****

**REAL TIME SYSTEM AND INTERNET OF THINGS FINAL PROJECT REPORT**

**DEPARTMENT OF ELECTRICAL ENGINEERING**

**UNIVERSITAS INDONESIA**

**Bluetooth RC**

**Group B6**

**Aldrian Raffi Wicaksono 2106653256 1**

**Ivan Indrastata Ramadhan 2106706981 D 2**

**Naufal Febriyanto 2106702674 D 3**

**Raden Bagus S. K. R. G. R. 2106702610 D 4**

**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 Tiny Desktop Robot (TDR) 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 TDR, 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 TDR 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 TDR 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**](#_Toc133669167)

[**INTRODUCTION 4**](#_Toc133669168)

[1.1 PROBLEM STATEMENT 4](#_Toc133669169)

[1.3 ACCEPTANCE CRITERIA 5](#_Toc133669170)

[1.4 ROLES AND RESPONSIBILITIES 5](#_Toc133669171)

[1.5 TIMELINE AND MILESTONES 5](#_Toc133669172)

[**CHAPTER 2 7**](#_Toc133669173)

[**IMPLEMENTATION 7**](#_Toc133669174)

[2.1 HARDWARE DESIGN AND SCHEMATIC 7](#_Toc133669175)

[2.2 SOFTWARE DEVELOPMENT 7](#_Toc133669176)

[2.3 HARDWARE AND SOFTWARE INTEGRATION 8](#_Toc133669177)

[**CHAPTER 3 9**](#_Toc133669178)

[**TESTING AND EVALUATION 9**](#_Toc133669179)

[3.1 TESTING 9](#_Toc133669180)

[3.2 RESULT 9](#_Toc133669181)

[3.3 EVALUATION 10](#_Toc133669182)

[**CHAPTER 4 11**](#_Toc133669183)

[**CONCLUSION 11**](#_Toc133669184)

# CHAPTER 1

# INTRODUCTION

## 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 Tiny Desktop Robot (TDR). 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 TDR addresses the following key issues:

* **Lack of Playfulness**: Traditional desktop accessories often focus solely on functionality, neglecting the potential for playfulness and amusement. The TDR 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 TDR 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 TDR 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 TDR has the potential to create a more stimulating and engaging workspace experience, fostering creativity, reducing stress, and enhancing overall well-being.

* 1. **PROPOSED SOLUTION**

The Tiny Desktop Robot (TDR) 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 TDR injects a sense of fun and companionship into the daily routine. Sensors allow the TDR 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 TDR's playful nature and interactive capabilities have the potential to transform the workspace, fostering creativity, reducing stress, and enhancing overall well-being.

## 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

## ROLES AND RESPONSIBILITIES

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

|  |  |  |
| --- | --- | --- |
| Roles | Responsibilities | Person |
| Programming | Program the code needed for the project | Person 1 |
| Schematic and Supplier | Create schemes, as well as search for and procure the goods and tools needed | Person 2 |
| Role 3 | Role 3 responsibilities | Person 3 |
| Role 4 | Role 4 responsibilities | Person 4 |

Table 1. Roles and Responsibilities

## 

## TIMELINE AND MILESTONES

Insert Gantt Chart here. The Gantt Chart should consist of date interval for:

a) Hardware Design completion: A milestone indicating the date when the hardware design for the embedded system is finalized, including schematic.  
b) Software Development: The date when the development of the user-created assembly code (software) begins, focusing on specific tasks and functionalities.  
c) 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.  
d) 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 TDR 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 TDR. It handles data processing, sensor readings, motor control, wireless communication, and all other core functionalities.
* **Motor Drivers**: The TDR 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 TDR 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 TDR incorporates ultrasonic sensor to detect obstacle in front of it. The sensor act as the eye of the device to “see” obstacle in front of it.
* **OLED Display**: The TDR 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 TDR 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 TDR 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 TDR's software.

The core functionalities of the TDR 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 TDR, 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 TDR 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 TDR 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 TDR'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 = "SINGTEL-42E6";//Change your wifi name

const char \* password = "huiweimaev";//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.

Include the flowchart.

## 2.3 HARDWARE AND SOFTWARE INTEGRATION

The successful operation of the Tiny Desktop Robot (TDR) 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 TDR, 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 TDR to ensure proper connection and configuration.

Here's an illustration showing a typical hardware connection layout for a TDR:

# CHAPTER 3

# TESTING AND EVALUATION

## 3.1 TESTING

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ornare accumsan nisl sit amet sodales. Suspendisse sed dictum velit, in suscipit sem. Vestibulum egestas neque vel velit tristique, id venenatis nunc fringilla. Mauris condimentum diam consequat egestas tincidunt. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus semper pharetra commodo. Integer hendrerit ultricies lacus. Nullam id magna sed risus placerat luctus sed at mauris. Curabitur ligula urna, pharetra eget mi sit amet, sagittis feugiat magna. Curabitur ex nisl, eleifend et mattis sit amet, condimentum non nisi.

Donec at iaculis leo. Integer congue sed lacus suscipit iaculis. Nulla a augue ut sapien rutrum consectetur. Sed ac dignissim lorem. Maecenas hendrerit nisl a metus posuere, vel vehicula metus eleifend. Mauris blandit, dolor nec malesuada tempor, purus nibh aliquet nibh, at faucibus leo felis a nisi. Donec pharetra leo risus, in vestibulum dui laoreet in. Nulla facilisi. Etiam nec consequat justo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Etiam pharetra eleifend hendrerit.

## 3.2 RESULT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ornare accumsan nisl sit amet sodales. Suspendisse sed dictum velit, in suscipit sem. Vestibulum egestas neque vel velit tristique, id venenatis nunc fringilla. Mauris condimentum diam consequat egestas tincidunt. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus semper pharetra commodo. Integer hendrerit ultricies lacus. Nullam id magna sed risus placerat luctus sed at mauris. Curabitur ligula urna, pharetra eget mi sit amet, sagittis feugiat magna. Curabitur ex nisl, eleifend et mattis sit amet, condimentum nonnisi.

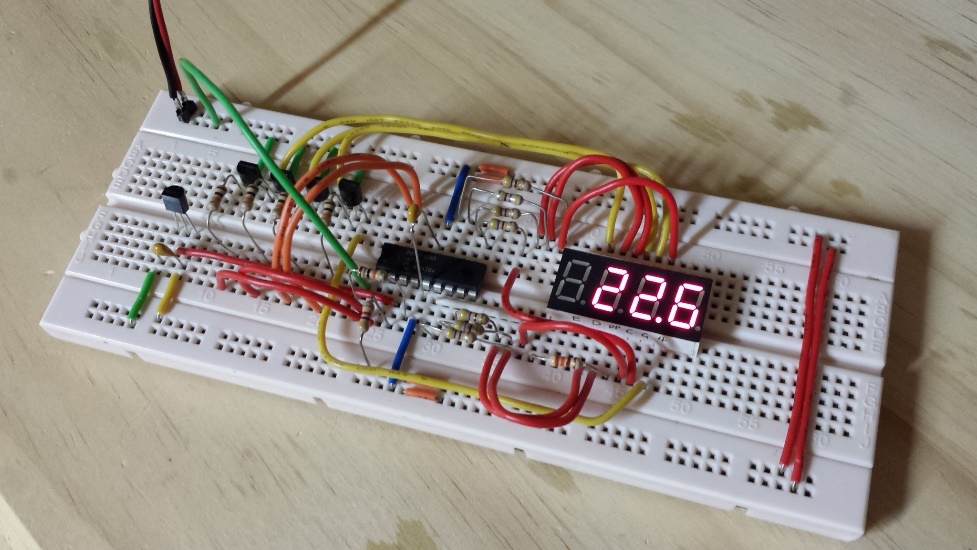


Fig 2. Testing Result

Donec at iaculis leo. Integer congue sed lacus suscipit iaculis. Nulla a augue ut sapien rutrum consectetur. Sed ac dignissim lorem. Maecenas hendrerit nisl a metus posuere, vel vehicula metus eleifend. Mauris blandit, dolor nec malesuada tempor, purus nibh aliquet nibh, at faucibus leo felis a nisi. Donec pharetra leo risus, in vestibulum dui laoreet in. Nulla facilisi. Etiam nec consequat justo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Etiam pharetra eleifend hendrerit.

## 3.3 EVALUATION

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ornare accumsan nisl sit amet sodales. Suspendisse sed dictum velit, in suscipit sem. Vestibulum egestas neque vel velit tristique, id venenatis nunc fringilla. Mauris condimentum diam consequat egestas tincidunt. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus semper pharetra commodo. Integer hendrerit ultricies lacus. Nullam id magna sed risus placerat luctus sed at mauris. Curabitur ligula urna, pharetra eget mi sit amet, sagittis feugiat magna. Curabitur ex nisl, eleifend et mattis sit amet, condimentum non nisi.

Donec at iaculis leo. Integer congue sed lacus suscipit iaculis. Nulla a augue ut sapien rutrum consectetur. Sed ac dignissim lorem. Maecenas hendrerit nisl a metus posuere, vel vehicula metus eleifend. Mauris blandit, dolor nec malesuada tempor, purus nibh aliquet nibh, at faucibus leo felis a nisi. Donec pharetra leo risus, in vestibulum dui laoreet in. Nulla facilisi. Etiam nec consequat justo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Etiam pharetra eleifend hendrerit

# CHAPTER 4

# CONCLUSION

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ornare accumsan nisl sit amet sodales. Suspendisse sed dictum velit, in suscipit sem. Vestibulum egestas neque vel velit tristique, id venenatis nunc fringilla. Mauris condimentum diam consequat egestas tincidunt. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus semper pharetra commodo. Integer hendrerit ultricies lacus. Nullam id magna sed risus placerat luctus sed at mauris. Curabitur ligula urna, pharetra eget mi sit amet, sagittis feugiat magna. Curabitur ex nisl, eleifend et mattis sit amet, condimentum non nisi.

Donec at iaculis leo. Integer congue sed lacus suscipit iaculis. Nulla a augue ut sapien rutrum consectetur. Sed ac dignissim lorem. Maecenas hendrerit nisl a metus posuere, vel vehicula metus eleifend. Mauris blandit, dolor nec malesuada tempor, purus nibh aliquet nibh, at faucibus leo felis a nisi. Donec pharetra leo risus, in vestibulum dui laoreet in. Nulla facilisi. Etiam nec consequat justo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Etiam pharetra eleifend hendrerit.

**REFERENCES**

1. Reference 1 all in IEEE format
2. Reference 2
3. Reference 3
4. Reference 4
5. Reference 5
6. Reference 6
7. And so on

**APPENDICES**

**Appendix A: Project Schematic**

Put your final project latest schematic here

**Appendix B: Documentation**

Put the documentation (photos) during the making of the project