



An-Najah National University
Faculty of Engineering & Information Technology
Department of Computer Engineering
Graduation Project II



chrono clock

Mira Bashar Jamous & Tala Hamad

Supervisor: Dr. Sufyan Samra

Presented in partial fulfillment of the requirements for the
Bachelor degree in Computer Engineering.

June, 2024

Acknowledgement

In the name of Allah, the Most Gracious, the Most Merciful.

We are filled with gratitude for the blessings that have guided us to complete this hardware project. We express our deep appreciation to the Almighty for bestowing upon us the strength, skills, and understanding necessary for this achievement. Our sincere thanks go to our supervisor, Dr. Sufyan Samra, for his invaluable guidance and unwavering support, which significantly shaped the trajectory and success of our project. His expertise and insight have been instrumental in leading us to a result we are proud to present. We are also thankful to the esteemed faculty members of the Computer Engineering Department for their valuable contributions that have enriched our academic journey. Our heartfelt gratitude extends to our families and friends, whose constant encouragement and belief in our abilities have been a source of strength and motivation. Your support has been a pillar throughout this journey. We hope that our efforts and the knowledge we have gained will serve our community and contribute positively to humanity.

Disclaimer

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

Table of Contents

- Acknowledgement I
- Disclaimer II
- Table of Contents III
- List of Figures V
- Abstract VII

- 1 Introduction 1**

- 2 Constraints & Earlier coursework 2**
 - 2.1 Limitations & Constraints 2
 - 2.1.1 Limited Time 2
 - 2.1.2 Limited Resources 2
 - 2.1.3 Safety 2
 - 2.2 Earlier coursework 3

- 3 Literature Review 4**

- 4 Methodology 5**
 - 4.1 Standards and specifications 5
 - 4.2 Hardware Components 5
 - 4.3 Implementation 15
 - 4.3.1 Overview of the system 15
 - 4.3.2 POV Display 15
 - 4.3.3 Modes 16

- 5 Results and Analysis 21**

- 6 Conclusion and Future Work 22**

| | | |
|-----|-----------------------|-----------|
| 6.1 | Conclusion | 22 |
| 6.2 | Future Work | 22 |
| | References | 24 |

List of Figures

| | | |
|------|---------------------------------------------------------|----|
| 4.1 | Arduino Mega 2560 | 6 |
| 4.2 | ESP32-WROOM-32 | 6 |
| 4.3 | DC Motor With Coupler | 7 |
| 4.4 | RGB LED | 7 |
| 4.5 | Resistors | 8 |
| 4.6 | Shift Registers | 8 |
| 4.7 | Real Time Clock | 9 |
| 4.8 | Hall Magnet Sensor Module | 9 |
| 4.9 | Rechargeable Lithium Batteries With Holder | 10 |
| 4.10 | Step Down Buck Converter Module | 10 |
| 4.11 | Toggle Switch | 10 |
| 4.12 | 4X4 16-Key Numeric Keypad | 11 |
| 4.13 | Piezo Buzzer | 11 |
| 4.14 | Mg811 CO2 Sensor | 11 |
| 4.15 | DHT11 Digital Temperature and Humidity Module | 12 |
| 4.16 | Ultrasonic Sensor Module HC-SR04 | 12 |
| 4.17 | UNI-T UTP3303 DC Power Supply | 12 |
| 4.18 | Other Components Used In The Project | 13 |
| 4.19 | Comparison of Different Views of the Design | 13 |
| 4.20 | Comparison of Different Views of the Design | 14 |
| 4.21 | POV Board | 14 |
| 4.22 | Digital Character Representation | 15 |
| 4.23 | Digital Mode | 16 |
| 4.24 | Analog Mode | 16 |
| 4.25 | Air Quality Mode | 17 |
| 4.26 | Pomodoro Figures | 18 |
| 4.27 | Reminder Figures | 19 |
| 4.28 | Notes Figures | 19 |

| | | |
|------|---------------------------------|----|
| 4.29 | Settings Screen | 20 |
| 4.30 | Clock Settings Screen | 20 |
| 4.31 | Color Settings Screen | 20 |

Abstract

Workplace wellness and productivity are essential in modern living. ChronoClock is a Smart Propeller Clock designed to improve these areas by integrating advanced hardware and innovative features.

ChronoClock uses a high-speed DC motor to spin 16 RGB LEDs in a propeller configuration, creating a unique and engaging time display. It includes a Real-Time Clock (RTC) module for precise timekeeping. The clock also has sensors to monitor CO2 levels, temperature, and humidity, providing real-time information about air quality. Motion sensors track inactivity and encourage regular breaks using the Pomodoro Technique to promote healthier focusing habits. A keypad and a mobile application allow users to customize the clock's settings and appearance.

ChronoClock enhances user interaction with its dynamic time display and real-time environmental monitoring. It helps maintain a healthy workspace by tracking air quality. The reminder and break features support continuous productivity and well-being. The keypad and mobile app enable users to easily personalize settings and features.

ChronoClock is a significant improvement for modern living or workplaces, with advanced hardware with user-friendly design combined. This project shows how integrating wellness and productivity features into everyday tools can improve environments, making them healthier and more efficient.

Chapter 1

Introduction

Well-being and productivity are critical in modern environments. Traditional timekeeping devices lack the interactivity and multi-functionality needed to support these goals. ChronoClock is a Smart Propeller Clock designed to enhance productivity and comfort by integrating advanced hardware and innovative features.

The primary objectives of ChronoClock are to improve time management by providing both analog and digital clock displays using a high-speed DC motor and RGB LEDs. It aims to enhance environmental awareness through real-time monitoring of CO2 levels, temperature, and humidity. Additionally, ChronoClock promotes healthier habits by incorporating motion sensors that encourage regular breaks using the Pomodoro Technique. Furthermore, it offers customization options, allowing users to personalize settings and appearance via a keypad and a mobile application.

ChronoClock addresses the growing need for interactive and multi-features tools. By integrating timekeeping, environmental monitoring, and productivity enhancement features, ChronoClock meets the demand for healthier and more efficient environments. The device's ability to improve air quality awareness and promote regular breaks is particularly relevant given the increasing focus on a balanced lifestyle.

This report begins with addressing the Introduction (Chapter 1), then moves on to addressing constraints, challenges, and relevant coursework that shaped the project (Chapter 2). A literature review (Chapter 3) delves into existing research, establishing a foundation for the project's context. The methodology (Chapter 4) details the systematic approach, development processes, and tools used. Results and analysis (Chapter 5) are then presented, offering data interpretation and comparisons to benchmarks. The project wraps up with a summary of key findings and potential avenues for future work (Chapter 6).

Chapter 2

Constraints & Earlier coursework

2.1 Limitations & Constraints

2.1.1 Limited Time

Working on Chrono Clock was hard because we had limited access to the university's electrical workshop. We could only use it from 8:00 AM to 3:30 PM, Sunday to Thursday. This schedule gave us a limited time to get things done, which was challenging given the complexity of the project. The limited hours meant we had to carefully plan our time to meet our deadlines. And sometimes caused delays in our progress.

2.1.2 Limited Resources

We faced several challenges with the resources needed for the Chrono Clock project. First, we needed a high-speed DC motor with a very high RPM, which wasn't easily available in our city. This meant we had to spend extra time searching for it, which slowed us down. On top of that, the high-speed motor required a specialized power supply that was not available in our university's electrical workshop. We had to borrow one from another lab in our department, and the whole process took a lot of time and effort.

After we got the basic structure of the Chrono Clock built, we moved on to designing and laying it out in Eagle for PCB. Unfortunately, the university only had access to the free version of the software, which did not support all the libraries we needed. This version limitation made it really hard for us to progress with the design, and we got stuck several times trying to find workarounds.

We also wanted to add an O₂ sensor to detect oxygen levels and display them, but unfortunately, the sensor was not available in our city. Additionally, we had to buy a CO₂ sensor, which was quite expensive, adding to the overall project cost.

2.1.3 Safety

One of our biggest safety concerns was when the board of the Chrono Clock was spinning at high speeds. We were really worried that the board might come loose and fly off from the coupler of the DC motor. This could have been dangerous, so we had to take extra precautions to make sure everything was secure and safe.

2.2 Earlier coursework

Our university coursework guided us through our learning journey and provided us with a solid foundation of knowledge, which was crucial in the development of Chrono Clock. Courses in hardware engineering, like Digital Circuits Design I and II, Microcontroller Programming, and Image Processing, gave us the essential skills and understanding needed for this project.

In the Digital Circuits Design courses, we learned about the fundamentals of circuit design and how to work with various electronic components, which helped us design the hardware for Chrono Clock. The Microcontroller course taught us how to program and control microcontrollers, which was critical for managing the different features of our clock, such as the real-time clock (RTC) module and sensors. The Image Processing course, although not directly related to the clock's functionality, helped us understand how to manipulate and display data, which was useful for creating the dynamic LED display.

Chapter 3

Literature Review

The Rotating Display project (lhm0, 2024) is a compact disc-sized device that utilizes a CD motor to quietly rotate 40 LEDs, displaying time and weather data sourced from the internet and controlled wirelessly via a web interface using Arduino Nano and ESP-01s microcontrollers. This project is significant because it illustrates how rotational motion can be used to create a dynamic display, a key feature of Chrono Clock. It also emphasizes the importance of user-friendly design with wireless control, a concept we adopted in Chrono Clock by incorporating real-time environmental monitoring and customization through a mobile application.

Similarly, the article POV Persistence of Vision Display Using Arduino Nano by Prachethi Reddy (Reddy, 2017) explores the persistence of vision (POV) concept, where eight LEDs spin rapidly to create the illusion of a continuous message due to the way our brain processes images. The article underscores the necessity of having a motor with a speed of at least 1000 RPM to avoid flickering, which was a challenge we encountered in ensuring our high-speed DC motor was capable of displaying clear images on Chrono Clock.

These resources provide a foundation for understanding the creation of rotational LED displays. The Rotating Display project showcased the practicality of using rotational motion to present information dynamically, while the POV display highlighted the technical requirements needed to prevent flickering and ensure smooth visual output.

In developing Chrono Clock, we drew on these principles to design a device that integrates advanced features like real-time environmental monitoring with a visually appealing and user-friendly interface. The challenges of sourcing appropriate motors and sensors, and ensuring reliable performance at high speeds, were informed by the insights gained from these earlier projects.

Chapter 4

Methodology

4.1 Standards and specifications

In our project, we have written 3 main pieces of code which are for the Arduino Mega, main design ESP32 and the second ESP32 for the wireless communication. The codes were implemented using the Arduino IDE in C++ language which includes several libraries such as DHT11.h, Keypad.h, CO2Sensor.h, WiFi.h, WiFiClient.h and RtcDS1302 and many other libraries that helped to implement the required functionalities. The mobile application was developed using App Inventor and its very own Web service.

4.2 Hardware Components

In this section, we will discuss the hardware components we utilized and the reasons for their selection:

- **Arduino Mega 2560**

The Arduino Mega 2560 is a flexible and powerful microcontroller board, an extended version of the popular Arduino Uno. It features the ATmega2560 microcontroller, providing 54 digital I/O pins, 16 analog inputs, and sufficient memory.

In our project, we use the Arduino Mega 2560 to connect to two essential sensors: the DHT11 sensor, and the CO2 sensor. These sensors record temperature, humidity and CO2 levels data, which the Arduino Mega 2560 processes and formats. The data is then sent to the ESP32 via the TX/RX serial connection. The ESP32 handles further processing and displays the information.

We started with the Arduino Uno, but as our project grew, we needed faster serial communication speeds since our communication operates on a 115200 baud rate. That's why we chose the Arduino Mega 2560. Mega's faster serial communication speed compared to the Uno ensures our data transfers are quick and reliable, which is crucial for our project's performance.



Figure 4.1: Arduino Mega 2560

- **ESP-WROOM-32**

We used two ESP32 microcontroller modules. The first ESP32 is part of the main setup where it controls shift registers which in return control the RGB LEDs connected to resistors, and handles the POV display. It communicates with the second ESP32 using WiFi to get data from sensors like temperature, humidity, CO2 levels, ultra sonic value, and the mode selection inputted by the user from a keypad. This ESP32 processes this data and shows the relevant information. It also processes the incoming requests from the mobile app and displays the corresponding mode specified in the request.

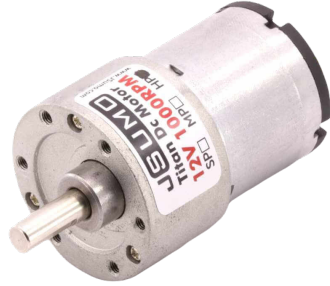
The second ESP32 communicates through serial communication with the Arduino Mega. Its main job is to get CO2, temperature, and humidity data from the Arduino Mega, which it uses for display or further processing.



Figure 4.2: ESP32-WROOM-32

- **DC Motor**

We used a 12-volt DC motor capable of operating at 1000 RPM (Power, 2022), and paired it with a DC motor coupler glued to the project so it can spin it and optimized it to spin at 840 RPM.



((a)) DC Motor 12 Volt 1000 RPM



((b)) DC Motor Coupler

Figure 4.3: DC Motor With Coupler

- **RGB LEDs**

We used 16 common cathode RGB LEDs for our POV Clock as an upgrade to the previously done POV clocks which usually use one color-coded LEDs, which provides customizability to the clock so that it meets the user's needs and preferences.



Figure 4.4: RGB LED

- **Resistors**

We used 48 100Ω resistors for the green and blue pins and 16 150Ω resistors for the red pins of the used RGB LEDs. According to the data-sheet (Pro, 2021) we measured the appropriate resistors.

$$R_{red} = \frac{5V - 2.0V}{0.02A} = 150\Omega \quad (4.1)$$

$$R_{green} = \frac{5V - 3.0V}{0.02A} = 100\Omega \quad (4.2)$$

$$R_{blue} = \frac{5V - 3.0V}{0.02A} = 100\Omega \quad (4.3)$$



((a)) 150 Ω Resistor



((b)) 100 Ω Resistor

Figure 4.5: Resistors

- **Shift Registers(74hc595)**

Since we have a large number of LEDs, and each one requires 3 connections, we need a total number of 48 micro-controller pins. It is not possible to get this many outputs directly from an ESP32. To address this, we used shift registers, which allowed us to control the LEDs with as few ESP32 outputs as possible.

Each shift register has an 8 outputs capacity which we used to 8 RGB LED pins, arranged in the order (RGBRGBRG) for the first shift register and so on. Therefore, for controlling 16 RGB LEDs, we used 6 shift registers to get the required 48 outputs.

$$\#_{Pins} = 8 * 16 = 48 \quad (4.4)$$

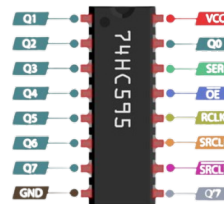
$$\#_{ShiftRegisters} = \frac{48}{8} = 6 \quad (4.5)$$

We connect the output enable pins to the ground (since they are active low). The storage register clock input (RCLK), serial data pin (DS), and shift register clock input (SRCLK) are connected to the ESP's digital pins designated for clock, data, and latch pins.

To chain the registers together, we connect the QS7 (serial output) of the first register to the SER (serial input) of the next register, and this goes on for all subsequent registers.



((a)) Shift Registers



((b)) Shift Register Pinout

Figure 4.6: Shift Registers

- **Real Time Clock Module RTC(DS1302)**

An RTC module was used to fetch information about time and date so that it can be displayed on the POV clock correctly and in sync. The RTC module has its own clock source, a quartz crystal oscillator, which provides a stable timing reference independently of the microcontroller. Once the Arduino C code initializes the RTC module upon upload, it sets an initial time reference which the module continues to track using its built-in battery.

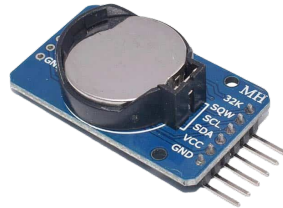


Figure 4.7: Real Time Clock

- **Hall Effect (Magnet) Switch Sensor Module**

To effectively manage the system, it is essential to monitor the motor's speed and to detect when it completes a full rotation. The Hall Effect sensor was a perfect fit due to its ease of use and high-speed performance.

In our project, we have used the Hall Effect sensor module, which is attached to the POV clock board, while a magnet is placed on the design, maintaining a small gap between the two. As the board spins, the sensor detects the magnet's position when it reads a HIGH value on the corresponding connected ESP32 Hall Effect Sensor digital pin for each complete full rotation.

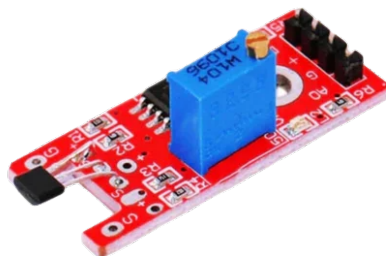


Figure 4.8: Hall Magnet Sensor Module

- **Rechargeable Lithium Batteries**

We used Rechargeable 3.7 lithium batteries (EEMB Co., Ltd., 2010) to power up the ESP32 via the VIN pin, which in return powers the RGB LEDs, RTC Module, Hall Magnetic Sensor module and the shift registers through the VCC pin.



((a)) Rechargeable Lithium Batteries

((b)) Battery Holder

Figure 4.9: Rechargeable Lithium Batteries With Holder

- **LM2596 Adjustable Step Down Buck Converter Module**

We used the buck converter module to convert 7.4 volts (outputted by the lithium batteries) to 5 volts which is the required voltage to power on the ESP32, RGB LEDs, RTC Module, Hall Magnetic Sensor module and the shift registers.



Figure 4.10: Step Down Buck Converter Module

- **Toggle Switch**

We used a toggle switch to turn off and on the POV Clock by connecting one side to the ground of the batteries and the other to the ground of the buck converter.



Figure 4.11: Toggle Switch

- **4X4 16-Key Numeric Keypad**

We have used a 4x4 keypad and interfaced it with the second ESP32 which is used to select features to utilize on the POV Clock. Once the user pressed on a button on the Keypad the ESP32 registers the selection and sends it through the WiFi module to the main ESP32, which in return processes the data and displays the correct query.



Figure 4.12: 4X4 16-Key Numeric Keypad

- **Piezo Buzzer**

We used a Piezo Buzzer and interfaced it with the second ESP32 to issue an alarm sound whenever the timer was up in the Pomodoro or Reminders features which will be discussed later on.



Figure 4.13: Piezo Buzzer

- **Mg811 CO2 Sensor**

After better understanding for the health effects and safety guidelines related to carbon dioxide (CO₂) exposure. This inspired us to use the MG811 CO₂ sensor, to ensure safe indoor air quality monitoring by detecting CO₂ levels and alerting the user about the current air quality status.

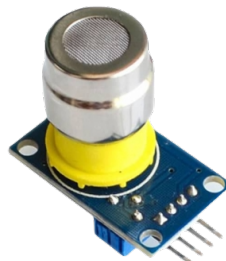


Figure 4.14: Mg811 CO2 Sensor

- **DHT11 Digital Temperature and Humidity Module**

The DHT11 was used for its simplicity, reliability, and affordability, providing accurate data about the temperature and humidity, which was ideal for our application.

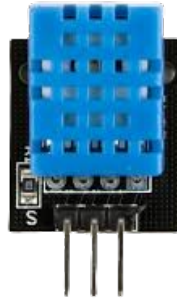


Figure 4.15: DHT11 Digital Temperature and Humidity Module

- **Ultrasonic Sensor Module HC-SR04**

We have used the ultrasonic sensor to keep track of the user's physical activity, what it does is measure the distance between the design and the user's position. If the user has been inactive for a long time there will be procedures to be taken which will be discussed later on.



Figure 4.16: Ultrasonic Sensor Module HC-SR04

- **Power Supply**

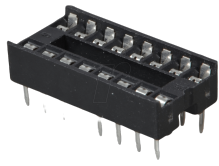
We specifically chose this power supply to power the DC motor because it needs to provide sufficient torque to spin the motor at high speed, even with the additional weight of the POV clock board, lithium battery, and other components on the board. The increased torque is essential to overcome the weight of these components and maintain the desired motor speed.



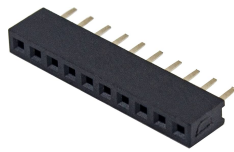
Figure 4.17: UNI-T UTP3303 DC Power Supply

- **Other Components**

IC-SOCKET-18-pins, IC-SOCKET-30-pins, wires, soldering board



((a)) 16 Pin IC Socket



((b)) ESP32 Socket



((c)) Wires



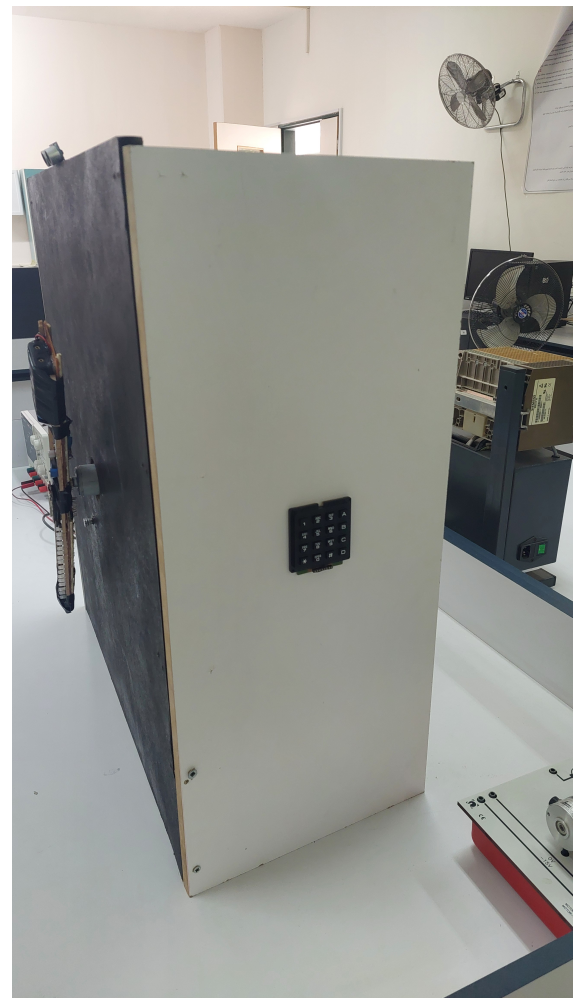
((d)) Copper Clad

Figure 4.18: Other Components Used In The Project

- **Final Design**

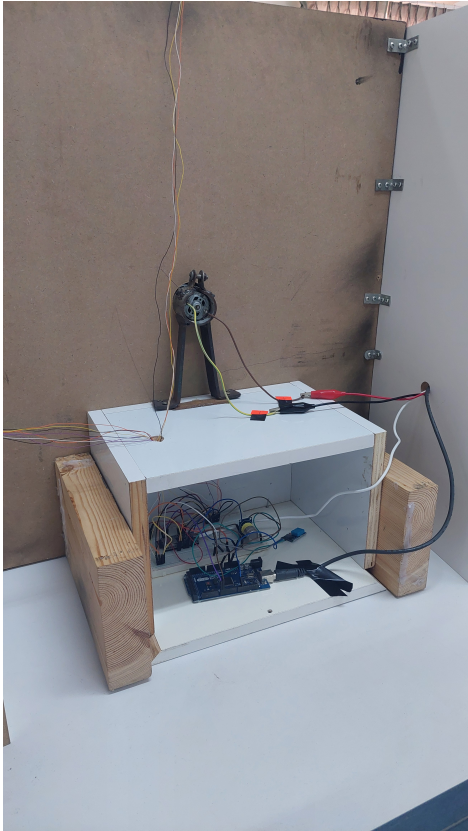


((a)) Front Of Design

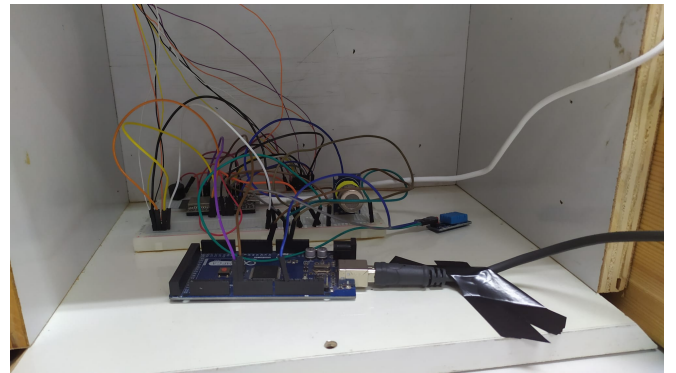


((b)) Side Of The Design

Figure 4.19: Comparison of Different Views of the Design



((a)) Back Of Design



((b)) Arrangement Of The Components

Figure 4.20: Comparison of Different Views of the Design

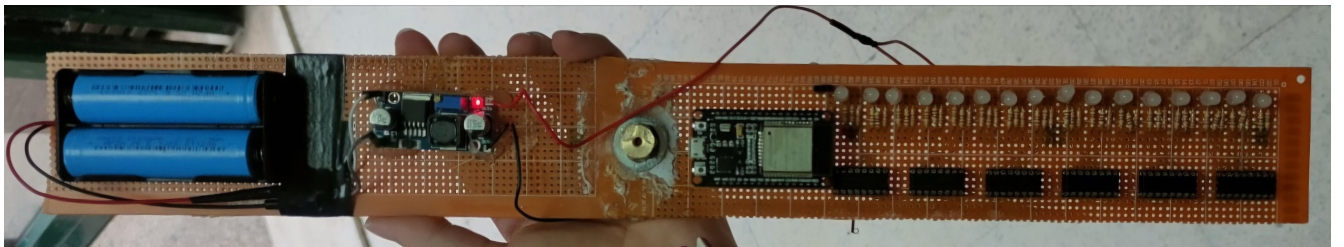


Figure 4.21: POV Board

4.3 Implementation

4.3.1 Overview of the system

The Chrono Clock system is designed to provide a fun user experience, beginning with the power-up phase. Once the power supply is turned on, the DC motor is activated and speeds up to a specific speed. At this point, the system is ready to receive user input via a 4x4 keypad. When the user presses the number "1" on the keypad, the system enters Menu Mode, which displays the project name, "Chrono Clock," for 3 seconds, as an initial greeting. Following this, the system presents a menu that guides the user on how to move through different modes. The menu provides options such as "PRESS 1 for Menu" to re-visit the main menu, "PRESS 2 for Digital Hour" to display the current time in a digital format, "PRESS 3 for Analog Hour" to show the time with an analog clock representation, and "PRESS 4 for Air Quality" to access air quality data from the CO2 and DHT11 digital temperature and humidity sensor. Additionally, users can explore more features by visiting the mobile application, each option in the menu will display for 3 seconds too.

4.3.2 POV Display

The POV (Persistence of Vision) display in the Chrono Clock system utilize RGB LEDs controlled by the main ESP32 on the board to create dynamic effects and display characters, symbols, and strings. In general, when display a character, the function will identify the character display position (the upper part of the display or the lower part), then lights up LEDs at in a specified period with delays between each vertical strip of the character's digital representation.

| | | | | | | | | | |
|----|--|--|--|--|--|--|--|--|----|
| 10 | | | | | | | | | 10 |
| 9 | | | | | | | | | 9 |
| 8 | | | | | | | | | 8 |
| 7 | | | | | | | | | 7 |
| 6 | | | | | | | | | 6 |
| 5 | | | | | | | | | 5 |
| 4 | | | | | | | | | 4 |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

Figure 4.22: Digital Character Representation

4.3.3 Modes

Digital Mode

In Digital Mode, Chrono Clock displays the current time, day of the week, and date. The current time is shown in the upward direction in a format 'hh:mm'. Additionally, the day of the week is displayed in a three-letter format, such as 'MON' for Monday, Then the current date is displayed in a 'dd/MM' format.



Figure 4.23: Digital Mode

Analog Mode

In this mode we display the current time in hour and minutes clock hand format fetched from the RTC Module. It was implemented by using four main functions `minutesHand`, `hourHand`, `sharpHourHand` and `verticalLine`.

1. Minutes Hand: the `minutesHand` specifies where and when to light up the strip of LEDs where the minute should be located in an analog clock.
2. Hour Hand: the `hourHand` specifies where and when to light up the strip of LEDs where the hour should be located in an analog clock.
3. Sharp Hour Hand: the `sharpHourHand` specifies where and when to light up the strip of LEDs where the hour and minutes are on the same location, for example when the hour is 10 and the minutes are 50.
4. Vertical Lines: the `verticalLine` specifies where to light up the hour marks from 1-12.

The 4 functions are incorporated together with specified delays to create the analog clock.



Figure 4.24: Analog Mode

Air Quality Mode

In Air Quality Mode, it displays the current CO2 level (of Health Services, 2023), categorizing it as either 'Poor' or 'Good' based on the ppm thresholds detected by the CO2 sensor. Additionally the mode displays temperature and humidity readings sent from the DHT11 sensor.



Figure 4.25: Air Quality Mode

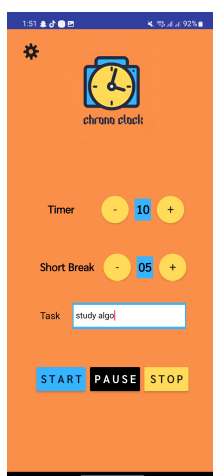
Pomodoro Mode

The Pomodoro Technique (Pomofocus, 2017) is a time management method designed to improve productivity and focus by breaking work into intervals, traditionally 25 minutes in length, separated by short breaks. This technique helps maintain sustained concentration and prevents burnout.

In our project, the Pomodoro Technique is implemented using the mobile application where the main ESP32 receives 3 values from the application integrated into the ChronoClock system to enhance productivity and encourage healthier work habits. Here's how it works:

1. **Timer Period:** Users can set a timer ranging from 1-60 minutes. During this period, the ChronoClock's display shows the countdown, allowing users to stay aware of their remaining time.
2. **Break Period:** After the Timer Period ends, the ChronoClock signals the user with a buzzer sound, which is issued by sending a response to the second ESP32 that processes it and turns on the buzzer alert, and a notification on the clock. Users are then prompted to take a break ranging from 1-60 minutes based on the user's preference. The display will countdown this break period, helping users to rest effectively before the next focus session.
3. **Note:** User can enter a specific task that they want to get done during the timer period.
4. **Command:** The User can send 3 commands which are: "Start" to begin the timer, "Resume/Pause" which can resume or pause the timer or the short break, and finally "Stop" which terminates the session completely.

By incorporating the Pomodoro Technique, ChronoClock not only helps users manage their time more effectively but also promotes regular breaks, reducing mental fatigue and improving overall productivity.



((a)) Pomodoro Application Page

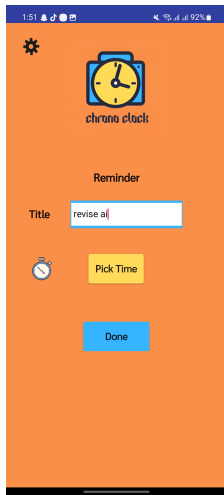


((b)) Pomodoro Timer in Action

Figure 4.26: Pomodoro Figures

Reminders Mode

We have implemented a reminder functionality where the user sets a title for the reminder and sets the time when they want to be notified. The application sends two parameters to the main ESP32 which are **Title** and **The Picked Time**. The microcontroller keeps checking for when the current hour and minute equal the specified reminder hour and minute then a buzzer is issued in the same way the pomodoro buzzer [1] is.



((a)) Reminder Application Page

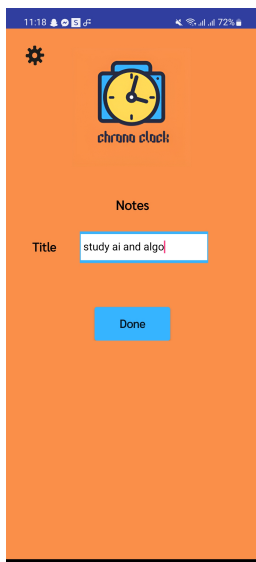


((b)) Reminder In Action

Figure 4.27: Reminder Figures

Notes Mode

We have implemented the notes functionality where the user sets a note to be displayed on the POV Clock. The application sends the note to the main ESP32 and it will



((a)) Notes Application Page



((b)) Notes Screen

Figure 4.28: Notes Figures

Settings

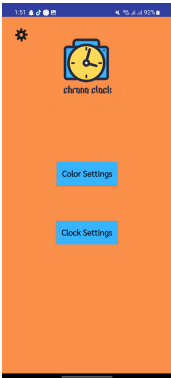


Figure 4.29: Settings Screen

- 1. Clock Settings: To make the app more customizable we have implemented this functionality where the user is able to set a custom time and change its format (12 or 24) and can reset to the default time settings using the default button

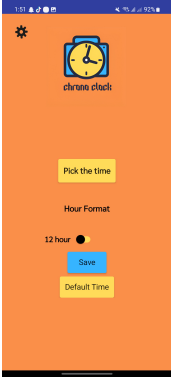


Figure 4.30: Clock Settings Screen

- 2. Color Settings: In order to bring the productivity spirit to the environment a section where the user can change the text, inner border and outer borders colors using a color picker was implemented



Figure 4.31: Color Settings Screen

Chapter 5

Results and Analysis

By the time this project comes to a finish, we have already created an integrated system that offers the primary capabilities we were expecting.

- **User Engagement:** The dynamic time display created by the DC motor and RGB LEDs displaying various colors was made time tracking engaging and visually appealing.
- **Customizable User Experience:** The keypad and mobile application allowed users to personalize the settings and appearance. the ability to customize the clock to their preferences, enhancing the overall experience.
- **Improved Air Quality Awareness:** The sensors monitoring CO2 levels, temperature, and humidity provided real-time data on air quality, which will lead users to take actions like opening windows of the room or using a humidifier for high humidity levels. This leads to a healthier work environment, reducing the risk of issues related to poor air quality.
- **Enhanced Productivity and Wellness:** The integration of the Pomodoro Technique increased productivity of ChronoClock users. They reported more effective focus periods and better breaks, leading to improved mental health and reduced burnout.

We learned the importance of effective integration between hardware components and software systems. Combining a DC motor, RGB LEDs, and multiple sensors with custom software enabled us to create a dynamic time display. This project demonstrated the necessity of understanding both hardware specifications and software development to achieve cohesive functionality

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Chrono Clock is a great addition to the work space, especially for students, by combining hardware with wellness and productivity features. The results showcases that integrating such features into everyday tools can enhance the quality of the environment, making it healthier and more efficient.

The successful implementation of motion sensors to track inactivity, and the Pomodoro Technique, encourages better focusing habits and healthier consistent breaks, which are essential productivity and well-being. Real-time environmental monitoring helps maintain optimal air quality, further contributing to a healthy workspace.

The dynamic and engaging time display, along with the ease of customization through a keypad and mobile app, enhances user interaction and satisfaction. These make Chrono Clock a user-friendly and effective tool for improving workplace wellness and productivity.

Overall, Chrono Clock stands out as a thoughtful integration of technology, and how it can lead to significant improvements in daily living and work environments, highlighting the potential for future developments in this field.

6.2 Future Work

For future enhancements, ChronoClock can benefit from several improvements.

1. Increasing the speed of the DC motor could lead to a more stable and smoother display, reducing flickering and improving visual clarity.
2. Integrating an OLED screen to display the battery percentage would provide users with information about the device's power status,
3. Creating a 3D display visible from multiple angles, similar to a hologram, could significantly elevate the device's visual appearance and functionality.
4. Integrating voice control can make ChronoClock even more user-friendly.
5. Developing a cloud-based system that allows users to access and control Chrono-Clock remotely via a web interface. This feature would enable users to manage their

schedules, check environmental data, and receive notifications even when they are away from their workspace.

6. In addition to displaying text, ChronoClock can be upgraded to display images. This functionality can be useful for showing motivational images, important reminders, or even environmental data in a more visual colorful and fun representation.

References

- EEMB Co., Ltd. (2010). LIR18650 Datasheet [Accessed: 2024-04-25]. <http://eemb.com>
- Pomofocus. (2017). Pomofocus - a pomodoro timer to boost your productivity [Accessed: 2024-06-27]. <https://pomofocus.io/>
- Reddy, P. (2017). *Pov persistence of vision display using arduino nano* [Medium]. Retrieved March 15, 2024, from <https://medium.com/@prachethire/pov-persistence-of-vision-display-using-arduino-nano-3a0c1a0ac0ba>
- Pro, M. (2021). 5mm round rgb led [Accessed: 2021-09-07]. <https://www.newark.com/multicomp-pro>
- Power, G. (2022). *Dc 12v 1000rpm gear motor specifications* [Greensky Power]. Retrieved December 22, 2022, from <https://greensky-power.com/dc-12v-1000rpm-gear-motor-specifications/>
- of Health Services, W. D. (2023). Carbon dioxide information [Accessed: 2024-06-05].
- lhm0. (2024). *Rotating display* [GitHub]. Retrieved March 3, 2024, from https://github.com/lhm0/rotating_display