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

SMARTLOOM CURTAIN MANAGER

GROUP A4

LUTHFI MISBACHUL MUNIR	**2106631961**
FAYZA NIRWASITA	**2106635700**
MUHAMMAD ZAKI NUR SAID HANAN	**2106733856**
MOCHAMMAD DYENTA DWIANTOMITARA	**2106731245**

PREFACE

As we enter an age defined by advances in technology, the need for the Internet of Things (IoT) has never been greater. In this dynamic landscape, where innovative solutions are paramount, our daily lives demand tools that are not only efficient but also effortlessly blend into our daily lives. It is within this context that the *IoT Smart Curtain* emerges, a beacon of ingenuity that promises to transform the way we interact with our homes and enhance our lives in countless ways.

This document ventures into the realm of Smartloom Curtain Managers, unveiling their potential to revolutionize the way we manage our comfort, privacy and even security. The technological marvel that transcends its traditional function in this product, makes you feel confident that your home is secure, because the curtains are now automatically closing to deter any unwanted intruders and control the intensity of sunlight. And the intelligence of the smart extends beyond automation. With the convenience of a dedicated remote control, the user holds the power to customize the environment to exact preferences. This seamless blend of automation and user control empowers you to personalize your home environment, creating a space that reflects unique needs and desires.

Depok, December 10, 2023

Group A4

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Developing an *Internet of Things (IoT)* enabled Smartloom Curtain Manager system presents several challenges that require careful consideration. The primal goal is to create an automated solution that optimizes curtain positions. Energy efficiency is a key concern, necessitating the implementation of algorithms that take into account factors such as time of day, weather conditions, and user preferences. This automation aims to not only enhance user convenience but also minimize energy consumption, contributing to a more sustainable and eco-friendly living environment.

Remote control functionality is another crucial aspect of the project. users should be able to control the Smartloom Curtain Manager remotely through a dedicated mobile application or other user-friendly interfaces. This entails the implementation of secure communication protocols to ensure the confidentiality and integrity of commands sent from the remote control to the curtain system. To ensure the successful deployment of the Smartloom Curtain Manager system, attention must be given to user interface design, reliability, safety, and power management. Additionally, the system's power management introduces complexities, necessitating strategic solutions to optimize energy consumption, extend lifespan, and align with sustainability goals. In essence, the challenge lies in striking a delicate balance between user-centric design to realize a comprehensive and efficient Smart Curtain system deployment.

1.2 PROPOSED SOLUTION

The "Smartloom Curtain Manager" tackles the complexities of automating your window coverings, offering an intelligent and seamless experience. Control your curtains remotely through an intuitive mobile application or user-friendly physical interfaces. Robust security measures ensure the confidentiality and integrity of your commands, protecting against unauthorized access.

Prioritizing user experience, the Smartloom Curtain Manager boasts an intuitive and customizable interface across mobile and physical controls. Robust fail-safes and error-handling mechanisms guarantee consistent and reliable operation. The balance between an auto mode using simple *Light Dependent Resistor* (LDR) for light detection and manual mode using remote control through smart devices efficiently benefits users to customize their preference environment. The product is also equipped with specific time automation, where the curtain will close and open on specific time being set by program. Designed with those features, the Smartloom Curtain Manager supports sustainability and explores optimal energy consumption.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. The system must be able to operates flawlessly and integrate accurately between hardware components and software components,
2. Firmware handles error and power efficiently, while Blynk app provides intuitive control and feedback
3. Curtain reacts within 8 seconds and reaches desired position within 5-7% margin
4. Simple and user-friendly interface with clear feedback and customization options
5. Functional Hardware where ESP32 operates as expected, LDR provides accurate readings, DC motor responds accurately,
6. Smart Curtains should be designed with convenient hardware components, to make them replaceable and easily swappable components, allowing for effortless replacement or repair within a timeframe around 6-12 months.

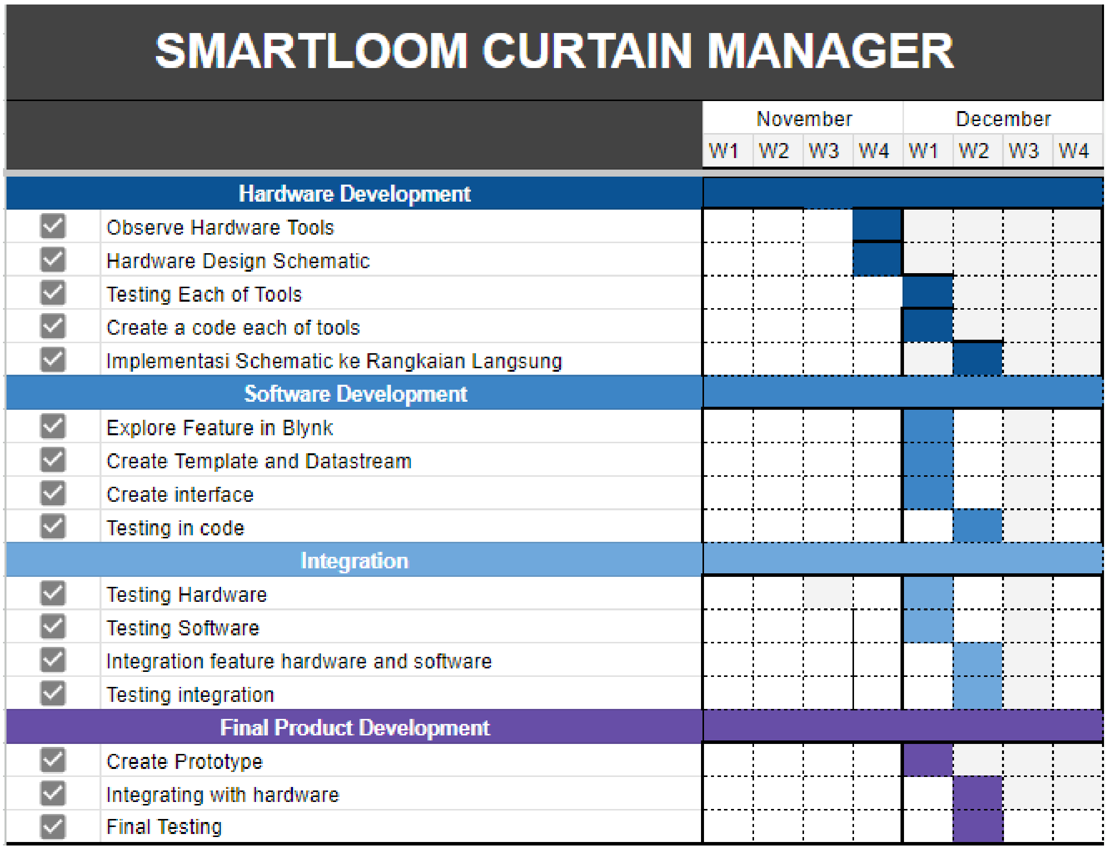
1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Project Leader	Coordinating team efforts, overseeing project integration, and ensuring alignment with project objectives.	Luthfi Misbachul Munir
Hardware Schematic Developer	Designing the blueprint for the hardware components, ensuring accuracy and coherence with project specifications.	Fayza Nirwasita
Hardware Developer	Actively contributing to the coding process and collaborating with the Hardware Schematic Developer to ensure seamless integration.	Muhammad Zaki Nur Said Hanan
Software Developer	Creating a user-friendly interface and managing datastreams for efficient communication between the user and the IoT device.	Mochammad Dyenta Dwiantomitara

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES



1.5.1 TIMELINE

1. Hardware Development
    a. Observe Hardware Tools
    Task Description: Conduct research and observation on various hardware tools required for the project, considering factors like compatibility, efficiency, and availability.
    b. Hardware Design Schematic
    Task Description: Create a detailed schematic design for the hardware components, specifying connections, modules, and their functionalities.
    c. Testing Each of Tools
    Task Description: Individually test each hardware tool to ensure functionality and identify any potential issues or limitations.
    d. Create a code each of tools
    Task Description: Develop code snippets or modules for each hardware tool, ensuring they operate correctly and can be integrated into the overall system.
    e. Implementing schematic to real hardware
    Task Description: Implement the finalized schematic design into a physical circuit, ensuring proper connections and alignments.
2. Software Development

    a. Explore Features in Blynk
       Task Description: Explore and understand the features offered by the Blynk platform, considering its capabilities and limitations for your project.
    b. Create Template and Datastream
       Task Description: Develop templates for datastreams in Blynk, establishing the communication structure between the hardware and the Blynk app.
    c. Create Interface
       Task Description: Design and implement the user interface on the Blynk app, considering user experience and interaction.
    d. Testing in Code
       Task Description: Perform testing on the software code to ensure proper communication with the hardware and responsiveness to user inputs.
3. Integration
    a. Testing Hardware
       Task Description: Conduct thorough testing of each hardware component to ensure they function correctly individually.
    b. Testing Software
       Task Description: Test the software components, including Blynk integration, to verify proper communication and functionality.
    c. Integration Feature Hardware and Software
       Task Description: Integrate the hardware and software features, ensuring seamless communication and functionality.
    d. Testing Integration
       Task Description: Conduct extensive testing on the integrated system to identify and resolve any issues arising from the combination of hardware and software.
4. Final Product Development
    a. Create Prototype
       Task Description: Develop a working prototype by assembling the final hardware components and loading the software.
    b. Integrating with Hardware
       Task Description: Integrate the software with the assembled hardware, ensuring all features work together as intended.
    c. Final Testing
       Task Description: Conduct comprehensive testing on the final product, verifying that all hardware and software components function correctly and meet project requirements.

1.5.2 MILESTONE

1. Hardware Design Completion

   Description: Finalize the hardware design, including the schematic and individual tool testing.

2. Software Development Commencement
   Description: Start developing the software, focusing on exploring Blynk features, creating templates, and designing the interface.
3. Integration and Testing of Hardware and Software
   Description: Combine the hardware and software components, conducting thorough testing to ensure seamless integration.
4. Final Product Assembly and Testing
   Description: Assemble the final product, integrating hardware and software, and conduct the last round of testing before the project completion.

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

Hardware design is the process of creating a functional circuit, while schematics are the graphical representation of that hardware design. Each component, meticulously chosen and integrated, plays a crucial role in transforming the concept of a curtain into a responsive and intelligent system. From the powerful processing capabilities of the ESP32 microcontroller to the precise light detection of the LDR module and the efficient motor control of the L298N driver, each element contributes to the seamless functionality of the Smart Curtain.

a. ESP32 Microcontroller

Fig 2.11 ESP32 Microcontroller

Responsible for processing data, communication with the Blynk app, and controlling L298N motor drivers. The microcontroller should be readily replaceable using standardized modules and sockets to ensure convenient maintenance.

b. Light Dependent Resistor (LDR) module

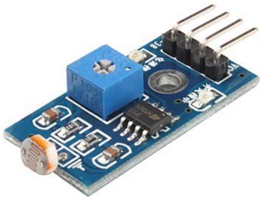


Fig 2.12 Light Dependent Resistor (LDR)

Accurately detects ambient light intensity to trigger automated curtin operation based on user-defined settings. The module also should be easily replaceable form factor and connectors for simple repair.

c. L298N Motor Driver Module

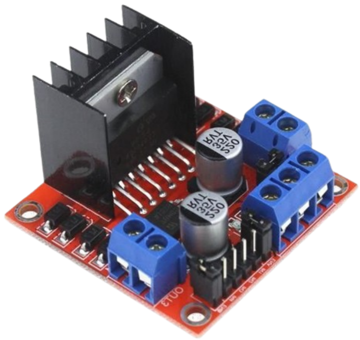


Fig 2.13 L298N Motor Driver Module

Efficiently controls the DC motor, enabling smooth and precise movement of the curtains. The driver module utilizes standard connectors for effortless swap-out in case of malfunctions.

d. ESP32 Expansion Board

Fig 2.14 ESP32 Expansion Board

Facilitates the addition of the optional components, such as additional sensors or displays, and simplifies software updates and configuration changes.

e. Cardboard safety package

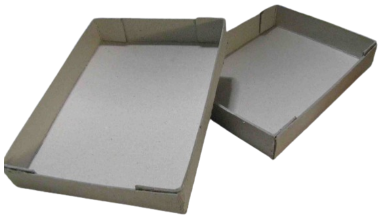


Fig 2.15 Smart Curtain Package

Provides convenient and eco-friendly protection for the hardware components.

As we delve into schematics, a clear picture emerges, revealing the intricate connections and pathways that orchestrate the symphony of hardware and software, ultimately culminating in the user-defined automation of customized home environments.

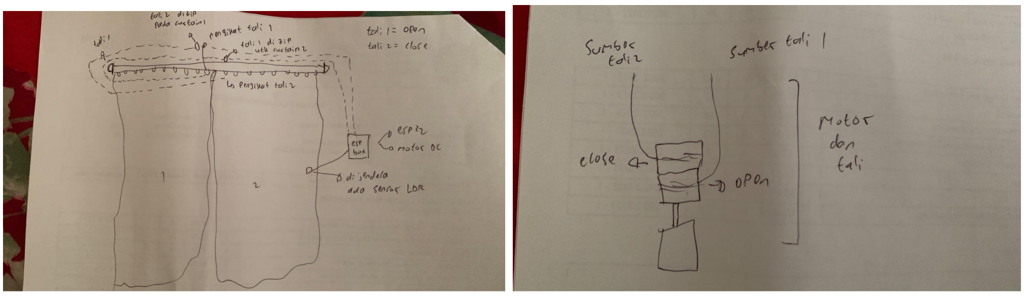
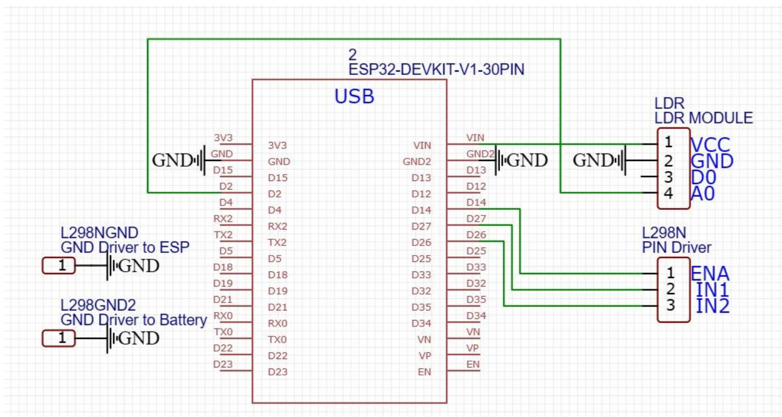


Fig. 2.16 Early Design Sketch



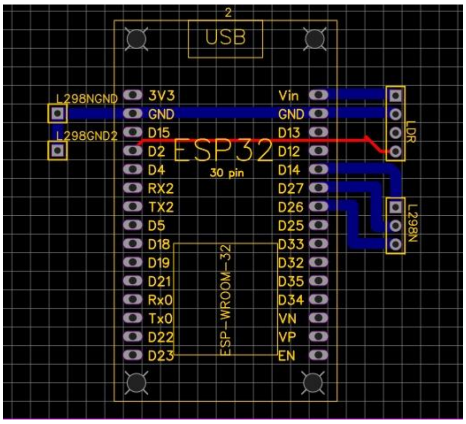


Fig. 2.17 Hardware Schematic

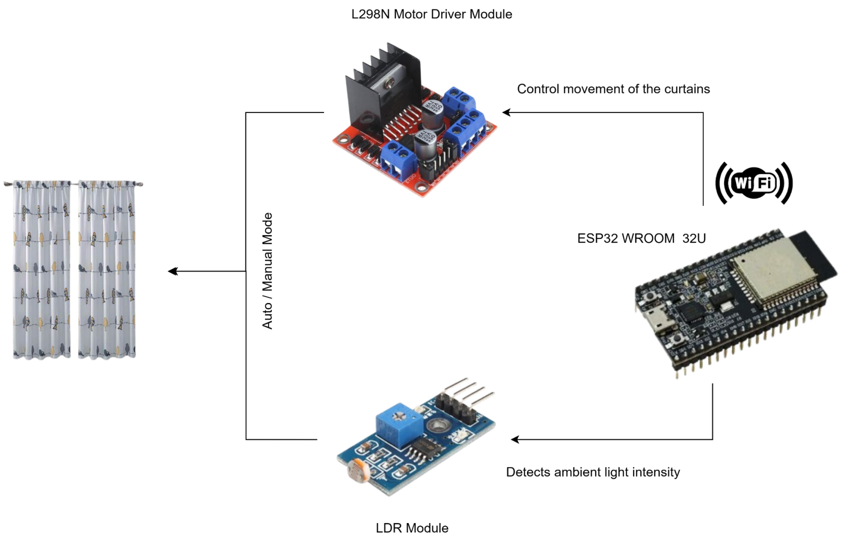


Fig 2.18 Hardware Design

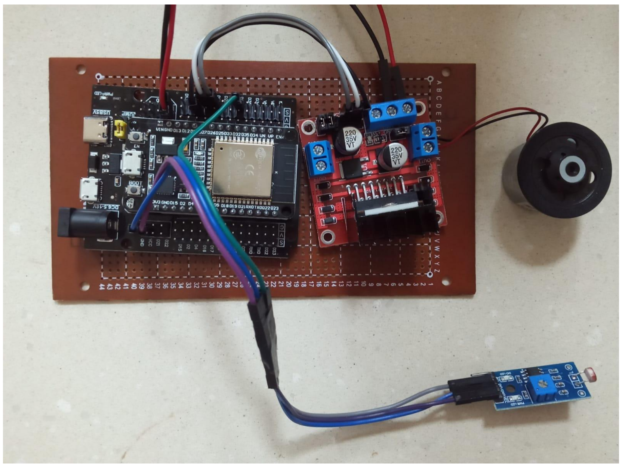
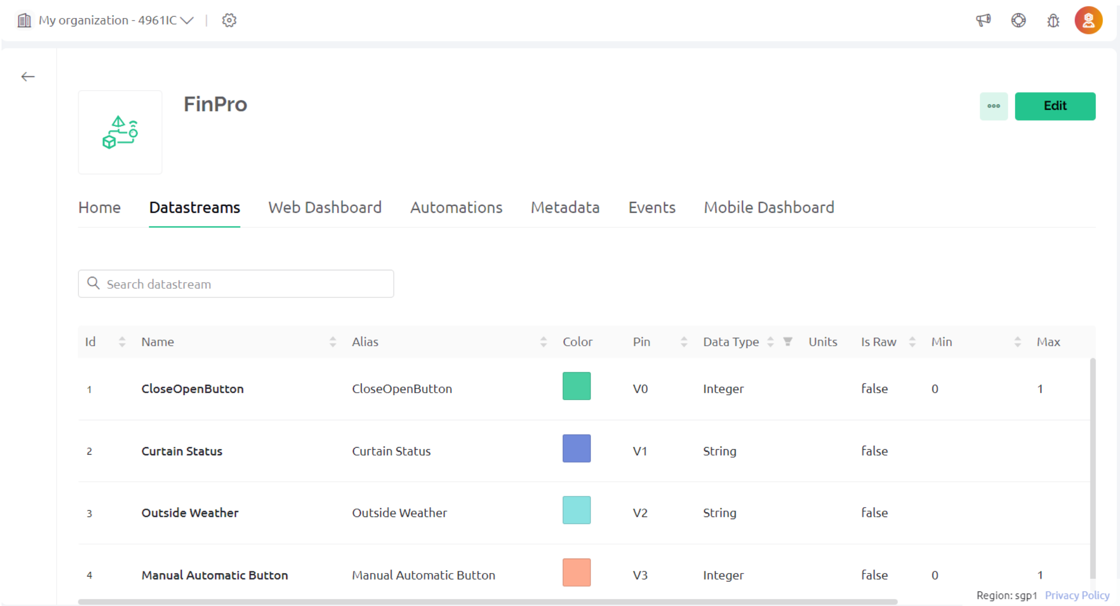


Fig. 2.19 Hardware Implementation

The LDR module, a sensitive light sensor, detects light and triggers the ESP32, the central system. The ESP32 translates light data into commands for the motor driver, to control the curtain movement in auto mode.

2.2 SOFTWARE DEVELOPMENT

In the software development phase of the project, meticulous attention has been devoted to configuring the Blynk platform, a pivotal component for seamless communication between the user interface and the embedded system. The project adopts a custom Blynk template named "FinPro," meticulously designed to cater to the specific requirements of the Smartloom Curtain Manager system. This template serves as an intuitive interface, facilitating user interaction and system monitoring.

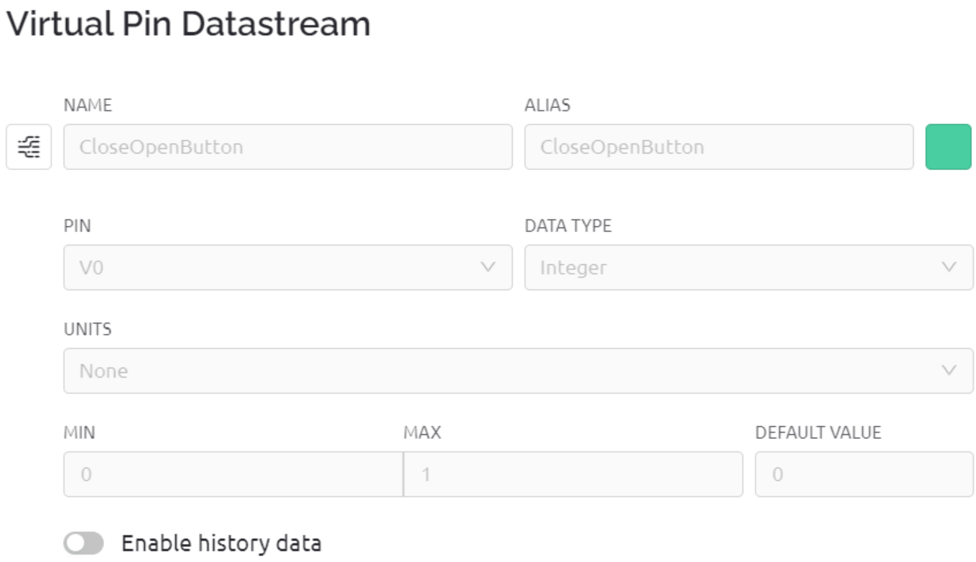


The Blynk template incorporates four distinct datastreams to enable comprehensive control and monitoring:

1. **CloseOpenButton (V0 - Integer)**

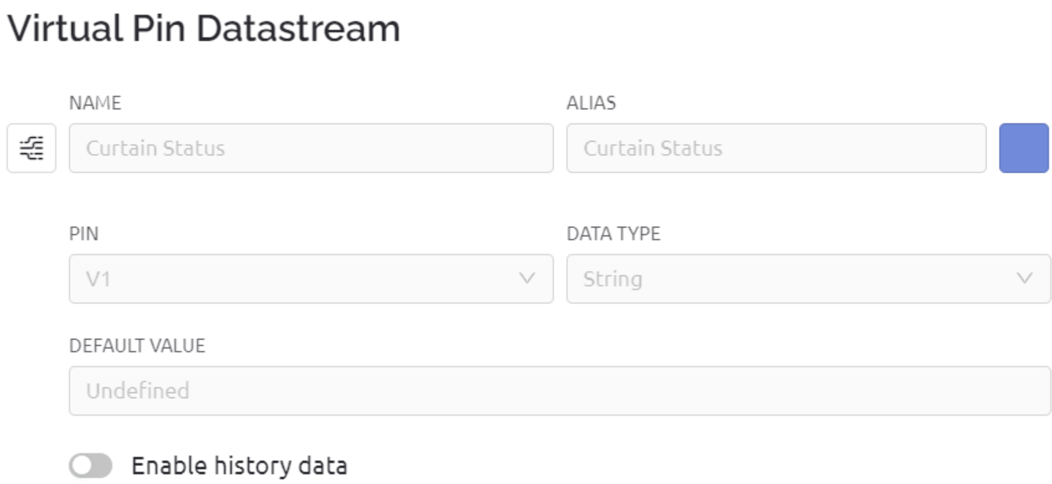
   This datastream allows users to control the curtain system efficiently. The Integer datatype is employed, where '0' corresponds to the close button, while

'1' corresponds to the open button.



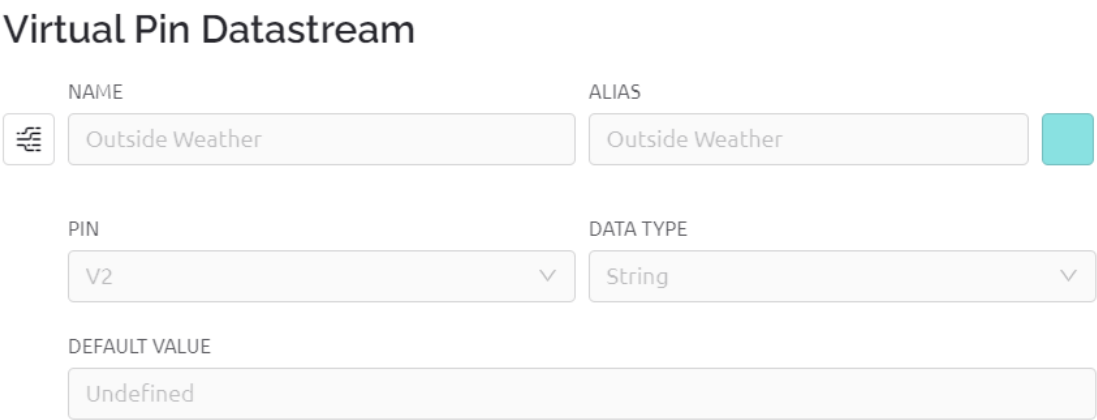
2. **Curtain Status (V1 - String)**

   Offering real-time feedback, this datastream, utilizing a String datatype, dynamically reflects the current state of the curtain, indicating whether it is in a closed or open position.



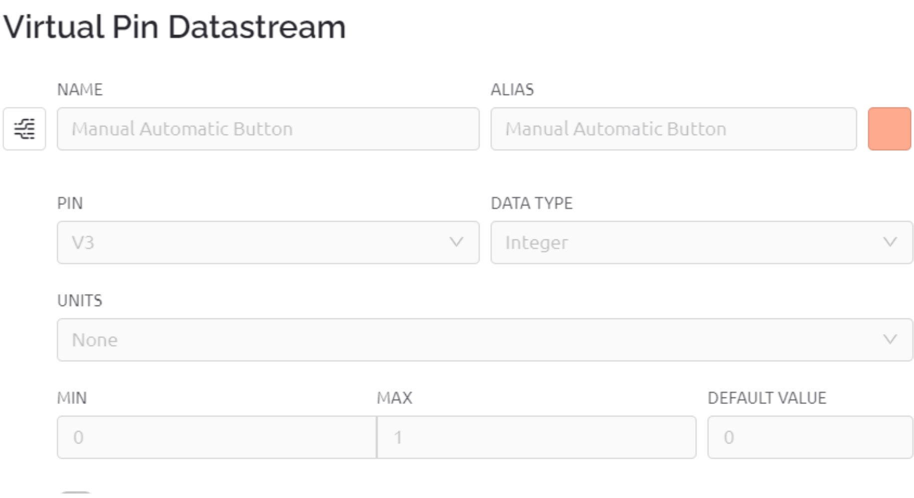
3. **Outside Weather (V2 - String)**

   Tracking outside lighting conditions, this datastream, utilizing a String datatype, conveys whether the environment is "TERANG" (bright) or "GELAP" (dark). This information is crucial for the system to make informed decisions based on ambient light.



4. **Manual Automatic Button (V3 - Integer)**

   To facilitate manual and automatic modes, this datastream employs an Integer datatype. The value '0' signifies automatic mode, where the system autonomously makes decisions based on predefined conditions. Conversely, '1' indicates manual mode, empowering users to take control.

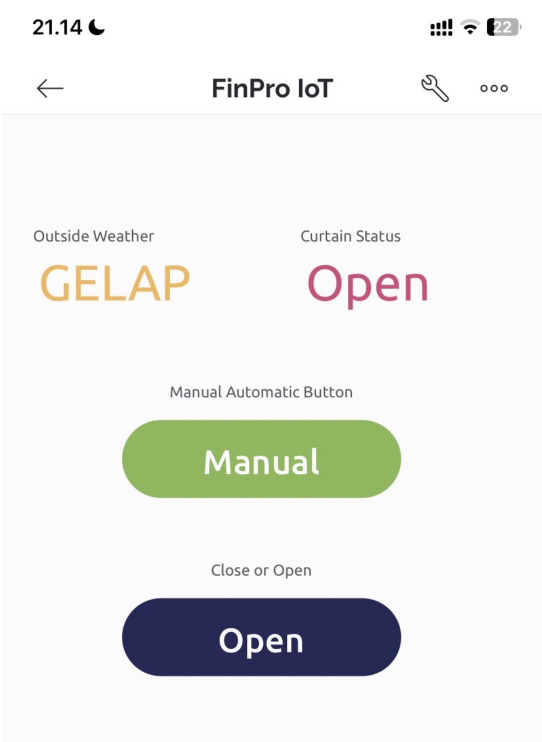


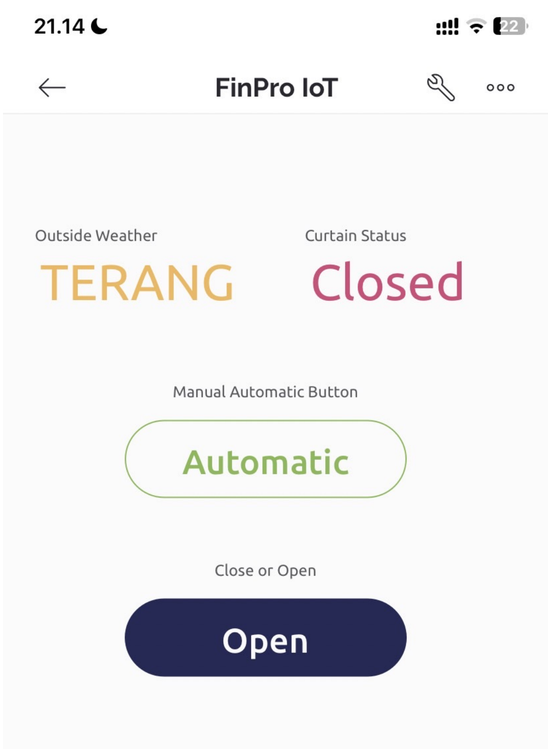
The Blynk template's user interface is thoughtfully designed to offer accessibility and ease of use. Users can effortlessly toggle between curtain open and close states, monitor the current status, and select between manual and automatic operation modes. Throughout the software development process, rigorous testing of the Blynk integration has been conducted. This ensures accurate data exchange between the user interface and the embedded system. Datastreams are continuously monitored to verify correct values and responsiveness to user inputs. By configuring Blynk with these specific datastreams, the software achieves a robust and user-centric design aligned with the functionality and goals of the Smartloom Curtain Manager system. The carefully selected datatypes and interface elements contribute to an intuitive and efficient user experience.

Within the device information section of the FinPro project, crucial firmware configurations are provided, necessitating inclusion at the forefront of the firmware code. These configurations, namely Template ID, Template Name, and AuthToken, play a pivotal role in establishing a secure and functional connection between the hardware and the Blynk platform. To seamlessly integrate the FinPro project with the Blynk platform, the following configurations are declared at the outset of the firmware code:

```
#define BLYNK_TEMPLATE_ID "TMPL67rpC5vL_"
#define BLYNK_TEMPLATE_NAME "FinPro"
#define BLYNK_AUTH_TOKEN "NXAuylaPxw7J23d-lkUp7veeleo5iKEP"
```

These declarations serve as a foundational step in configuring the firmware to communicate with the Blynk server effectively. The Template ID uniquely identifies the project, while the Template Name provides a human-readable reference. The AuthToken acts as a secure key, authenticating the device to establish a secure connection with the designated Blynk project. Including these configurations at the beginning of the firmware code ensures that the device is properly identified and authenticated within the Blynk ecosystem. This meticulous approach to firmware setup is crucial for establishing a reliable and secure interaction between the embedded system and the Blynk platform, ultimately contributing to the overall success of the FinPro Smartloom Curtain Manager project.





The user interface in the Blynk app for the FinPro project is thoughtfully designed to provide users with intuitive control and real-time information about the smart curtain system. Positioned at the top left corner is the "Outside Weather" indicator, leveraging the datastream V2. This feature dynamically communicates the current lighting conditions outside, offering a quick and accessible reference for users to gauge whether it is bright ("TERANG") or dark ("GELAP") outdoors. Adjacent to the "Outside Weather" display is the "Curtain Status" feature, powered by datastream V1. This section serves as a visual cue, promptly informing users of the current state of the curtain, whether it is "Closed" or "Opened." This real-time feedback enhances user awareness and ensures that the system's status is easily discernible.

In the center and vertically aligned in the middle of the interface lies the mode-setting section. Here, users can seamlessly toggle between "Automatic" and "Manual" modes using the dedicated button. This functionality is linked to datastream V3, providing users with convenient control over the operational mode of the IoT-enabled curtain system. Directly beneath the mode-setting button, users find the "Close" and "Open" buttons. These buttons facilitate manual control of the curtain's position, allowing users to initiate the closing or opening of the curtain with a simple tap. The functionality of these buttons is driven by datastream V0, ensuring a responsive and user-friendly experience. This carefully crafted user interface ensures that users can effortlessly interact with and monitor the FinPro smart curtain system, making the Blynk app a central hub for controlling and staying informed about the smart features integrated into their living space.

Donec 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.

Include the flowchart.

Maecenas ultrices ac felis et faucibus. Suspendisse cursus eget neque non tempus. Integer id nunc blandit, mollis risus ut, rhoncus erat. Donec eleifend porttitor justo, ut suscipit ipsum fermentum eget. Proin lacinia erat et cursus suscipit. Morbi ut neque sit amet magna posuere tempor sed at urna. Ut at faucibus libero. Sed ut massa dui. In sit amet dolor fermentum, condimentum lorem interdum, aliquam metus. Aenean tincidunt elit et mollis consectetur. Nam a elit et leo vulputate gravida convallis sed lacus.

2.3 HARDWARE AND SOFTWARE INTEGRATION

In the implementation of the FinPro project, the Blynk platform serves as a pivotal component, enabling seamless communication between the IoT device and the user's mobile application. This section provides insights into the Blynk configuration and the integration of hardware with software. The configuration of the Blynk platform involves three critical parameters - Template ID, Template Name, and AuthToken. These parameters are essential for uniquely identifying and authenticating the device on the Blynk server. The following code snippet illustrates the inclusion of these parameters in the firmware:

```
// Blynk authentication settings
#define BLYNK_TEMPLATE_ID "TMPL67rpC5vL_"
#define BLYNK_TEMPLATE_NAME "FinPro"
#define BLYNK_AUTH_TOKEN "NXAuylaPxw7J23d-lkUp7veeleo5iKEP"
#define BLYNK_PRINT Serial

#include <Arduino.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <freertos/timers.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiManager.h>
#include <BlynkSimpleEsp32.h>
#include "time.h"

char auth[] = BLYNK_AUTH_TOKEN;
```

These parameters need to be declared at the beginning of the firmware code to establish a secure and authenticated connection with the Blynk server.

To enhance user experience and device flexibility, the WiFi Manager library is employed. This library facilitates the dynamic configuration of WiFi credentials without the need for hardcoding them into the firmware. The WiFi Manager presents a user-friendly interface, allowing users to connect their ESP32 device to a WiFi network directly from their mobile phones.

```
// WiFiManager
WiFiManager wm;

// Reset WiFi settings for testing
wm.resetSettings();

// Set up WiFiManager
bool res = wm.autoConnect("Smart Curtain", "12345678");

if(!res) {
    Serial.println("Failed to connect and hit timeout");
    delay(3000);
    // Reset and try again, or maybe put it to deep sleep
    ESP.restart();
}
else {
    // Connected to WiFi
    Serial.println("Connected to WiFi");

    // Initialize connect to Blynk
    Blynk.config(auth);
}
```

The above code snippet demonstrates the integration of WiFi Manager. Upon powering up the device, it attempts to connect to a WiFi network. If unsuccessful, it initiates a configuration portal, allowing users to choose and enter WiFi credentials through their smartphones.

The integration of hardware with software is a crucial aspect of the FinPro project. This integration is achieved through the BLYNK_WRITE function, which is called whenever a Blynk widget (such as buttons or sliders) sends data to the device. The following sections highlight key aspects of this integration:

1. Datastream V0 (CloseOpenButton)

   This datastream is linked to a button on the Blynk app, allowing users to open or close the curtain. The BLYNK_WRITE(V0) function processes the incoming button state, triggering the corresponding action in the motorTask.
2. Datastream V1 (Curtain Status)

   The status of the curtain, whether open or closed, is reflected in this datastream. The openMotor() and closeMotor() functions update this status, providing real-time feedback to the user.
3. Datastream V2 (Outside Weather)

   The LDR sensor's readings determine whether the outside lighting is bright or dark (TERANG or GELAP). This information is sent to the Blynk app via V2, providing users with environmental status.
4. Datastream V3 (Manual Automatic Button)

   This datastream allows users to switch between manual and automatic modes. The BLYNK_WRITE(V3) function processes the mode change and triggers corresponding actions in the application logic.

TESTING AND EVALUATION

3.1 TESTING

In this project, testing will be conducted on the implemented system. The testing will involve the DC motor, LDR sensor, Blynk, and ESP32 utilized in the system. Testing on the ESP32 will encompass evaluating the WiFi connection and Blynk functionality. Success will be measured by the fulfillment of specified conditions and the accuracy of sensor readings. This can be observed in the serial monitor, which displays data from the system.

3.2 RESULT

A. LDR Sensor

Dark:

```
LDR Value : 2151
Jam : 21
Prev Jam : 21
Application : 1
Is active : 0
```

Bright:

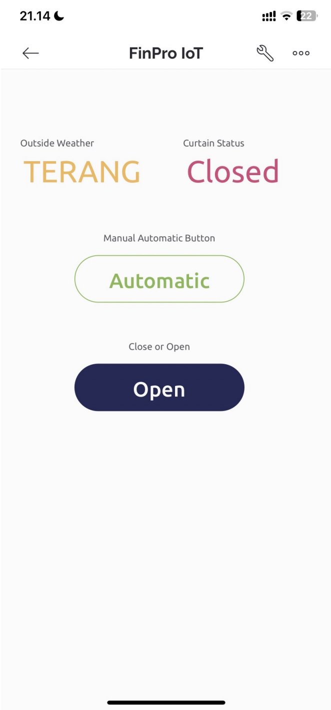
```
prev : GELAP
status : TERANG
LDR Value : 439
Jam : 21
Prev Jam : 21
Application : 0
Is active : 0
```

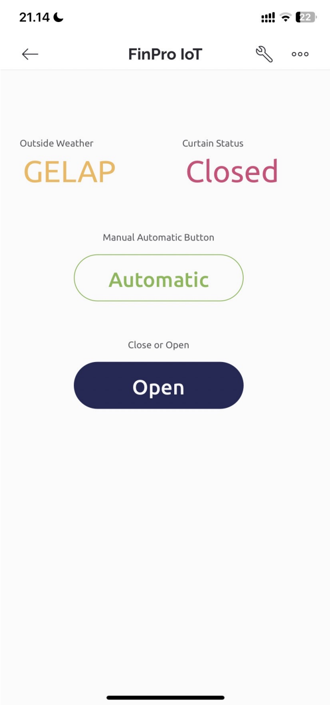
B. DC Motor

```
Button State: 0
Button Close Pressed
Closing Motor
LDR Value : 2312
Jam : 21
Prev Jam : 21
Application : 1
Is active : 0
```

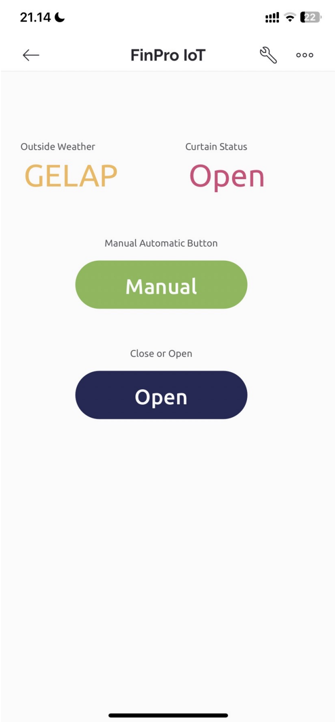
```
Button State: 1
Button Open Pressed
Opening Motor
LDR Value : 2383
Jam : 21
Prev Jam : 21
Application : 1
Is active : 0
```

C. Blynk



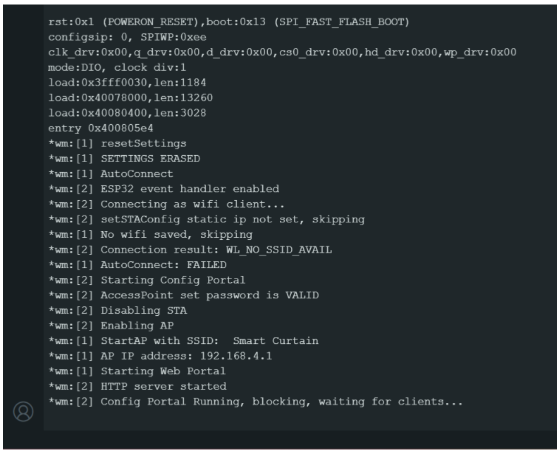


Because testing is done after 6 PM, it will automatically closed.

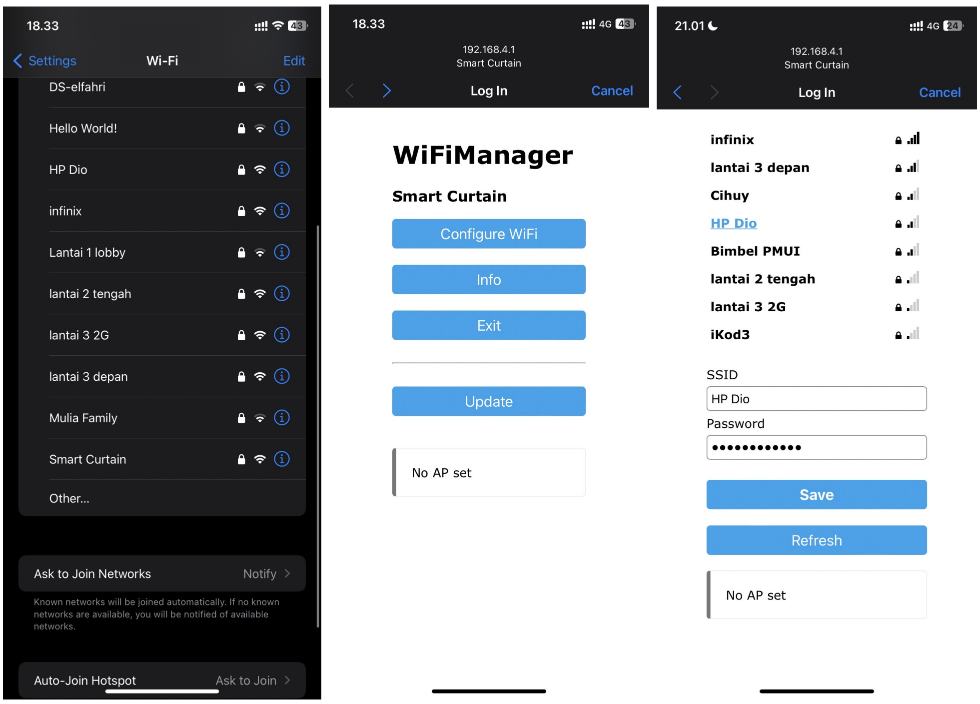


D. ESP32

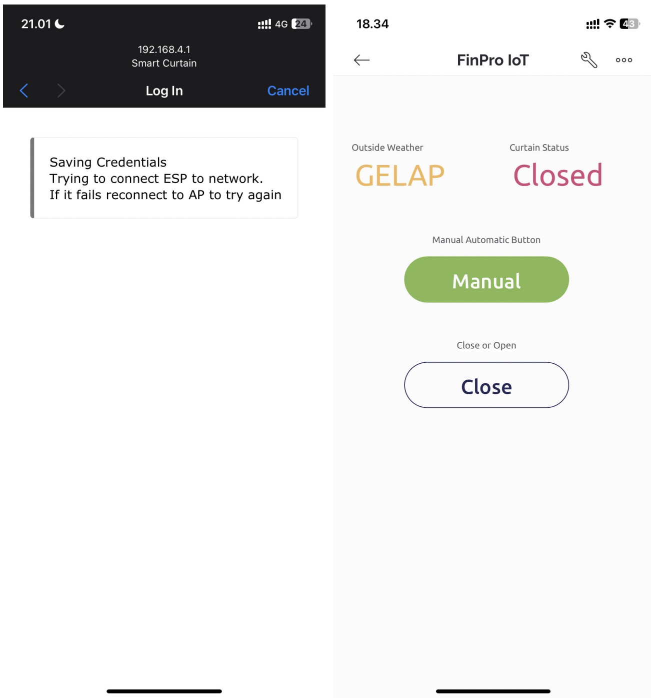
- Before connected to WiFi:



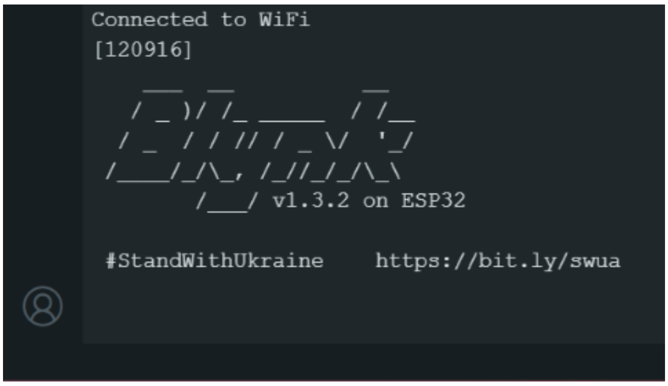
- Connecting to WiFi



To connect the Smart Curtain to WiFi, we need to first establish a connection from our smartphone to the Smart Curtain's WiFi. Subsequently, we will automatically access the WiFiManager page where we can choose which WiFi network to connect with the Smart Curtain.



- Connected to WiFi



3.3 EVALUATION

After successfully completing the testing phase for the Smartloom Curtain Manager device, the evaluation stage becomes essential to assess its overall performance, reliability, and user-friendliness. This section delineates the primary criteria utilized to gauge the system's effectiveness, taking into account both technical functionality and user experience.

The assessment commences by concentrating on the precision and responsiveness of the sensors integrated into the Smartloom Curtain Manager device. The accuracy of the LDR sensor is evaluated by comparing its readings with actual conditions to verify that the sensor furnishes dependable data, ensuring the smooth operation of automation features.

The user interface, exemplified by the Blynk application, constitutes a pivotal element impacting user satisfaction. Input from users engaging with the system is gathered to appraise the lucidity and user-friendly nature of the presented information. The objective is to guarantee that users can easily comprehend real-time sensor readings and navigate the interface efficiently.

The resilience of the hardware and software integration is scrutinized during extended usage. The system undergoes exposure to diverse environmental conditions to pinpoint potential issues or vulnerabilities. This evaluation aims to assure that the Smartloom Curtain Manager device remains consistently operational over time, delivering a dependable solution for continuous monitoring.

CHAPTER 4

CONCLUSION

In summary, the Smart Curtain project offers an advanced and unified solution for the automated and manual control of curtains. Utilizing the capabilities of the ESP32 microcontroller, LDR sensor, and Blynk, the system captures light data and provides relevant information to users. The seamless integration with the Blynk platform enhances the user experience by allowing remote access and control through a user-friendly application. The code's modular design, organized into various FreeRTOS tasks, ensures efficient parallel execution and scalability.

A notable feature of the Smart Curtain is the incorporation of a remotely controlled DC motor for opening or closing the curtains. The system also employs a light sensor to autonomously adjust the curtains based on ambient light conditions. Moreover, it automatically opens the curtains at 6 a.m. and closes them at 6 p.m.

Smart Curtain showcases a comprehensive and adaptable solution for a diverse user base, embodying a well-balanced fusion of sensor technology, remote accessibility, and automation. This project serves as a testament to the potential of integrating hardware and software components to create a sophisticated and user-friendly system for the automatic control of curtains.

REFERENCES

[1] S. PAUL, "Smart Curtains using ESP32 and Blynk IoT," YouTube. Oct. 25, 2021. Accessed: Dec. 10, 2023. [YouTube Video]. Available: https://www.youtube.com/watch?v=9ebqP7wjTLk

[2] Hubitat Inc, "[Release] HubDuino v1.1.9 - Hubitat to Arduino / ESP8266 / ESP32 / ThingShield Integration (ST_Anything)," Hubitat, Mar. 04, 2021. https://community.hubitat.com/t/release-hubduino-v1-1-9-hubitat-to-arduino-esp8266-esp32-thingshield-integration-st-anything/399/1883?page=94 [Accessed Dec. 5, 2023].

[3] M. Fiala, "Building a Smart Curtains System with the ESP32 - The ESP Journal," Medium, Nov. 24, 2022. https://blog.espressif.com/smart-curtains-system-150fcbf853d2 [Accessed Dec. 5, 2023].

[4] "eg321/esp32-motorized-roller-blinds: Control your motorized blinds with Web interface/ MQTT. Easy integration with automation systems like Home Assistant and OpenHab. 3d printing model for Leroy Merlin blinds.," GitHub, Dec. 11, 2021. https://github.com/eg321/esp32-motorized-roller-blinds [Accessed Dec. 5, 2023].

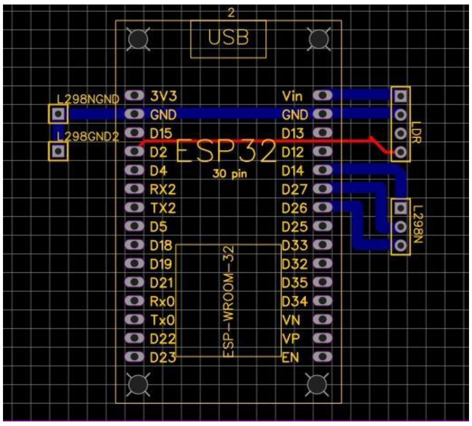
[5] Instructables, "DIY Motorized WiFi Roller Blind - ESP8266 & Blynk," Instructables, Jun. 12, 2022. https://www.instructables.com/DIY-Motorized-WiFi-Roller-Blind-ESP8266-Blynk/ [Accessed Dec. 5, 2023].

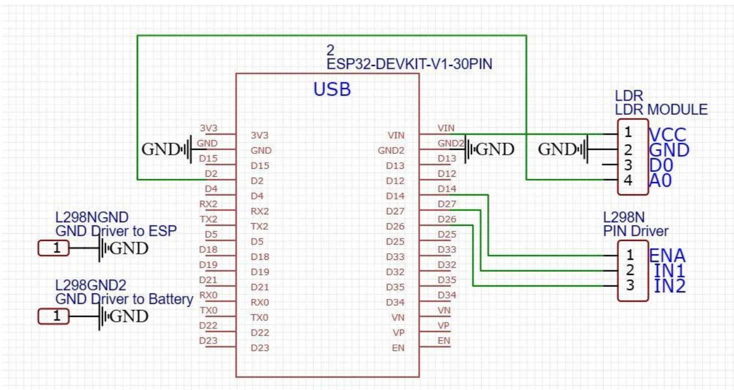
[6] "SMART CURTAIN WITH BLYNK - Wokwi ESP32, STM32, Arduino Simulator," Wokwi.com, 2023. https://wokwi.com/projects/358343439880224769 [Accessed Dec. 10, 2023].

[7] "Build DIY Smart Curtains using ESP32 and LDR Sensors," Circuitdigest.com, 2023. https://circuitdigest.com/microcontroller-projects/diy-smart-curtains-using-esp32-and-ldr-sensors [Accessed Dec. 10, 2023].

APPENDICES

Appendix A: Project Schematic





Appendix B: Documentation