, OLED display, and wireless connectivity, allowing for a wider range of interactions and possibilities.

By addressing these issues, the BRC has the potential to create a more stimulating and engaging workspace experience, fostering creativity, reducing stress, and enhancing overall well-being.

1.2 PROPOSED SOLUTION

The Bluetooth RC (BRC) proposes a playful and engaging solution to the monotony and limited functionality of the modern workspace. By incorporating interactive movements, animated expressions, and playful sound effects, the BRC injects a sense of fun and companionship into the daily routine. Sensors allow the BRC to perceive user presence and respond with corresponding actions, while wireless communication enables remote control and personalized interactions. The powerful ESP32 microcontroller, combined with the FreeRTOS operating system, provides robust processing power for real-time multitasking and advanced functionalities. Moreover, the open-source platform encourages further development and innovation, fostering a community of creators and enthusiasts. The BRC's playful nature and interactive capabilities have the potential to transform the workspace, fostering creativity, reducing stress, and enhancing overall well-being.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. The system must be able to be controlled via bluetooth
2. The system must be able to drive by itself using Ultrasonic Sensor to detect obstacle

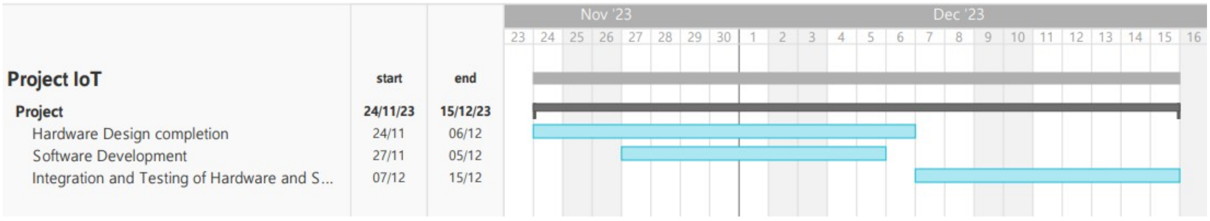
1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Programming	Provide the code needed for the project	Aldrian Raffi Wicaksono
Schematic and Supplier	Create schemes, as well as search for and procure the goods and tools needed	Raden Bagus S. K. R. G. R.
Role 3	Role 3 responsibilities	Person 3
Role 4	Role 4 responsibilities	Person 4

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES



1. Hardware Design completion: A milestone indicating the date when the hardware design for the embedded system is finalized, including schematic.
2. Software Development: The date when the development of the user-created assembly code (software) begins, focusing on specific tasks and functionalities.
3. Integration and Testing of Hardware and Software: A milestone indicating when the hardware and software components are integrated and tested together to ensure proper functionality.
4. Final Product Assembly and Testing: A milestone marking when the final system product is assembled, tested, and verified to meet the acceptance criteria.

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

The hardware design of the BRC revolves around the powerful ESP32 microcontroller as the central processing unit. The ESP32 is chosen for its ample processing power, memory capacity, and integrated Wi-Fi and Bluetooth connectivity, allowing for a feature-rich and versatile robot. These are the core components,

- **ESP32 Microcontroller**: As mentioned above, the ESP32 serves as the brain of the BRC. It handles data processing, sensor readings, motor control, wireless communication, and all other core functionalities.
- **Motor Drivers**: The BRC utilizes two motor drivers to control its movement. These drivers receive signals from the ESP32 and translate them into corresponding commands for the motors, enabling the robot to move forward, backward, and turn.
- **Motors**: The BRC is equipped with two small DC motors that provide the driving force for its movements. These motors should be chosen based on their size, weight, and torque output to ensure optimal performance within the robot's design constraints.
- **Sensors**: The BRC incorporates ultrasonic sensors to detect obstacles in front of it. The sensor acts as the eye of the device to “see” obstacles in front of it.
- **OLED Display**: The BRC utilizes an OLED display to showcase its animated expressions and provide information to users. This display should be chosen based on its size, resolution, and color capabilities to effectively convey the desired information and expressions.
- **Battery**: The BRC requires a power source to operate. A rechargeable battery is recommended for convenience and portability. The battery should be chosen based on its capacity and voltage to ensure sufficient power for the robot's operations.

2.2 SOFTWARE DEVELOPMENT

The software for the BRC plays a crucial role in controlling its movements, interpreting sensor data, and orchestrating its various functionalities. The ESP32's powerful processing capabilities and the FreeRTOS operating system provide a robust platform for developing the BRC's software.

The core functionalities of the BRC software can be broadly categorized into three key areas:

1. Motor Control: The software receives motor commands from the user interface or internal logic and translates them into signals for the motor drivers. Different motor commands are used to control the direction, speed, and duration of the robot's movement. Motor control algorithms are implemented to ensure smooth and coordinated movement of the robot.
2. Sensor Data Interpretation: The software continuously reads data from the various sensors equipped on the BRC, such as the accelerometer, gyroscope, proximity sensor, and light sensor. This data is then processed and used to trigger appropriate responses from the robot. For example, the robot may change its movement direction based on the accelerometer data or activate its OLED display based on proximity detection.
3. User Interaction and Communication: The software allows users to interact with the BRC through a user interface, either directly on the robot itself or remotely via a mobile app or web interface. This user interface provides controls for movement, animation selection, and customization of various robot functionalities. The software also utilizes the ESP32's Wi-Fi and Bluetooth capabilities to facilitate communication with external devices and enable remote control of the robot.

The FreeRTOS operating system enables multitasking on the ESP32, allowing the software to handle multiple tasks simultaneously. This is crucial for the BRC as it needs to perform motor control, sensor data interpretation, and user interaction concurrently.

The software utilizes FreeRTOS tasks to run these functionalities independently. Each task has its own priority and execution frequency, ensuring that critical tasks are executed in a timely manner while maintaining overall system responsiveness.

Code Structure and Workflow: The BRC's software is typically structured into several modules, each responsible for a specific functionality. This modular approach promotes code readability, maintainability, and reusability.

The workflow of the software can be summarized as follows:

- Initialization: This phase involves setting up the ESP32, initializing peripherals (such as sensors and motor drivers), and loading the necessary configuration data.
- Main Loop: The main loop continuously runs and performs the following tasks:
    - Reads sensor data
    - Processes user input and commands
    - Updates motor commands based on sensor data, user input, and internal logic
    - Drives the motors according to the updated commands
    - Updates the OLED display with animations and information
    - Handles communication with external devices
- Interrupt Service Routines (ISRs): These routines handle high-priority events such as sensor triggers and button presses. They typically perform minimal processing and then signal the main loop to take further action.

The code is really simple, it receives data over wifi and separates the string and controls the motor depending on the command reviewed. "roboteyes.h","seperatestring.h" and "MotorController.h" are custom libraries. These are the libraries that used in the project.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "WiFi.h"
#include "roboteyes.h"
#include "AsyncUDP.h"
#include "seperatestring.h"
#include "MotorController.h"
#define freertos/queue.h
```

This code is where the wifi is defined

```
const char * ssid = "Aldrian Raffi";//Change your wifi name
const char * password = "rianraffi2";//Change your wifi password
```

This is the UDP port. We made it 1234 so it's easy to remember

```
if(udp.listen(1234)) {//UDP port number if there is any problem change this.
```

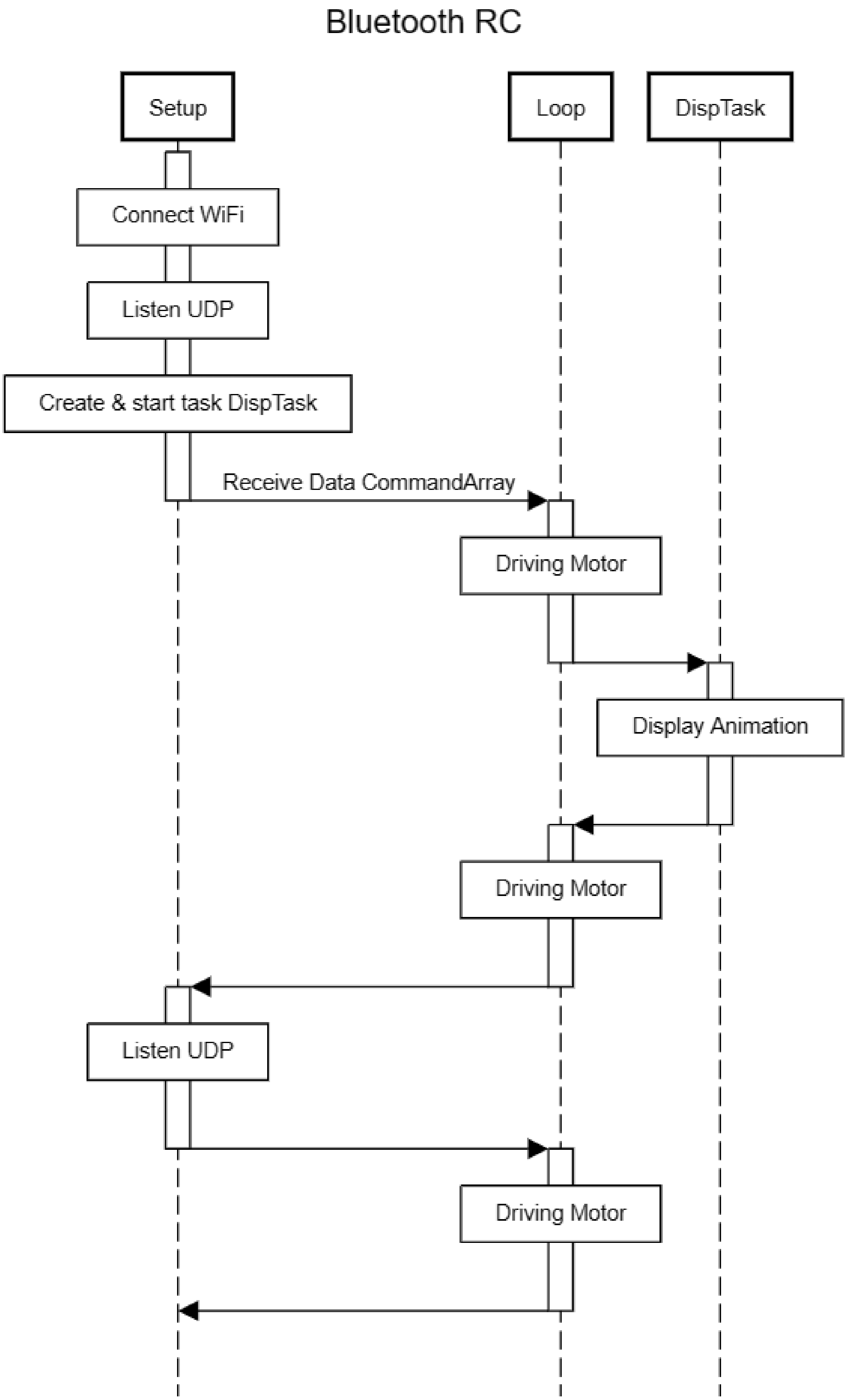
Here we create "dispTask" which handles the facial animations. Thanks to FreeRTOS we can play the animation and move the robot simultaneously. There are better ways of doing this, like using a queue, but we could not make it work.

```
xTaskCreatePinnedToCore(
    dispTask,   // Display task
    "dispTask", // Task name
    4096,       // Stack size
    NULL,       // No parameters
    0,          // Priority
    &xHandle,       // No handle returned
 0);
```

This function reads the incoming data over wifi and separates each command into an array

```
Plzgetstring(PhrasedData);// Get the incoming data and seprerate the commands into array.
```

Sequence Diagram

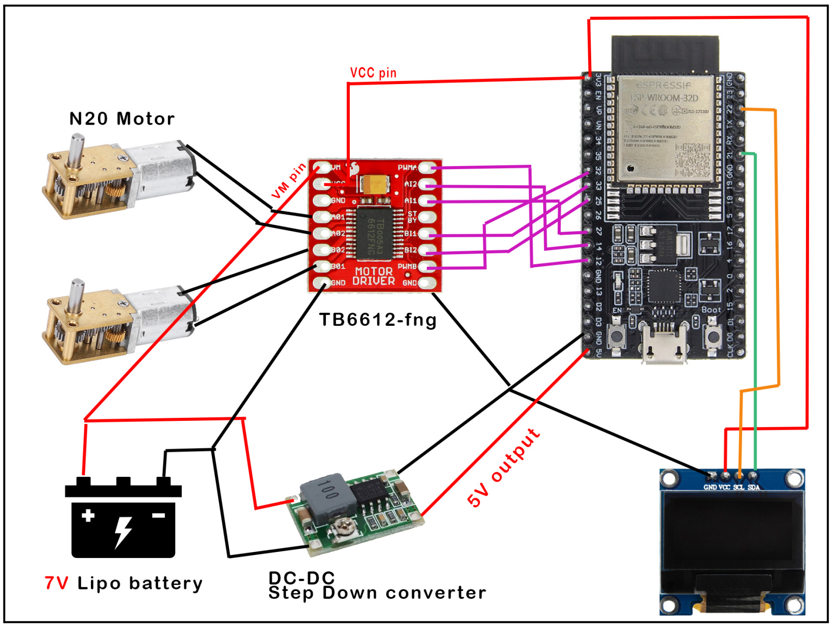


2.3 HARDWARE AND SOFTWARE INTEGRATION

The successful operation of the Bluetooth RC (BRC) hinges on seamless integration between its hardware and software elements. This process involves connecting the various physical components to the ESP32 microcontroller and developing software that controls and orchestrates their individual functionalities.

The physical components of the BRC, such as the motors, sensors, OLED display, and battery, need to be connected to the ESP32 microcontroller using appropriate wiring and connectors. This typically involves:

- Motor Drivers: Connecting the motor drivers to the ESP32's PWM pins and power supply.
- Sensors: Connecting each sensor to the designated ESP32 pins according to its specific communication protocol (e.g., I2C, SPI).
- OLED Display: Connecting the display to the ESP32 using appropriate data and power lines based on its interface (e.g., SPI, I2C).
- Battery: Connecting the battery to the power supply circuit of the ESP32, ensuring proper voltage regulation.
- Additional Components: Connecting any additional components, such as buttons, LEDs, or communication modules, based on their specific requirements.
- It's essential to consult the datasheets and specifications of each component used in the BRC to ensure proper connection and configuration.



CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

I have not been able to do hardware testing on my system yet, due to some unforeseen circumstances. However, we have uploaded the code to the ESP32 board, which was compiled without any errors using the Arduino IDE. This shows that the code is syntactically valid and compatible with the board. The next step is to test the robot using the Android app that we have developed using Android Studio. We will use the app to send commands to the robot, and observe its behavior and response. We will also measure the battery life, the wifi range, and the latency of the communication. We will document the results and any issues that arise during the testing process. We hope to complete the testing as soon as possible and demonstrate the functionality and performance of the robot.

3.3 EVALUATION

For this project, we will use both quantitative and qualitative methods to evaluate the robot and the app. The quantitative methods will include measuring the accuracy, speed, and reliability of the robot's movements and expressions, as well as the app's usability and functionality. The qualitative methods will include conducting surveys and interviews with potential users and stakeholders to gather their opinions and suggestions on the robot and the app. We will compare the results with the objectives and requirements of the project, and identify the strengths and weaknesses of the system. We will also discuss the limitations and challenges of the evaluation process, and propose future directions for further development and testing of the system. We planned to present the findings and conclusions of the evaluation in a clear and concise manner, using tables, graphs, and charts to illustrate the data.

CHAPTER 4

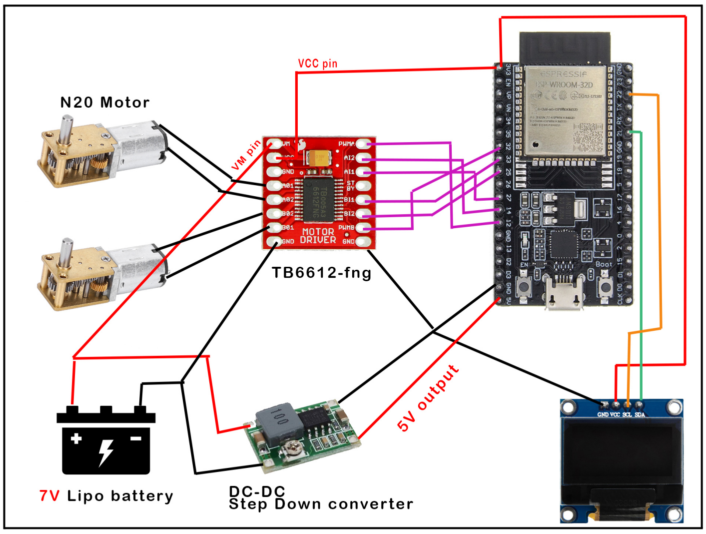
CONCLUSION

Conclusion is a vital part of any project, as it summarizes the main points and outcomes of the project, and reflects on the learning and challenges encountered. For this project, we have designed and developed a tiny desktop robot that can display facial expressions and be controlled wirelessly using an ESP32 board and freeRTOS. we have also created an Android app that can communicate with the robot and send commands to it. we have used 3D printing, Arduino, and Android Studio as the main tools and technologies for this project. However, we have not met the deadline on time, due to some unforeseen circumstances that prevented me from doing hardware testing on my system. This is a major limitation and drawback of my project, as it means that we have not verified the functionality and performance of the robot and the app, and we have not evaluated the quality and effectiveness of the system. Therefore, we cannot claim that my project has achieved the objectives and requirements that we have set.

However, we have a clear plan for testing and evaluation that we have not done yet, which involves using the app to send commands to the robot, and observing its behavior and response. We will also measure the battery life, the wifi range, and the latency of the communication. We will document the results and any issues that arise during the testing process. We will also plan to conduct surveys and interviews with potential users and stakeholders to gather their opinions and suggestions on the robot and the app. We will use various tools and techniques to collect and analyze the data, and compare the results with the objectives and requirements of the project. We planned to identify the strengths and weaknesses of the system, and propose future directions for further development and testing of the system. We hope to complete the testing and evaluation as soon as possible and demonstrate the functionality and performance of the robot. We have learned a lot from this project, and we have faced many challenges and difficulties. We have also gained valuable skills and knowledge in robotics, Arduino, and Android development. We have enjoyed working on this project, and we are proud of what we have accomplished so far. We are also grateful for the feedback and support that we have received from my supervisor and peers. We look forward to completing the project and presenting it to the audience.

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation