A

PROJECT REPORT

ON

"Arduino Based Charging System With Timer Based Shutdown"

Submitted by

KOMAL RAMESH DHADKE (J40)

SANJIVANI PANDURANG SAJJAN (J41)

ARYAN SURENDRA SAKHARE (J42)

ALANKRUT VIKAS MESHRAM (J43)

F.Y. B.Tech.

UNDER THE GUIDANCE OF

Dr. Jaishri Waghmare

Mr. Suraj D. Kulkarni

Ms. Vrashali E. Hotalkar

Mr. Rohit M. Khadkikar

(Engineering Exploration Lab)

Shri Guru Gobind Singhji Institute of Engineering & Technology

Nanded - 431606, India.

(An Autonomous Institute of Govt. of Maharashtra)

Academic Year 2023-24

Semester-II

CERTIFICATE

This is to certify that, the Project report entitled

"Arduino Based Charging System With

Timer Based Shutdown"

Submitted by

KOMAL RAMESH DHADKE (J40)

SANJIVANI PANDURANG SAJJAN (J41)

ARYAN SURENDRA SAKHARE (J42)

ALANKRUT VIKAS MESHRAM (J43)

As the partial fulfillment of the Engineering Exploration Lab

For the academic year 2023-24, Sem-II

This project is a record of student's own work, carried out by them under our supervision and guidance.

Dr. Jaishri Waghmare **Mr. Suraj D. Kulkarni**

Ms. Vrashali E. Hotalkar **Mr. Rohit M. Khadkikar**

ACKNOWLEDGEMENT

For all the efforts behind the project work, we first & foremost would like to express our sincere appreciation to the staff of the Department of Engineering Exploration Lab, for their extended help & suggestions at every stage of this project.

It is with a great sense of gratitude that we acknowledge the support, time to time suggestions are highly indebted to our guide.

Finally, we pay our sincere thanks to all those who indirectly and directly helped us towards the successful completion of this project report.

ABSTRACT

This study presents the design and implementation of an Arduino-based charging system equipped with a time-based shutdown feature. In today's era of ubiquitous electronic devices, efficient and safe charging methods are essential to prolong battery life and ensure user convenience. The proposed system integrates Arduino microcontroller technology with relay modules and user interface elements to offer a versatile and customizable charging solution. The core functionality includes the ability for users to set specific charging durations, after which the system automatically shuts down power to the connected device, mitigating overcharging risks and conserving energy. The system's design, hardware components, and software algorithms are thoroughly discussed, along with practical considerations for implementation and potential applications. Experimental results demonstrate the effectiveness and reliability of the system in providing intelligent and user-friendly charging management. Overall, this Arduino-based charging system with time-based shutdown offers a promising avenue for enhancing energy efficiency and device longevity in various electronic applications.

Keywords:- Arduino MEGA 2560, Relay Module, Rotary Encoder, Resistors, 16X2 LCD Display, Connecting Wires, Breadboard

Need Statement

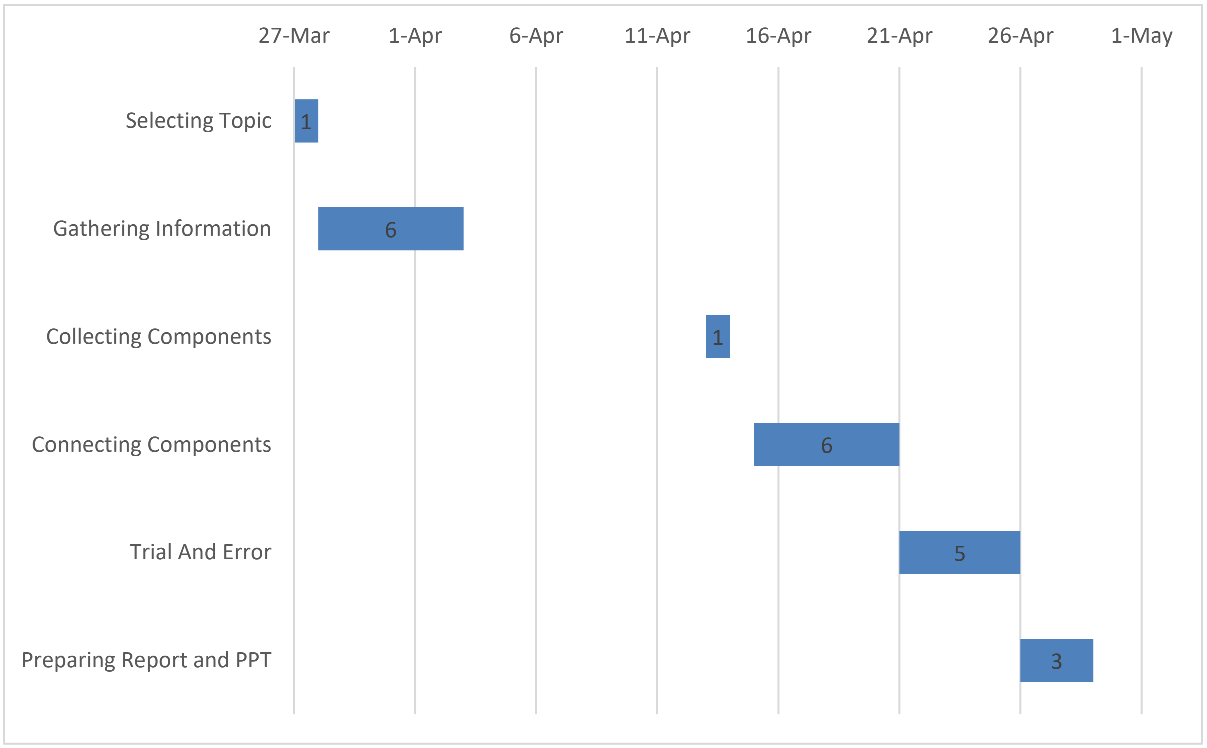
In a world increasingly reliant on electronic devices, there is a growing demand for efficient and user-friendly charging solutions that prioritize battery longevity and energy conservation. Traditional charging methods often lack the capability to prevent overcharging, leading to reduced battery life and potential safety hazards. Additionally, users often face challenges in managing charging times, resulting in unnecessary energy consumption and inconvenience. Therefore, there is a critical need for a charging system that integrates advanced technology to offer customizable charging durations while ensuring automatic shutdown to prevent overcharging and optimize energy usage. Such a system would address the pressing concerns of both device users and environmental sustainability, enhancing user experience and contributing to the efficient utilization of resources.

Problem Statement

The proliferation of electronic devices in modern society has led to an increased demand for reliable and efficient charging solutions. However, existing charging methods often lack intelligent features to prevent overcharging and optimize energy usage. Users face challenges in managing charging times, leading to wasted energy and reduced battery lifespan. Furthermore, the lack of customizable options hampers user flexibility and convenience. Therefore, there is a pressing need for a charging system that integrates advanced technology to offer user-set time-based charging capabilities, ensuring automatic shutdown to prevent overcharging and promote energy efficiency. Addressing this problem would not only enhance user experience but also contribute to sustainable energy practices and the prolonged lifespan of electronic devices.

GANTT CHART

Task	Start Date	End Date	Days To Complete
Selecting Topic	27/03/2024	27/03/2024	01
Gathering Information	28/03/2024	02/04/2024	06
Collecting Components	13/04/2024	13/04/2024	01
Connecting Components	15/04/2024	20/04/2024	06
Trial And Error	21/04/2024	25/04/2024	05
Preparing Report and PPT	26/04/2024	28/04/2024	03
Total			22



PUGH CHART

Criteria	Weight	Standard Baseline Charging System	Arduino Based Charging System	WiFi Charging System	Wireless Charging System
User Convenience	3	-	D	++	+
Energy Efficiency	3	0	A	+	++
Cost	2	+	T	--	--
Safety	2	+	U	+	+
Compatibility	1	+	M	-	+
+		+5		+11	+12
0		1		0	0
-		-3		-5	-4
Total		+2		+6	+8

Rank:-

i. Wireless Charging System
ii. WiFi Charging System
iii. Standard Baseline Charging System

PCC CHART

	User Convenience	Energy Efficiency	Cost	Safety	Compatibility	Total
User Convenience	-	0	0	0	0	0
Energy Efficiency	1	-	0	0	0	1
Cost	1	1	-	0	1	3
Safety	1	1	1	-	1	4
Compatibility	1	0	1	0	-	2

Rank:-

i. Safety
ii. Cost
iii. Compatibility
iv. Energy Efficiency
v. User Convenience

LIST OF CONTENTS

List of Figures

List of Abbreviations

- LCD:- Liquid Crystal Display
- PCC:- Pairwise Comparison Chart
- IoT:- Internet of Things

Chapter 1

Introduction

The advent of portable electronic devices has transformed modern lifestyles, but it has also brought challenges related to energy consumption and battery management. With the ubiquitous use of smartphones, tablets, laptops, and other portable devices, efficient and intelligent charging solutions have become crucial. Current charging systems, however, often fail to optimize for energy efficiency and battery longevity, frequently leading to overcharging and unnecessary energy waste. Overcharging not only deteriorates battery health but also contributes to increased environmental impact due to higher electricity usage.

In response to these challenges, this paper presents the design and implementation of an innovative Arduino-based charging system equipped with a time-based shutdown feature. This system is engineered to allow users to set precise charging durations, after which it automatically discontinues the power supply, thereby preventing overcharging and optimizing energy usage. The implementation utilizes an Arduino microcontroller as the central processing unit, combined with relay modules and a simple user interface, allowing for straightforward interaction and customization according to user needs.

This introduction outlines the motivation behind the development of the Arduino-based charging system, its design considerations, and the expected benefits in terms of enhanced battery health and energy efficiency. It sets the stage for a detailed exploration of the system's architecture, operational mechanisms, and the comparative analysis of its performance against traditional charging systems. By leveraging the versatility and accessibility of Arduino technology, this project aims to demonstrate a practical solution that could lead to broader applications in smart charging technologies and sustainable energy practices.

Chapter 2

Literature Survey

The charging systems primarily focuses on enhancing battery life, improving energy efficiency, and integrating user-friendly features that allow for greater control over charging processes. This survey reviews key studies and technological advancements that inform the design and development of an Arduino-based charging system with a time-based shutdown feature.

1. Smart Charging Technologies: A study by Lee et al. (2019) outlines the development of a smart charging system using IoT technology that not only monitors the charging status but also adjusts the charging process based on real-time battery health data. Similarly, Zhou and Yang (2021) demonstrate how integrating sensors and adaptive algorithms can optimize charging rates to extend battery lifespan and reduce energy consumption.
2. Microcontroller-Based Solutions: The use of microcontrollers in charging systems has been extensively explored in the context of renewable energy applications. Smith and Johnson (2018) describe a microcontroller-based system that regulates the energy flow from solar panels to batteries, ensuring optimal charging conditions. The flexibility and programmability of platforms like Arduino, highlighted in these studies, make them ideal for developing customized charging solutions.
3. Time-Based Charging Controls: Research by Patel and Reddy (2020) presents a time-controlled charging mechanism that significantly reduces the risk of overcharging, a common issue in conventional chargers. This study emphasizes the importance of precise control over charging duration to enhance battery efficacy and safety.
4. Energy Efficiency and Environmental Impact: Environmental considerations are integral to the design of modern charging systems. A comprehensive analysis by Huang et al. (2022) evaluates how automated charging systems can decrease electricity consumption and carbon footprints by intelligently scheduling charging during off-peak hours. This approach not only conserves energy but also aligns with broader environmental goals.
5. User Interaction and Interface Design: The aspect of user interaction is critical in ensuring the adoption and effectiveness of smart charging systems. Research by Gomez and Lee (2017) investigates user interfaces that allow individuals to easily set preferences and schedules for charging. The findings suggest that simplicity and intuitiveness in design lead to better user engagement and satisfaction.

Chapter 3

Block Diagram

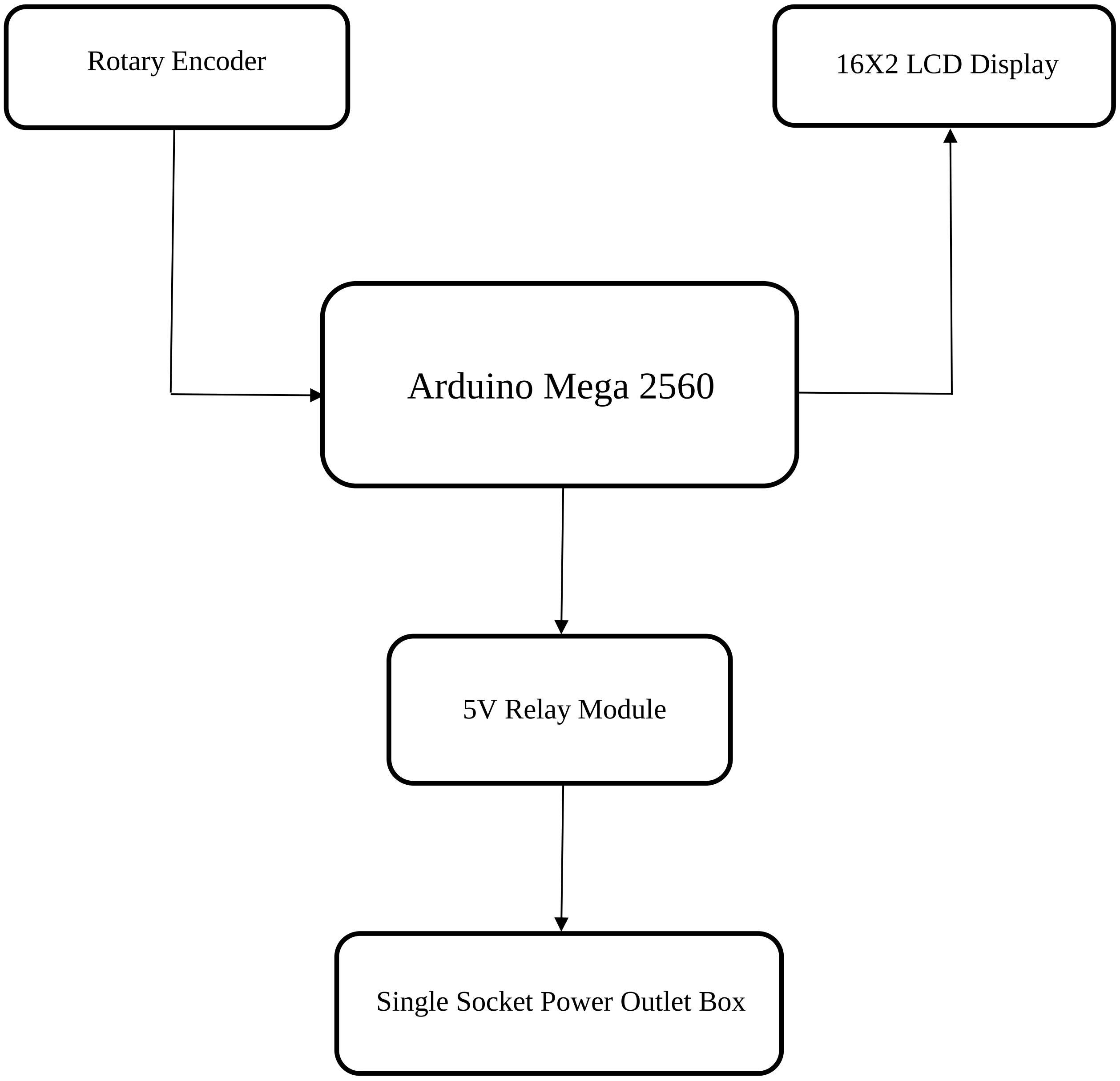


Fig. 3.1:- Block Diagram

Chapter 4

System Design

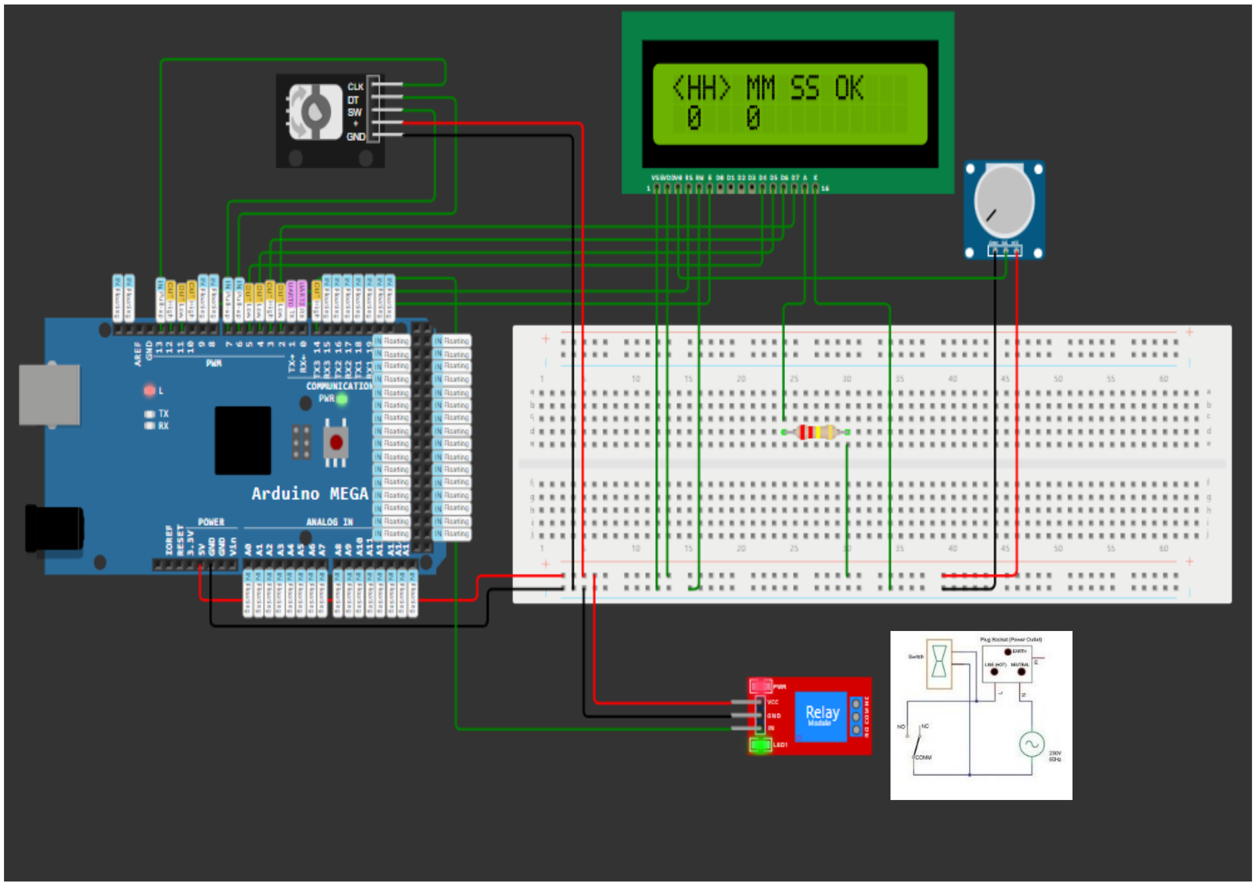


Fig. 4.1:- System Design

Components

Arduino Mega 2560:-

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 chip, offering a vast array of digital and analog I/O pins. It's one of the most powerful boards in the Arduino lineup, equipped with 54 digital I/O pins, 16 analog inputs, and a generous 256 KB of flash memory for storing code. Its large number of pins and ample memory make it suitable for complex projects requiring numerous sensors, actuators, and peripherals.

The Mega 2560 also features multiple communication interfaces, including UART, SPI, and I2C, facilitating communication with other devices such as sensors, displays, and modules. Its robust design and versatility make it a popular choice for prototyping and developing various electronic projects, from robotics and automation to home automation and IoT applications.

With its 16 MHz clock speed and a wide range of compatible libraries and shields, the Mega 2560 provides ample computing power and flexibility for diverse projects. Its compatibility with the Arduino software ecosystem ensures ease of programming and a vast community for support and resources. Whether you're a beginner experimenting with simple circuits or an experienced developer creating sophisticated embedded systems, the Arduino Mega 2560 offers the capabilities needed to bring your ideas to life.

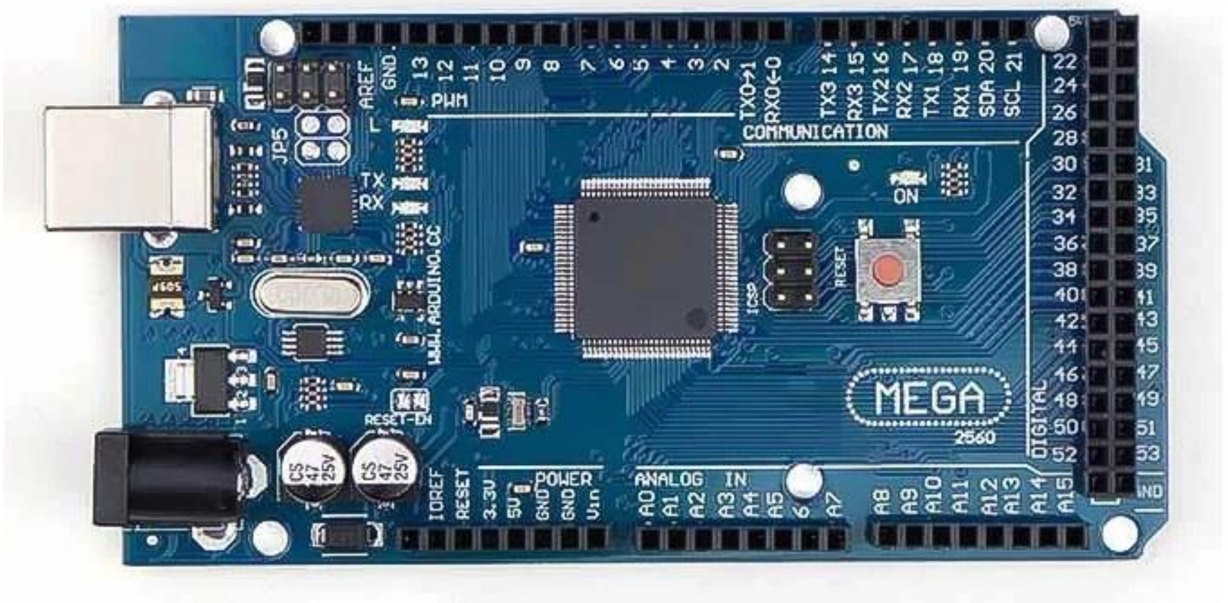


Fig. 4.2:- Arduino Mega 2560

Rotary Encoder:-

A rotary encoder is a device used to convert the angular position or rotation of a shaft into digital signals. The most common type of rotary encoder is the incremental rotary encoder, which generates pulses as the shaft is rotated. The five pins typically found on a rotary encoder include the positive and negative power supply pins (+,-), a pin for the switch (sw) which is often a momentary push-button switch, and two pins for the output signals (dt and clk), which stand for "data" and "clock."

The dt (data) pin outputs a signal that changes state whenever the encoder shaft is rotated, while the clk (clock) pin generates pulses at a fixed rate, indicating the direction of rotation. By monitoring the state changes of the dt pin relative to the clk pulses, the microcontroller or other digital system can determine both the direction and speed of rotation.

Rotary encoders find applications in various industries including robotics, industrial automation, CNC machines, and consumer electronics. They are commonly used in rotary control knobs for devices like volume controls on audio equipment, parameter adjustments on machinery, and user interface navigation in digital interfaces. They provide precise and reliable position sensing, making them ideal for applications requiring accurate rotational control. Additionally, their compact size, durability, and ability to provide continuous rotation without mechanical wear make them highly versatile in a wide range of applications.

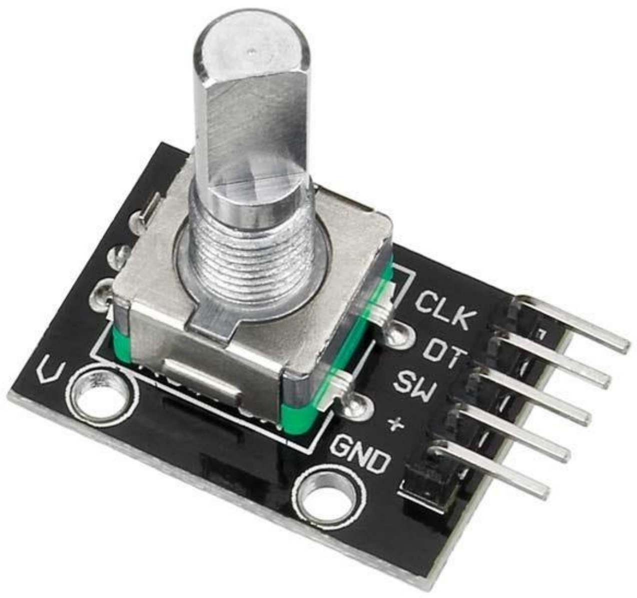


Fig. 4.3:- Rotary Encoder

Relay Module:-

A 5V relay module is an electronic device that acts as a switch when an electrical signal is applied to it. It typically consists of a coil, an armature, a spring, and one or more sets of contacts. When the coil is energized with a 5V signal, it generates a magnetic field that attracts the armature, causing the contacts to close or open, depending on the relay type.

These modules are commonly used in various applications, including home automation, industrial control systems, robotics, and IoT devices. They provide isolation between the control circuit (typically low voltage) and the load circuit (higher voltage or current), making them suitable for interfacing different voltage levels or protecting sensitive control circuitry.

In home automation, 5V relay modules are used to control appliances such as lights, fans, and motors remotely. In industrial settings, they play a vital role in controlling machinery, HVAC systems, and lighting. They are also frequently employed in Arduino and Raspberry Pi projects for controlling various electronic devices. Additionally, they can be utilized in safety circuits, alarm systems, and automotive applications.

Overall, the versatility, reliability, and ease of use of 5V relay modules make them indispensable components in a wide range of electronic systems, offering efficient switching capabilities and electrical isolation.

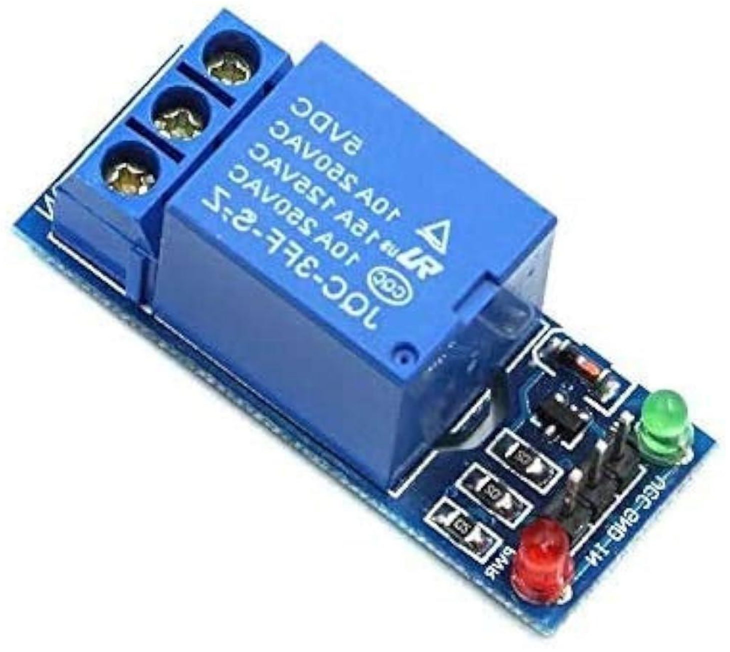


Fig. 4.4:- 5V Relay Module

16x2 LCD Display:-

A 16x2 LCD display refers to a liquid crystal display with 16 character columns and 2 rows, capable of displaying alphanumeric characters, symbols, and simple graphics. These displays are widely used in various electronic devices due to their simplicity, low power consumption, and cost-effectiveness.

One of the primary applications of a 16x2 LCD display is in embedded systems, such as microcontroller-based projects and Arduino prototypes. They serve as a convenient interface for users to interact with the system, providing real-time feedback and information display.

In consumer electronics, 16x2 LCD displays can be found in devices like digital thermometers, calculators, and clocks, where they show important data to users in a compact and readable format.

These displays are also utilized in industrial automation systems for monitoring and control purposes. They can display parameters like temperature, pressure, and humidity, enabling operators to monitor processes efficiently.

In the educational sector, 16x2 LCD displays are used in electronics and robotics labs to teach students about interfacing and programming microcontrollers. They provide a practical platform for learning about data output and manipulation.

Moreover, these displays find applications in automotive electronics, such as in-car infotainment systems and dashboard displays, where they convey vital information to drivers in a clear and concise manner.

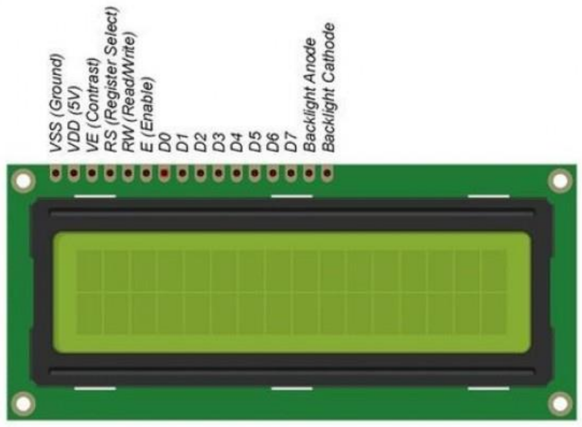


Fig. 4.5:- 16x2 LCD Display

Breadboard:-

A breadboard is a fundamental tool in electronics prototyping, providing a platform for quickly assembling and testing circuit designs without soldering. It typically consists of a plastic board with a grid of holes into which electronic components can be inserted and interconnected using wires. Each row of holes along the board is electrically connected, while the columns are usually isolated, allowing for easy organization of components.

Breadboards are widely used by hobbyists, students, and professionals alike for experimenting with circuit designs, testing ideas, and learning electronics principles. They are particularly valuable in the early stages of product development when rapid iteration and testing are essential.

One key advantage of breadboards is their reusability; components can be easily removed and rearranged to modify circuits as needed. This flexibility makes breadboards ideal for exploring different configurations and troubleshooting issues.

Breadboards are commonly used in educational settings to teach electronics concepts, as they provide a hands-on way for students to understand circuitry without the complexity and permanence of soldering.

In addition to education and prototyping, breadboards are also used in research laboratories for quick proof-of-concept experiments and in production environments for testing small-scale circuit designs before committing to final layouts.

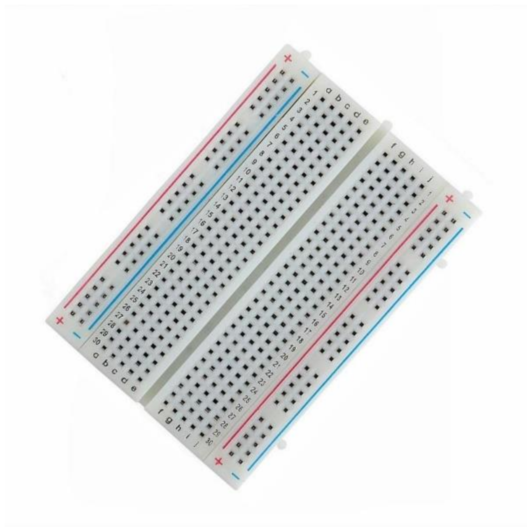


Fig. 4.6:- Breadboard

Single Socket Power Outlet Box:-

A single socket power outlet box is a compact electrical device designed to provide a single point of power access in residential, commercial, or industrial settings. It typically consists of a durable housing, a socket, and electrical connections. These outlets are widely used in various applications, including homes, offices, workshops, and retail spaces.

The single socket design allows for the connection of one electrical device at a time, making it ideal for situations where multiple outlets are not needed or space is limited. They are commonly installed in kitchens, bedrooms, living rooms, and offices for powering appliances, lamps, chargers, and other electronic devices.

In commercial and industrial settings, single socket outlet boxes may be used for specific equipment or machinery that requires dedicated power sources. They are also employed in temporary installations, such as construction sites or event venues, to provide reliable power supply in temporary setups.

These outlet boxes are available in different configurations to accommodate various voltage and current requirements, ensuring compatibility with a wide range of electrical devices. Safety features like grounding and surge protection may be integrated into the design to protect both the connected devices and users from electrical hazards.

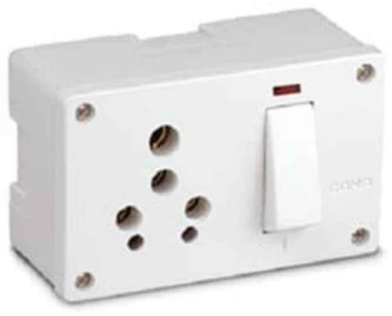


Fig. 4.7:- Single Socket Power Outlet Box

Connecting Wires:-

Connecting wires come in various configurations, typically classified as male-male (m-m), male-female (m-f), and female-female (f-f). Each configuration serves different purposes in electrical and electronic connections. Male-male cables feature connectors with pins on both ends, suitable for linking devices with corresponding sockets or ports. They are commonly used in data transfer, networking, and audio-visual setups. Male-female cables have a pin at one end (male) and a socket at the other (female), enabling connections between devices with different interfaces or genders. These are frequently utilized in extending cables or adapting between different connector types. Female-female cables consist of sockets on both ends, allowing for the extension or coupling of existing cables with the same gender connectors. They're handy for situations requiring lengthening or joining of cables without changing their gender. Overall, understanding the differences between these configurations is crucial for establishing secure and efficient connections in various electronic and electrical applications.

Fig. 4.8:- Connecting Wire

Chapter 5

Applications

1. Consumer Electronics: In the realm of smartphones, tablets, and laptops, where battery life and device longevity are paramount concerns, the Arduino Charging System offers users the ability to set precise charging durations, preventing overcharging and extending battery lifespan.

2. IoT Devices: Internet of Things (IoT) devices often operate on batteries and require periodic charging. With the Arduino Charging System, IoT developers can implement intelligent charging management, ensuring devices are only charged as needed, conserving energy, and prolonging battery life.

3. Healthcare: Medical devices, such as portable monitoring equipment or wearable health trackers, rely on continuous power supply. By integrating the Arduino Charging System, healthcare providers can ensure these devices are efficiently charged, minimizing downtime and enhancing patient care.

4. Education: In educational settings, where students use electronic devices such as tablets or Chromebooks, the Arduino Charging System promotes responsible battery management. Teachers can set charging schedules to coincide with off-peak energy hours, reducing electricity costs for schools and teaching students about energy conservation.

5. Smart Homes: Within smart home environments, where an array of connected devices requires regular charging, the Arduino Charging System can centralize and automate charging processes. Users can schedule charging cycles for various devices, optimizing energy usage and simplifying management.

6. Environmental Monitoring: Remote sensors and environmental monitoring devices deployed in harsh or inaccessible locations often rely on battery power. By incorporating the Arduino Charging System, researchers can ensure these devices are charged efficiently, maximizing data collection and reducing maintenance requirements.

Chapter 6

Conclusion and Future Scope

Conclusion:

In conclusion, the Arduino-based charging system with time-based shutdown presents a promising solution to address the challenges of overcharging and energy wastage in electronic devices. Through the implementation of user-settable timers and automatic shutdown features, the system offers enhanced user control, improved battery lifespan, and greater energy efficiency compared to traditional charging methods.

The comparative analysis conducted in this study, utilizing methodologies such as the Pugh matrix and Pairwise comparison, has provided valuable insights into the strengths and weaknesses of different charger designs. Charger C, with its wireless control capabilities and advanced features, emerges as a frontrunner in terms of user convenience and compatibility with various devices

Future Scope:

Looking ahead, there are several avenues for further research and development in this area. Firstly, refining the user interface and incorporating feedback mechanisms could enhance the usability and customization options of the charging system. Additionally, exploring integration with renewable energy sources and smart grid technologies could lead to even greater energy savings and sustainability.

Furthermore, considering the rapid evolution of electronic devices and charging technologies, ongoing innovation and adaptation will be essential to ensure the continued relevance and effectiveness of the Arduino-based charging system. Collaborations with industry partners and stakeholders could facilitate real-world testing and deployment, paving the way for widespread adoption and impact.

Chapter 7
Code

```
#include <LiquidCrystal.h>
#include <EEPROM.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#define outputA 13 // clk pin of rotary encoder
#define outputB 6 // dt pin of rotary encoder
#define button 7   // sw pin of rotary encoder
#define led 10     // status led
#define relay 14   // in1 of relay

int count = 1;
int current_state;
int previous_state;
int hh = 0;
int mm = 0;
int ss = 0;
int h = 0;
int m = 0;
int s = 0;
int ledToggle;
int previousState = HIGH;
unsigned int previousPress;
volatile int buttonFlag;
int buttonDebounce = 20;

void setup()
{
lcd.begin(16, 2);
pinMode(outputA, INPUT);
pinMode(outputB, INPUT);
pinMode(button, INPUT_PULLUP);
pinMode(led, OUTPUT);
pinMode(relay, OUTPUT);
digitalWrite(relay, HIGH);
attachInterrupt(digitalPinToInterrupt(2), button_ISR, CHANGE);

Serial.begin(9600);

previous_state = digitalRead(outputA);

if (EEPROM.read(0) != 0 || EEPROM.read(1) != 0 || EEPROM.read(2) != 0)
{
```

```
lcd.setCursor(0, 0);
lcd.print("Load Preset ?");
lcd.print(" ");
lcd.setCursor(0, 1);
lcd.print(" <NO> ");
int temp = 5;

while (temp > 0 && ledToggle == 0)
{
lcd.setCursor(14, 0);
lcd.print(temp);
temp--;
delay(1000);
}

if (temp == 0 && ledToggle == 0)
{
hh = EEPROM.read(0);
mm = EEPROM.read(1);
ss = EEPROM.read(2);
timer();
}
else
{
EEPROM.write(0, 0);
EEPROM.write(1, 0);
EEPROM.write(2, 0);
ledToggle = 0;
}
}

lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" HH MM SS OK ");
lcd.setCursor(0, 1);
lcd.print(" 0 0 0 ");
}

void loop()
{
encoder();

if (count == 1)
{
lcd.setCursor(0, 0);
lcd.print("<HH> MM SS OK ");

while (count == 1)
{
encoder();

if (ledToggle)
{
h = 1;
m = 0;
s = 0;

lcd.setCursor(0, 0);
lcd.print(" HH MM SS OK ");
lcd.setCursor(1, 1);
lcd.print(hh);
lcd.setCursor(5, 1);
lcd.print(mm);
lcd.setCursor(9, 1);
lcd.print(ss);
lcd.setCursor(0, 1);
lcd.print('<');
lcd.setCursor(3, 1);
lcd.print('>');

while (ledToggle)
{
encoder();
lcd.setCursor(1, 1);
lcd.print(hh);
}
}

EEPROM.write(0, hh);

h = 0;
m = 0;
s = 0;

lcd.setCursor(0, 0);
lcd.print("<HH> MM SS OK ");
lcd.setCursor(0, 1);
lcd.print(' ');
lcd.setCursor(3, 1);
lcd.print(' ');
```

```
}
}
else if (count == 2)
{
lcd.setCursor(0, 0);
lcd.print(" HH <MM> SS OK ");

while (count == 2)
{
encoder();

if (ledToggle)
{
h = 0;
m = 1;
s = 0;

lcd.setCursor(0, 0);
lcd.print(" HH MM SS OK ");
lcd.setCursor(1, 1);
lcd.print(hh);
lcd.setCursor(5, 1);
lcd.print(mm);
lcd.setCursor(9, 1);
lcd.print(ss);
lcd.setCursor(4, 1);
lcd.print('<');
lcd.setCursor(7, 1);
lcd.print('>');

while (ledToggle)
{
encoder();
lcd.setCursor(5, 1);
lcd.print(mm);
}
}

EEPROM.write(1, mm);

h = 0;
m = 0;
s = 0;

lcd.setCursor(0, 0);
lcd.print(" HH <MM> SS OK ");
lcd.setCursor(4, 1);
lcd.print(' ');
lcd.setCursor(7, 1);
lcd.print(' ');
}
}
else if (count == 3)
{
lcd.setCursor(0, 0);
lcd.print(" HH MM <SS> OK ");

while (count == 3)
{
encoder();

if (ledToggle)
{
h = 0;
m = 0;
s = 1;

lcd.setCursor(0, 0);
lcd.print(" HH MM SS OK ");
lcd.setCursor(1, 1);
lcd.print(hh);
lcd.setCursor(5, 1);
lcd.print(mm);
lcd.setCursor(9, 1);
lcd.print(ss);
lcd.setCursor(8, 1);
lcd.print('<');
lcd.setCursor(11, 1);
lcd.print('>');

while (ledToggle)
{
encoder();
lcd.setCursor(9, 1);
lcd.print(ss);
}
}

EEPROM.write(2, ss);
```

```
h = 0;
m = 0;
s = 0;

lcd.setCursor(0, 0);
lcd.print(" HH MM <SS> OK ");
lcd.print(' ');
lcd.setCursor(11, 1);
lcd.print(' ');
}
}
else if (count == 4)
{
lcd.setCursor(0, 0);
lcd.print(" HH MM SS <OK>");

while (count == 4)
{
encoder();

if (ledToggle)
{
timer();
}
}
}
}

void encoder()
{
current_state = digitalRead(outputA);

if (current_state != previous_state)
{
if (digitalRead(outputB) != current_state)
{
if (count < 4 && ledToggle == 0)
count++;
else
{
if (h == 1)
{
if (hh < 23)
hh = hh + 1;
}
else if (m == 1)
{
if (mm < 59)
mm = mm + 1;
}
else if (s == 1)
{
if (ss < 59)
ss = ss + 1;
}
}
}
else
{
if (count > 1 && ledToggle == 0)
count--;
else
{
if (h == 1)
{
if (hh > 0)
hh = hh - 1;
}
else if (m == 1)
{
if (mm > 0)
mm = mm - 1;
}
else if (s == 1)
{
if (ss > 0)
ss = ss - 1;
}
}
}

Serial.print("Position: ");
Serial.println(count);
}

previous_state = current_state;
}

void button_ISR()
{
```

```
buttonFlag = 1;

if ((millis() - previousPress) > buttonDebounce && buttonFlag)
{
previousPress = millis();

if (digitalRead(button) == LOW && previousState == HIGH)
{
ledToggle = !ledToggle;
digitalWrite(led, ledToggle);
previousState = LOW;
}
else if (digitalRead(button) == HIGH && previousState == LOW)
{
previousState = HIGH;
}

buttonFlag = 0;
}
}

void timer()
{
digitalWrite(relay, LOW);

while (1)
{
lcd.setCursor(0, 0);
lcd.print("Charging... ");
lcd.setCursor(0, 1);

if (hh < 10)
{
lcd.print('0');
lcd.print(hh);
lcd.print(':');
}
else
{
lcd.print(hh);
lcd.print(':');
}

if (mm < 10)
{
```

```
lcd.print('0');
lcd.print(mm);
lcd.print(':');
}
else
{
lcd.print(mm);
lcd.print(':');
}

if (ss < 10)
{
lcd.print('0');
lcd.print(ss);
lcd.print(" ");
}
else
{
lcd.print(ss);
lcd.print(" ");
}

Serial.println(ss);

for (ss; ss > 0; ss--)
{
Serial.println(ss);

if (ss < 10)
{
lcd.setCursor(6, 1);
lcd.print('0');
lcd.print(ss);
EEPROM.write(2, ss);
}
else
{
lcd.setCursor(6, 1);
lcd.print(ss);
EEPROM.write(2, ss);
}

delay(1000);
}
```

```
if (ss == 0 && mm == 0 && hh == 0)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Charged");
digitalWrite(relay, HIGH);

EEPROM.write(0, 0);
EEPROM.write(1, 0);
EEPROM.write(2, 0);

while (1)
;
}

ss = 59;

if (mm != 0)
mm = mm - 1;

if (hh != 0)
hh = hh - 1;

EEPROM.write(0, hh);
EEPROM.write(1, mm);
EEPROM.write(2, ss);
}
}
```

Chapter 8

References

[1] Arduino. (n.d.). Arduino - Home. Retrieved from https://www.arduino.cc/

[2] Lu, T., Zhang, L., & Xiong, N. (2016). Design and implementation of Arduino-based automatic battery charger. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 002933–002938. https://doi.org/10.1109/SMC.2016.7844725

[3] Gupta, N., & Khare, A. (2017). Design of automatic mobile charger using Arduino Uno. 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), 157–160. https://doi.org/10.1109/ICTUS.2017.8066502

[4] Haghighatpanah, N., Nezamabadi-pour, H., & Tabarraei, A. (2016). IoT-based smart charger for electric vehicles. 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 203–208. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2016.37

[5] https://www.electronicshub.org/arduino-based-smartphone-charging-controller/

[6] Chat GPT https://chat.openai.com/