SMART MOBILE CHARGER

A Project Report

Submitted in partial fulfillment of the
requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Name of The Student : Nikhilesh Pednekar

Seat Number : 17302A0006

Under the esteemed guidance of

Mr. Laxmikant Manchekar

Assistant Professor, Department of Information Technology

VSIT | Vidyalankar School of Information Technology
NAAC ACCREDITED COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY

(Affiliated to University of Mumbai)

MUMBAI, 400 037

MAHARASHTRA

2019 - 2020

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY

(Affiliated to University of Mumbai)

MUMBAI-MAHARASHTRA-400037

DEPARTMENT OF INFORMATION TECHNOLOGY

VSIT | Vidyalankar School of Information Technology

NAAC ACCREDITED COLLEGE

CERTIFICATE

This is to certify that the project entitled, **" SMART MOBILE CHARGER"**, is bonafied work of **NIKHILESH NILESH PEDNEKAR** bearing Seat No: 17302A0006 submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

Internal Examiner **External Examiner**

Date: **College Seal** **Principal**

ABSTRACT

Smart Mobile charger is a device that mainly is a switch mode power supply that has the ability to communicate with the battery of smart phone, batter management system in order to control and monitor the charging process of the phone. It automatically cuts off the power supply which disconnects the power from the main switch when the device is fully charged.The main concept behind the smart mobile charger is very simple. Set the time for which you wish to charge your mobile phone. Once the time is reached, turn off the power supply to the charger.The user can charge his phone for a particular time period by setting the timer for charging the device. Every battery has a limit to the number of time it can be charged (known as charge cycles). Also temperature plays an important role in the life of a battery. Higher temperatures might disrupt the chemistry of the battery.

ACKNOWLEDGEMENT

It gives us immense pleasure to express our sincere gratitude to those who are associated with our project **SMART MOBILE CHARGER** which is a hardware device for effectively charging our cell phone devices on daily basis as a Part of the course of BSc(IT) affiliated by the University of Mumbai.

We are grateful to our Principle Dr. Rohini Kelkar to give us an opportunity to take up a Project.

We would like to thank our Project Guide Mr, Laxmikant Manchekar who has been a constant source of motivation, encouragement and guidance, His Valuable guidance and timely help has helped us in completing the project.

Above all our thanks to our family and friends for their help and motivation in making of our project

DECLARATION

I hereby declare that the project entitled, "**SMART MOBILE CHARGER**" done at Vidyalankar School of Information Technology, has not been in any case duplicated to submit to any other universities for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Smart Mobile charger is a device that mainly is a switch mode power supply that has the ability to communicate with the battery of smart phone, batter management system in order to control and monitor the charging process of the phone. It automatically cuts off the power supply which disconnects the power from the main switch when the device is fully charged.

The main concept behind the smart mobile charger is very simple. Set the time for which you wish to charge your mobile phone. Once the time is reached, turn off the power supply to the charger.

The user can charge his phone for a particular time period by setting the timer for charging the device. Every battery has a limit to the number of time it can be charged (known as charge cycles). Also temperature plays an important role in the life of a battery. Higher temperatures might disrupt the chemistry of the battery.

The time for which we have set the charger to charger the device is displayed on the LCD screen. If there is a power failure, the remaining time is stored in the memory and when the power comes back, it will prompt you whether to continue with the countdown or to set a new time. Accordingly, the charging will be performed.

1.1 Background

Almost every mobile phone user faces the one issue i.e. connecting your phone to the charging adapter and forgetting that you plugged in the device. Almost all the modern charge controllers on mobile phones are very advanced and detect when your battery is fully charged and disconnect supply of power to the battery (not completely but it keeps the device in a charging state also known as the trickle charge state). But keeping the device plugged in even after the battery is full is its effect on the lifetime of the battery.

Smart Mobile charger take this concept of charging a step further. But setting the charge time (may be based on the previous observation or by mathematical calculations), we can control the charging process by setting sufficient amount of time required for charging that device. After the device gets charged for the charge time and when the countdown is reached, the relay is turned off and as a result, the power supply to the charger is also cut off.

1.2 Objective

The Objective for creating this project is to implement that this smart mobile charger is for controlling the amount of time you charge your mobile phones. It limits the charging power so that it does not affect the cell phone battery to be damaged or overheated due to excessive and unnecessary charging and hence creating a safe environment for charging phones.

1.3 Purpose

The purpose of this project is that you can plug in your phone to the charger, set the time for which your phone get charged and forget as the project will automatically disconnect power to the charger. This project is very useful for people, who tend to charge the phone during the night time or those who often forget that they plugged in the phone to the charger.

1.3 Scope

Scope of the this project are as follows :

- Better and efficient charging experience.
- It provides a safe environment for charging phones, as it cuts off the power supply to the charger, thus reducing the risk of overheating and this can help to preserve the battery life and also prevent the cell phone blasts due to overheating of batteries.

1.3 Applicability/Feasibility Study

A feasibility study is an analysis that takes all of a project's relevant fact into account to ascertain the likelihood of completing the project successfully.

Economic Feasibility :

This project is economically feasible. The project can be completed in time and its cost efficient. All the factors like.

- Cost of hardware.
- Cost of maintenance
- Cost of implementation of resources all are taken into consideration.

Technical Feasibility :

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. When writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study
- The part of the business being examined
- The human and economic factor
- The possible solutions to the problem

At this level, the concern is whether the proposal is both *technically* and *legally* feasible (assuming moderate cost).

Chapter 2

Survey of Technologies

Implementation Tools

1. **Arduino :-**

Arduino is an open source electronics platform based on easy to use hardware and software. Arduino boards are able to read inputs – light on a sensor, a finger on a button, or a twitter message, and turn it into a output – activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To so do you use the Arduino Programming language and the Arduino Software (IOE), based on processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

2. Raspberry Pi

The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards and mice) or cases. However, some accessories have been included in several official and unofficial bundles.

The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with Eben Upton as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.

Chapter 3

Requirements and Analysis

The goal of the requirement analysis phase is to clearly understand the customer requirements and to systematically organize the requirements into a specification document. The main activities carried out during requirements analysis phase are as follows:. Requirements gathering The main purpose of the requirements analysis activity is to analyze collected information .To obtain a clear understanding of the product to be developed, with a view to removing incompleteness, and inconsistencies from the initial customer perception of problem. Requirement Analysis is the first and important step in the Software development for building robust and user-friendly applications.

3.1 PROBLEM DEFINITION

People tend to charge the phone durning night time or some people forget that their charger is plugged in to the phone and leave their phone for charging.Keeping the device plugged in even after the battery is full affects the lifetime of battery.Also temperature plays an important role in the life of the battery.Higher temperature might disrupt the chemistry of battery.this problem can be solved with the help of this smart charger.as this charger will set the time for charging the device and will automatically stop the charging when the countdown is over. Hence it will not charge the phone further and reduce the risk of damaging the battery.

3.2 REQUIREMENT SPECIFICATION

1. Arduino

Arduino is an open source electronics platform based on easy to use hardware and software. Arduino boards are able to read inputs – light on a sensor, a finger on a button, or a twitter message, and turn it into a output – activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To so do you use the Arduino Programming language and the Arduino Software (IOE), based on processing.

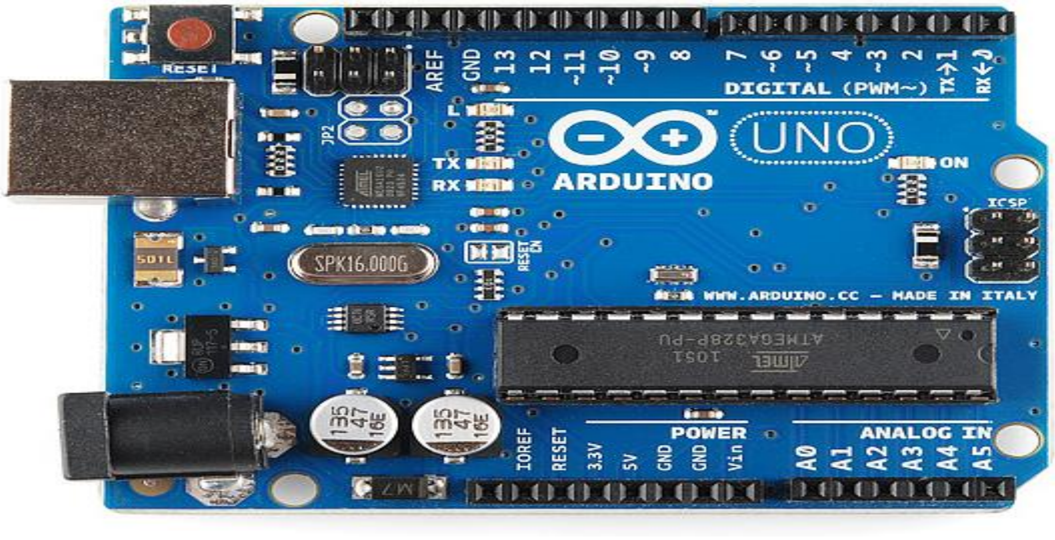


Figure 1:Arduino UNO

2. 16 X 2 LCD display :

An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16 X 2 LCD display is a very basic module commonly used in OIYs and circuits. The 16 x 2 translates a display 16 characters per lines in 2 such lines. In this LCD each character is displayed in a 5 x 7 pixel matrix.

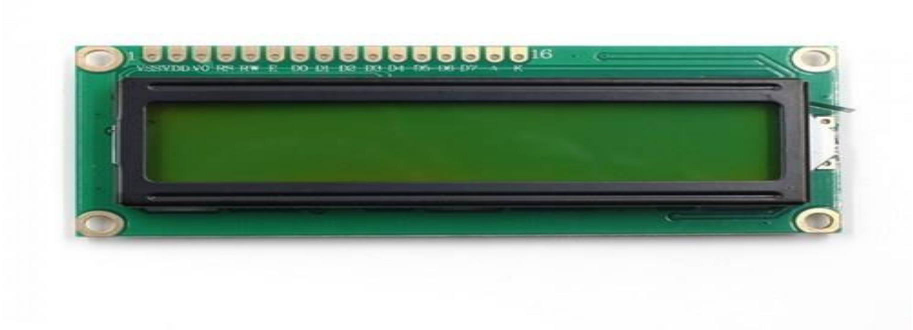


Figure 2:16x2 LCD Display

3. ROTARY ENCODER :

A rotary encoder, also called a shaft encoder is an electro mechanical device that converts the angular position or motion of a shaft or oxle to analog or digital output signals. Rotary encodes are use in a wide range of application that require monitoring or control, or both of mechanical systems, includes controls, robotics, photographic lenses, computer input devices such as optomechanical mice and trackballs, controlled stress rhemeters and rotating radar platforms

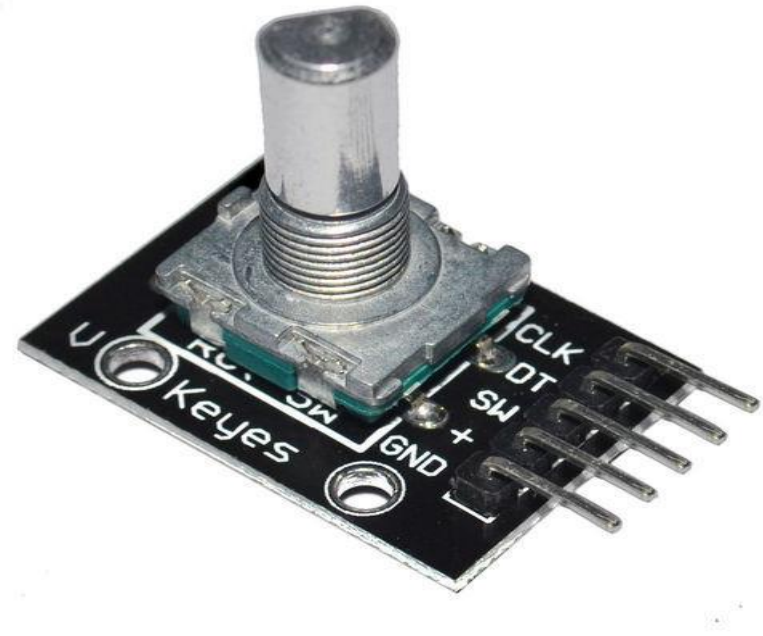


Figure 3:Rotary Encoder

4. CHANNEL SV-2 RELAY MODULE

This is a low level SV-2 channel relay interface board, and each channel needs a 15-20 MA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by controller directly by microcontroller.

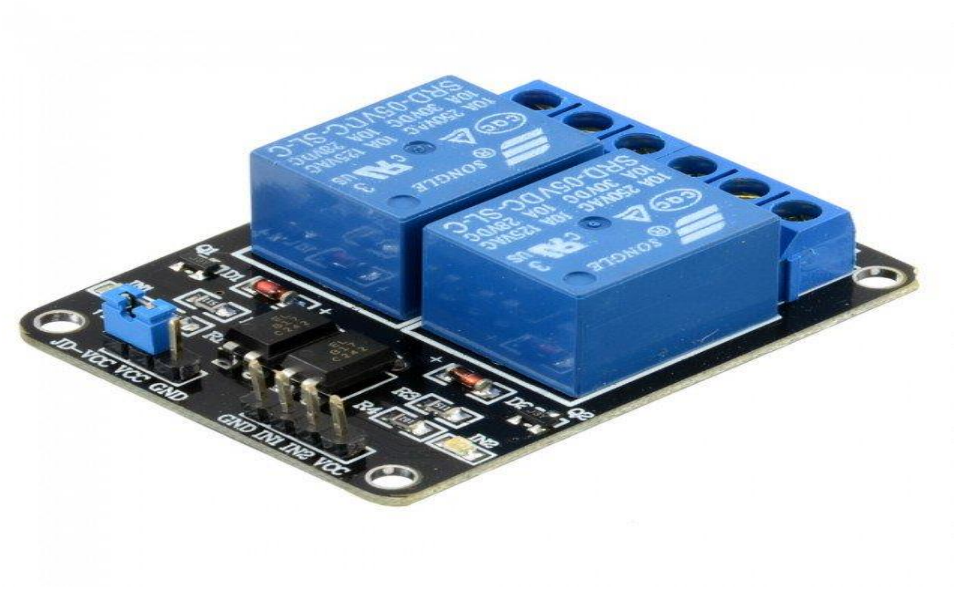


Figure 4:Relay Module

5. Breadboard :

A breadboard is a construction base for prototyping of electronics. Older breadboard types did not have this property. A variety of electronics systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units.

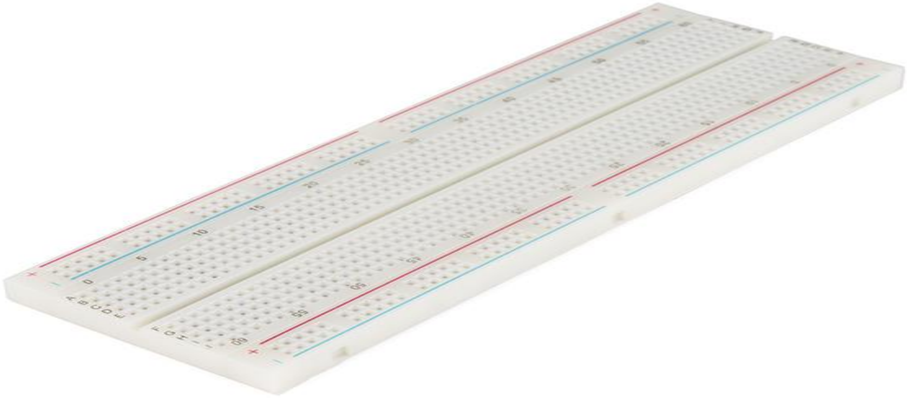


Figure 5:Breadboard

6. 10K POT

A **potentiometer** is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider.[1] If only two terminals are used, one end and the wiper, it acts as a **variable resistor** or **rheostat**. The measuring instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

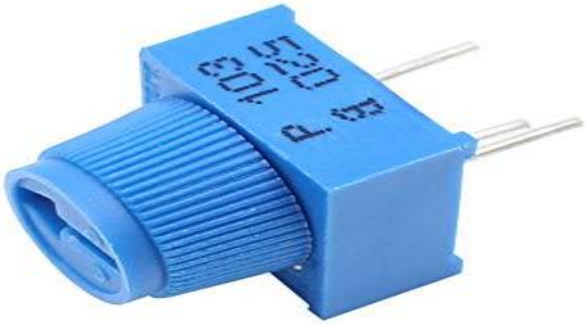


Figure 6:10K Pot

NON FUNCTIONAL REQUIREMENTS

Stability:- Objects will be stable for long time and will not need changes

Extensibility:- Application takes future growth into consideration

Reliability:- Answer and services provided will be 100% accurate

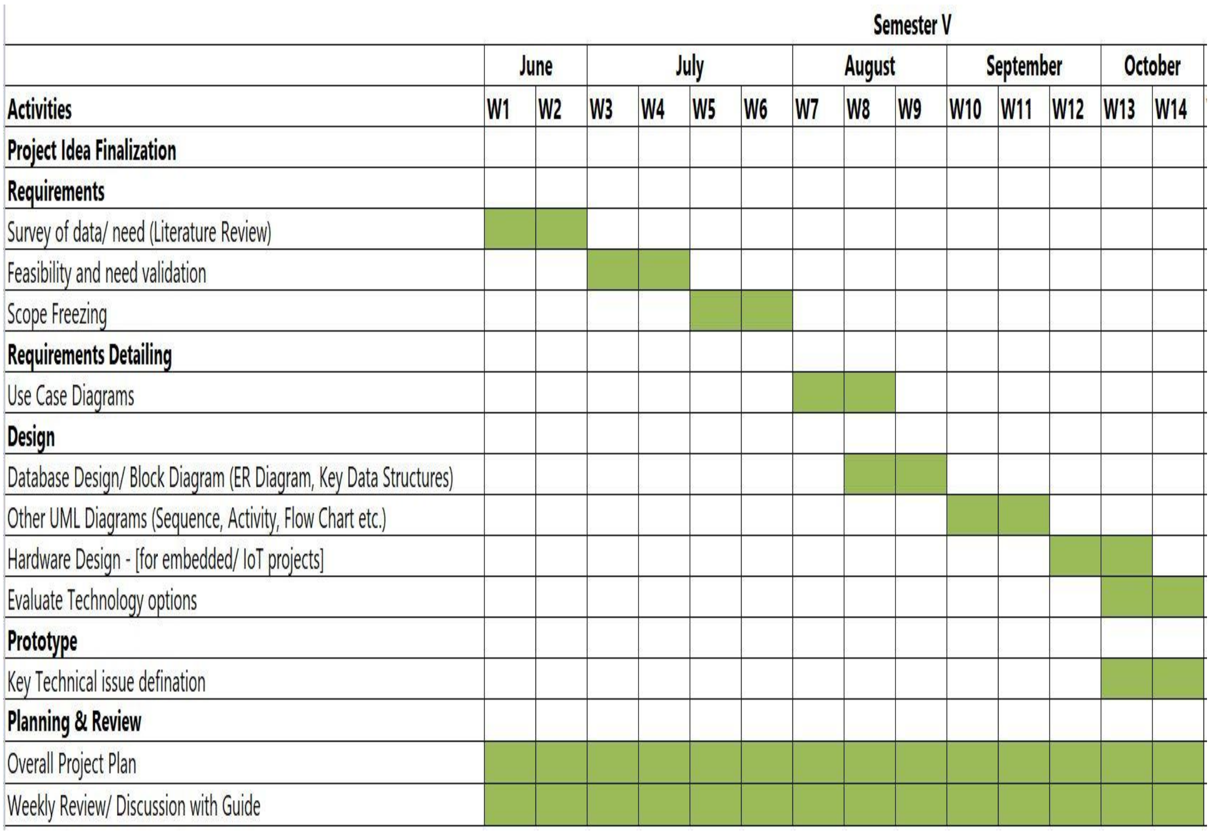
3.3 Planning and Scheduling

In order to complete our project, it was necessary for us to learn various technologies. Few technologies where familiar to us but for some of them we had to start learning from scratch. So we allocated time for each technology accordingly

Sr No.	Technology		Allocated Time for learning
1	Arduino IDE	• Auto Format • Sketch Archive • Fix Encoding & Reload • Board Selection & Management • Port Menu • Programmer Functions • Burn Bootloader	4 Weeks

Table 1:Planning and scheduling

Gant Chart



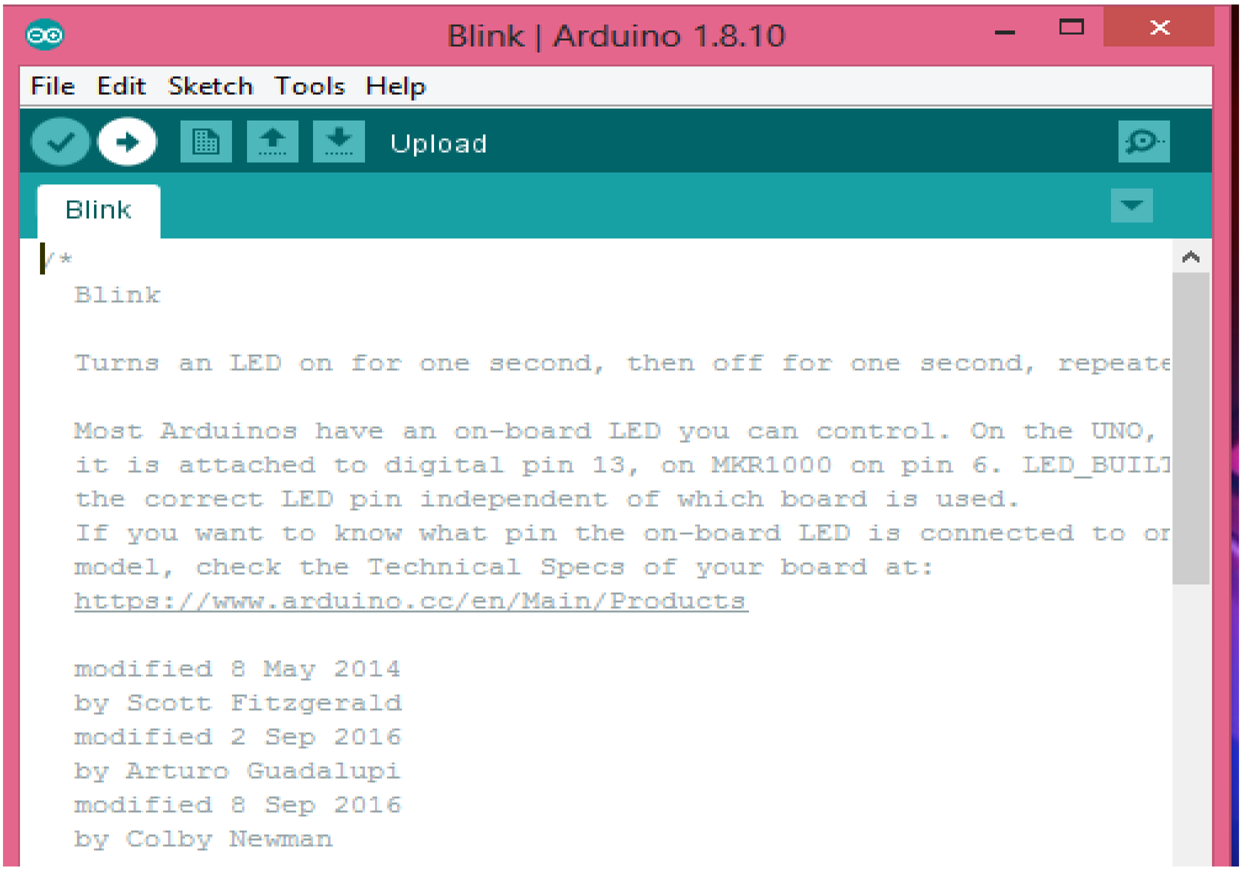
	Semester V													
	June		July				August			September			October	
Activities	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**	**W8**	**W9**	**W10**	**W11**	**W12**	**W13**	**W14**
Project Idea Finalization														
Requirements														
Survey of data/ need (Literature Review)	🟩	🟩												
Feasibility and need validation			🟩	🟩										
Scope Freezing					🟩	🟩								
Requirements Detailing														
Use Case Diagrams							🟩	🟩						
Design														
Database Design/ Block Diagram (ER Diagram, Key Data Structures)								🟩	🟩					
Other UML Diagrams (Sequence, Activity, Flow Chart etc.)										🟩	🟩			
Hardware Design - [for embedded/ IoT projects]												🟩	🟩	
Evaluate Technology options													🟩	🟩
Prototype														
Key Technical issue defination													🟩	🟩
Planning & Review														
Overall Project Plan	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Weekly Review/ Discussion with Guide	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Figure 7:Gant Chart

3.4 HARDWARE AND SOFTWARE REQUIREMENTS

The **Arduino integrated development environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.



```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeate

  Most Arduinos have an on-board LED you can control. On the UNO,
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILT
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to or
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman
```

Figure 8:Arduino

3.5 Preliminary Product Description

Smart Mobile charger is a device that mainly is a switch mode power supply that has the ability to communicate with the battery of smart phone, batter management system in order to control and monitor the charging process of the phone. It automatically cuts off the power supply which disconnects the power from the main switch when the device is fully charged.

The main concept behind the smart mobile charger is very simple. Set the time for which you wish to charge your mobile phone. Once the time is reached, turn off the power supply to the charger.

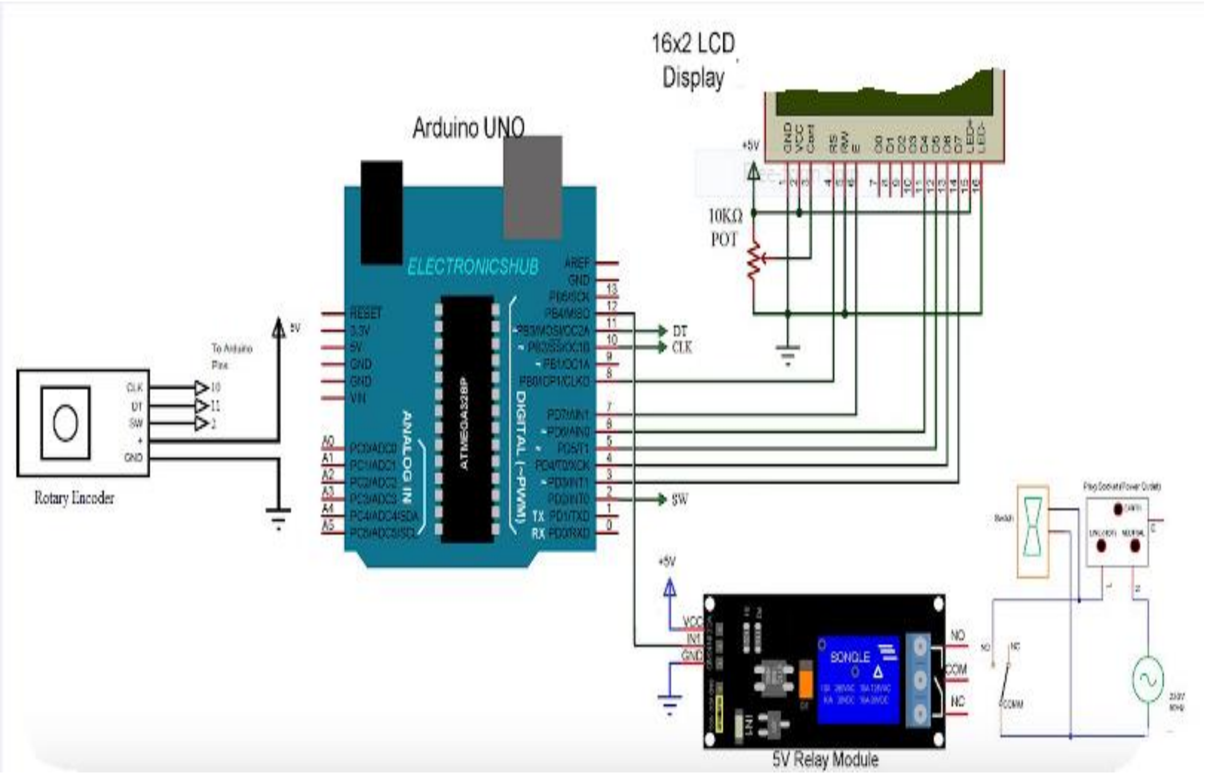


Figure 9:Preliminary Product Description

3.6 Conceptual Model

As we have used Waterfall Model for creating this Application

In “The Waterfall Model” approach, the whole process of software development is divided into separate phases/parts. In this Waterfall model, typically, the outcome of one phase act as the input for the next phase sequentially in flow.

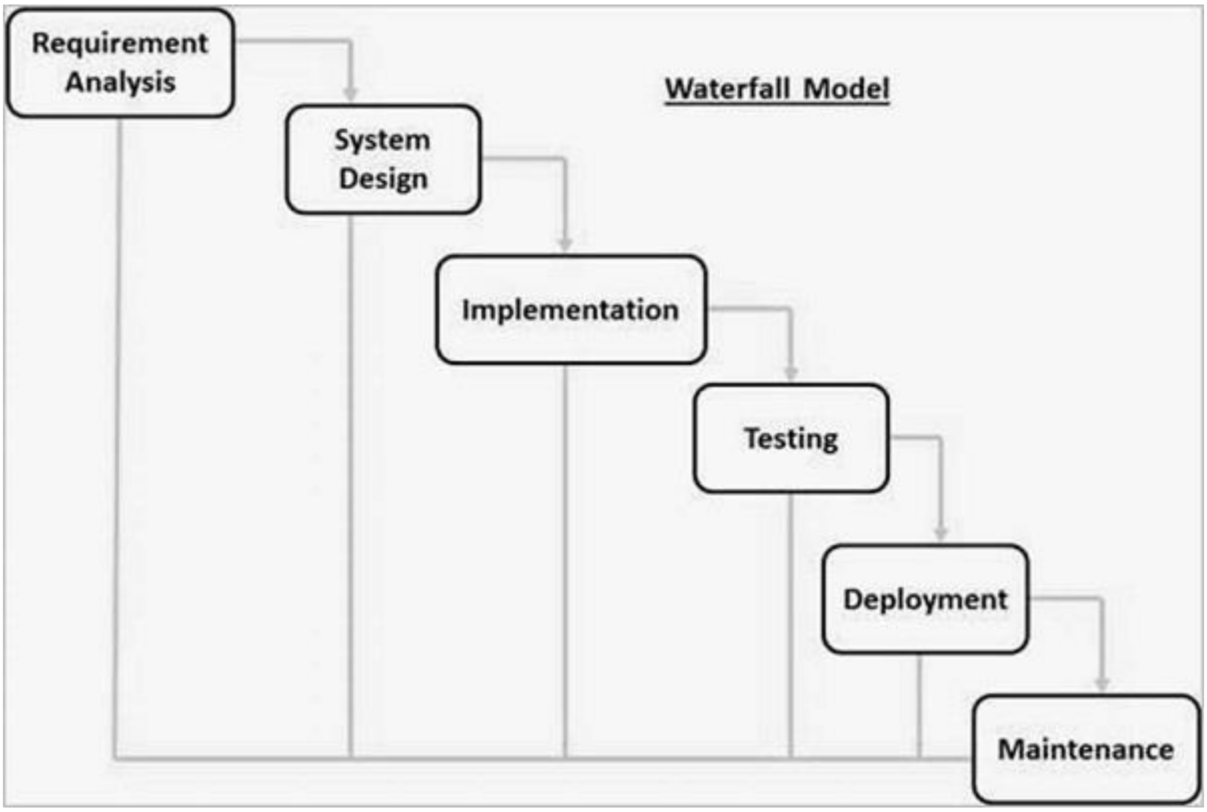


Figure 10:Waterfall Model

- Requirements are well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the item.

Chapter 4

System Designs

4.3 Diagrams

4.3.1 Block diagrams

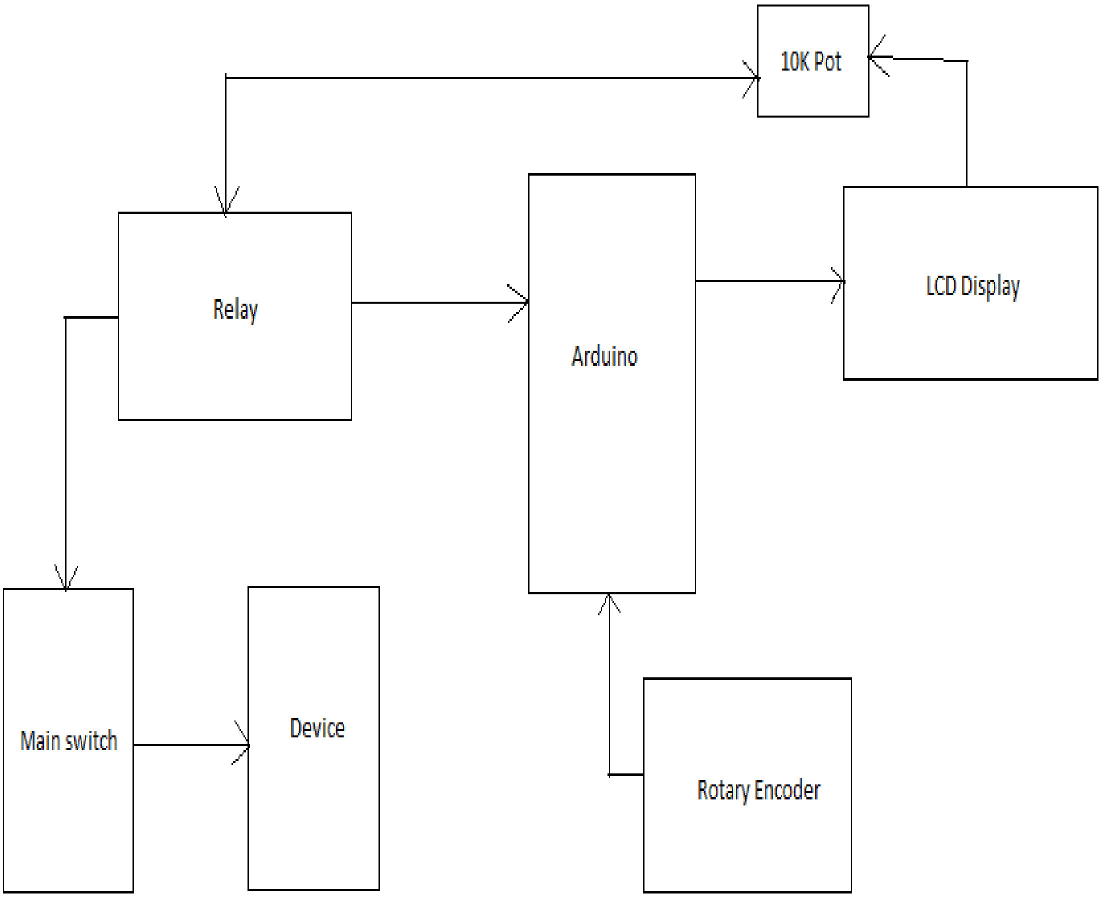


Figure 11:Block Diagram

A **block diagram** is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams.

Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction.

Purpose of Block Diagrams:

- Helps you to define terms related to each other
- Provide a preview of how all your tables should connect, what fields are going to be on each table.

4.3.2 Circuit Diagram

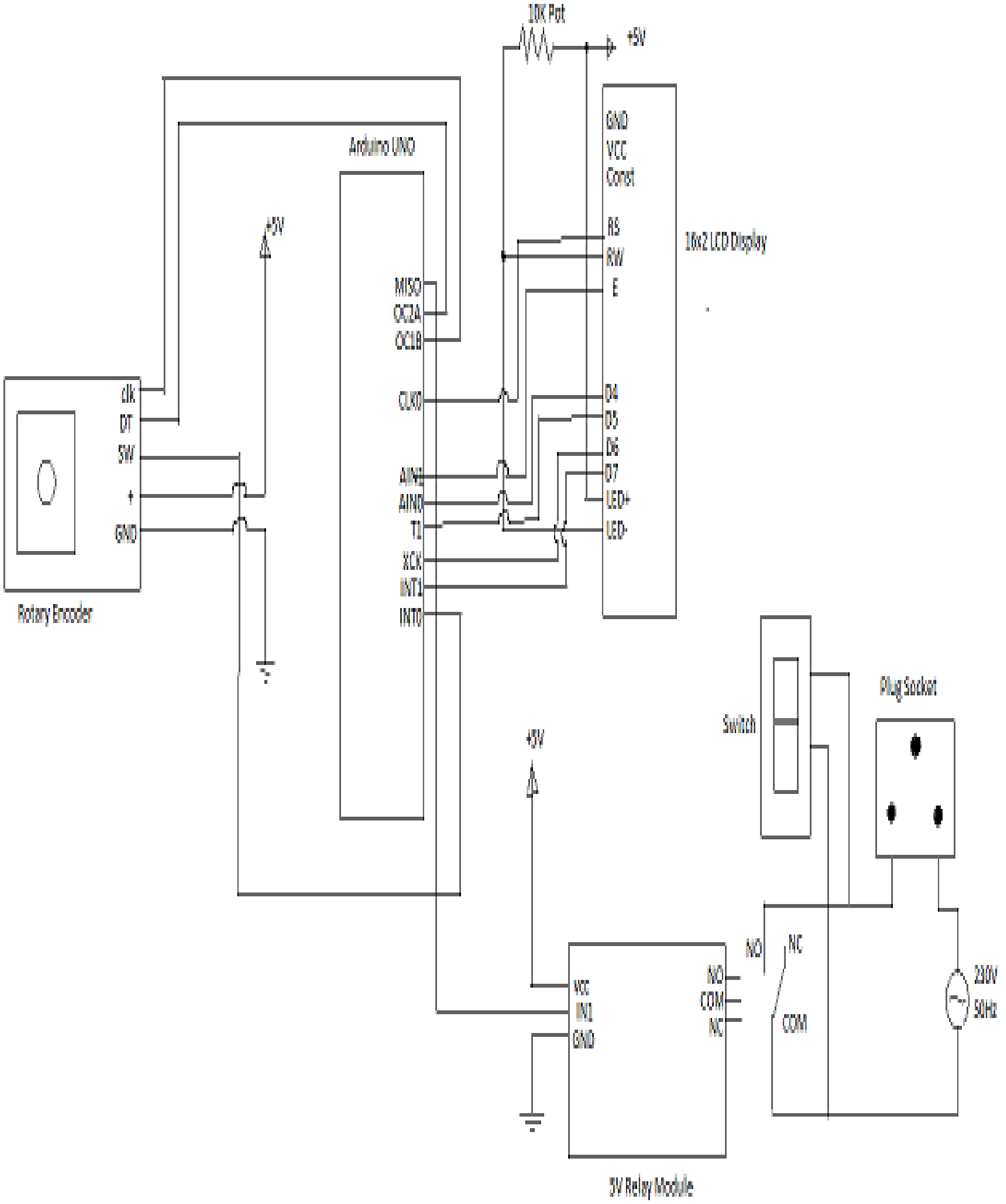


Figure 12:Circuit Diagram

A **circuit diagram** (**electrical diagram**, **elementary diagram**, **electronic schematic**) is a graphical representation of an electrical circuit. A pictorial circuit diagram uses simple images of components, while a schematic diagram shows the components and interconnections of the circuit using standardized symbolic representations. The presentation of the interconnections between circuit components in the schematic diagram does not necessarily correspond to the physical arrangements in the finished device.

Unlike a block diagram or layout diagram, a circuit diagram shows the actual electrical connections. A drawing meant to depict the physical arrangement of the wires and the components they connect is called *artwork* or layout, *physical design*, or *wiring diagrams.*

4.3.3 Sequence Diagram

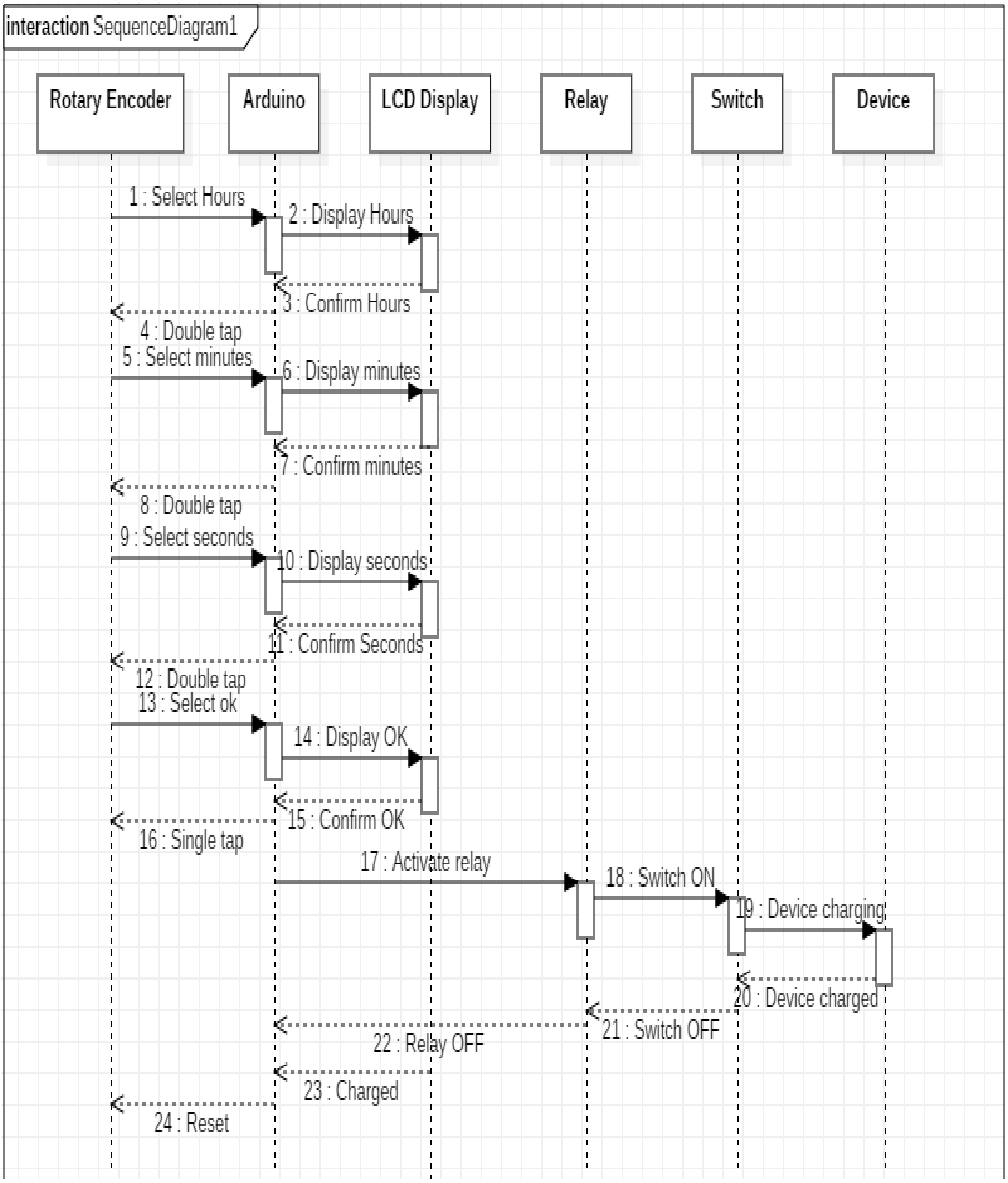


Figure 13:Sequence Diagram

A **sequence diagram** shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

Purpose of Sequence Diagrams:

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation

4.3.4 Activity Diagram

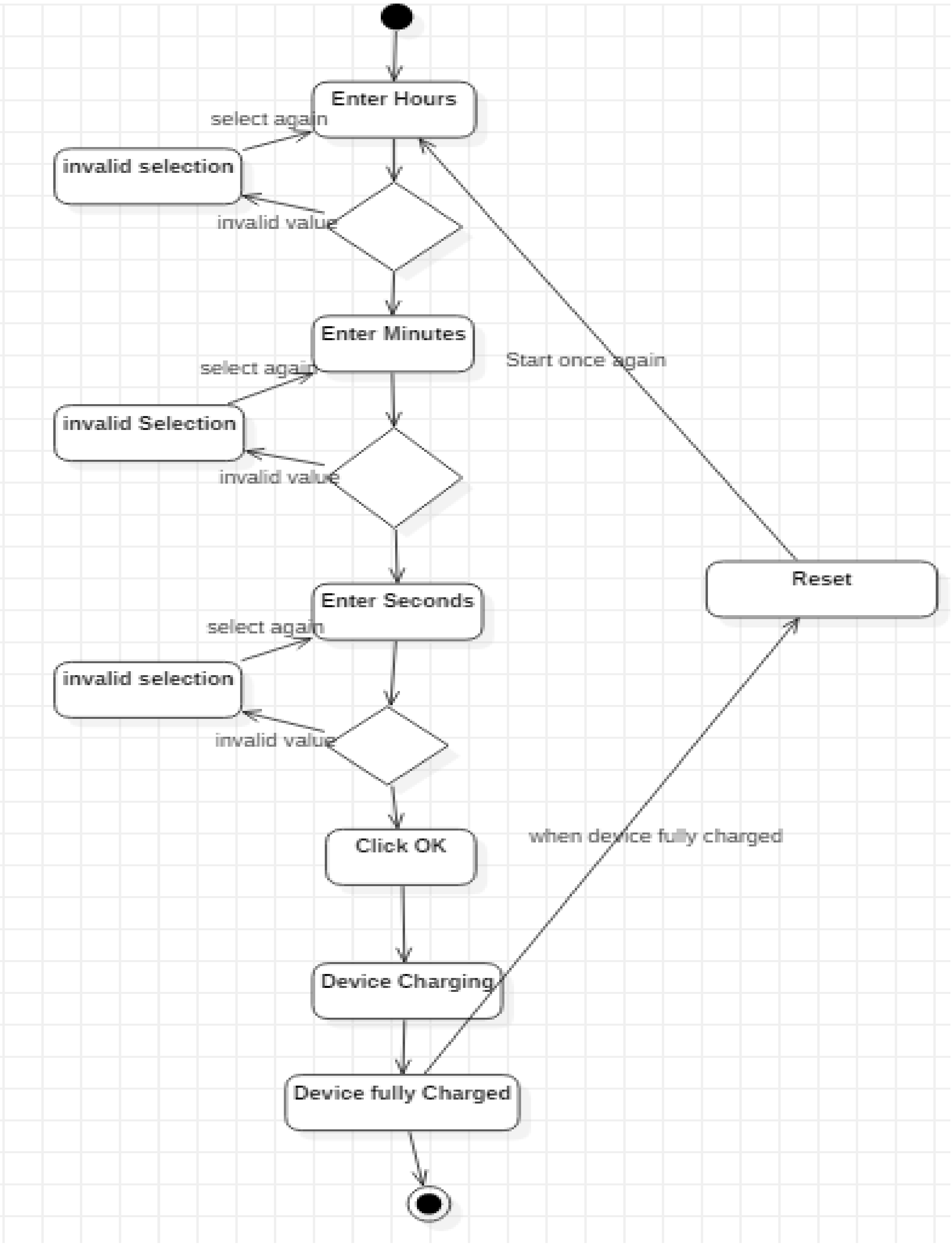


Figure 14:Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling

Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

Purpose of Activity Diagram:

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

4.3.5 DataFlow Diagram

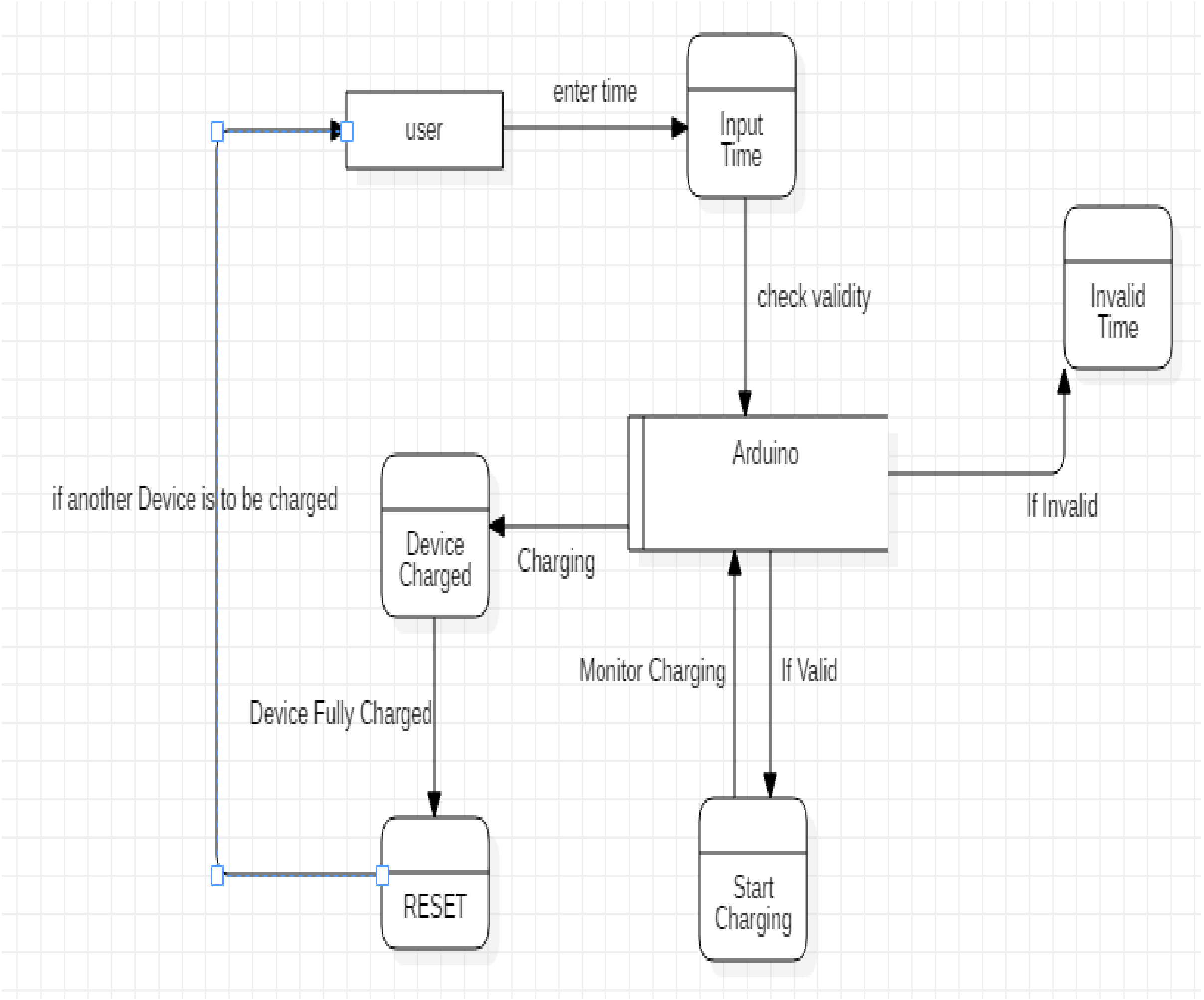


Figure 15:DataFlow Diagram

A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

Purpose of Dataflow diagram:

- Data flow diagrams provide a graphical representation of how information moves between processes in a system.
- Data flow diagrams follow a hierarchy; that is, a diagram may consist of several layers, each unique to a specific process or data function.

4.3.6 Use Case Diagram

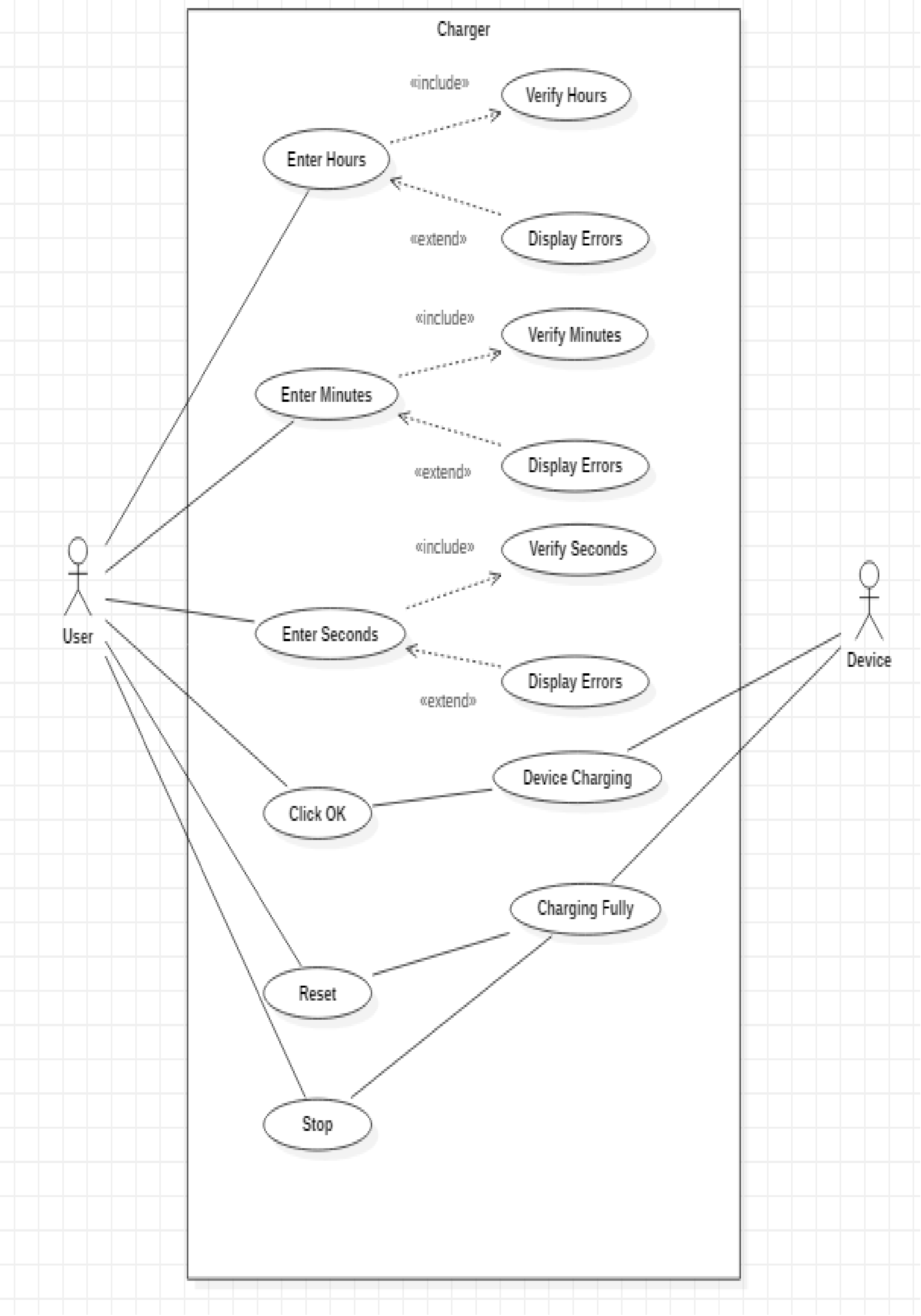


Figure 16:Use Case Diagram

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

Purpose of use case diagrams:

- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture

4.3.7 Event Tree

Event	Trigger	Initiator	Use case	Output	Recipient
Enter Time	Time	User	Display Time	Time Valid	User
Click OK	Time	User	Successful	Time OK	User
Charge	Service	User	Charging Device	Device is charging	Device
Reset/Stop	Service	User	Reset system/Stop system	Device fully charged	user

Table 2:Event Tree

4.8.9 Security Issues

A hardware vulnerability is an exploitable weakness in a computer system that enables attack through remote or physical access to system hardware.

Any means by which code can be introduced to a computer is inherently a hardware vulnerability. That means that when a user installs software, moves files such as CD/DVD ROMs or plugs in flash drives those items can all be thought of as hardware vulnerabilities, as can interfaces on the board by which the drives are connected. Securing physical access by locking any rooms, cabinets and cases housing computer equipment protects against this type of vulnerability.

Another type of hardware vulnerability is an unexpected flaw in operation that allows attackers to gain control of a system by elevating privileges or executing code. These vulnerabilities can sometimes be exploited remotely, rather than requiring physical access.

Power analysis:

In such attack the attacker monitors the power consumption by giving the system different inputs and simply observing the consumption of power. In our case the attacker might consider the time the system takes to send and receive information from the service provider just by comparing the fluctuations.

Error analysis:

In such type of attack the hardware will be damaged purposely or be activated falsely to retrieve an output pattern for certain input. In our case the battery of cellphone can be damaged by heat or the contents can be rewritten. Output for same through a healthy module can be obtained, later on which the charging can be retrieved by comparing correct or false result.

Space requirement for unit Installation:

The space required for this system is bigger than the current charger hence it takes up a lot of space as compared to the traditional chargers. It is also not sometimes convenient to carry this charger everywhere we go as it is big to carry. Hence this can act as a challenge to the project to be very feasible.

Cannot connect more than one device:

We can connect only one device at a time. Hence we cannot charge multiple Devices at a single time and hence at time the charger as well as the switch board will be engaged with the one device that is connected to that device for the given time that is specified for charging.

Only Cell Phones can be charged:

Only cell phones can be charged by this Project. No othe electronic device other than cell phone can be charged like laptop or othe electronic devices. Hence this is one of the drawback of this project.

4.3.10 Test Case Design

Sr no	Activity Name	Input	Expected Output	Actual Output
1	Enter valid time in minutes section	Time: 2:300:56	The time you have entered in minutes section is invalid. Enter time in minutes section again	The time you have entered in minutes section is invalid. Enter time in minutes section again
2	Enter valid time in seconds section	Time: 2:30:500	The time you have entered in seconds section is invalid. Enter time in seconds section again	The time you have entered in seconds section is invalid. Enter time in seconds section again

Table 3:Test Case Table

Chapter 5

Implementation and Testing

5.1 Implementation Approaches

Modules Used

- **Arduino Module**

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (For prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers can be programmed using C and C++ programming languages. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy,[2] aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors.

Circuit Diagram:

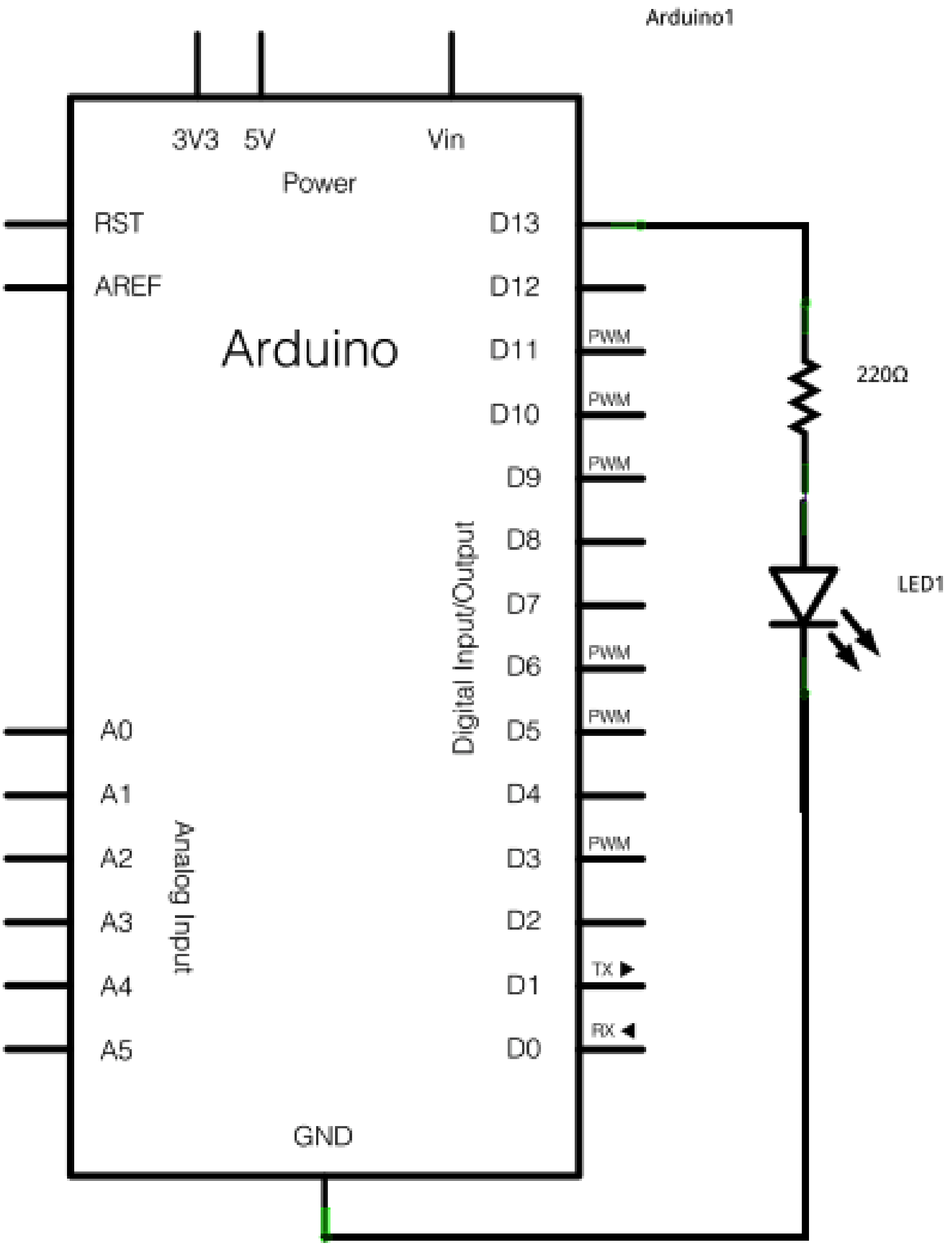


Figure 17:Arduino Pinout Diagram

Arduino Pinouts:

- **VIN Pin** - This pin is used to power the Arduino Uno board using an external power source. The voltage should be within the range mentioned above.

- **GND**-In the Arduino Uno pinout, you can find 5 GND pins, which are all interconnected.The GND pins are used to close the electrical circuit and provide a common logic reference level throughout your circuit. Always make sure that all GNDs are connected to one another and have a common ground.
- **RESET** - resets the Arduino
- **IOREF** - This pin is the input/output reference. It provides the voltage reference with which the microcontroller operates.
- Arduino Pins A0-A5 are capable of reading analog voltages. On Arduino the ADC has 10-bit resolution, meaning it can represent analog voltage by 1,024 digital levels. The ADC converts voltage into bits which the microprocessor can understand.
- Pins 0-13 of the Arduino Uno serve as digital input/output pins.
- In the Arduino Uno - pins 3,5,6,9,10,11 have PWM capability.
- **Serial (TTL) -** Digital pins 0 and 1 are the serial pins of the Arduino Uno.
- **SPI -** SS/SCK/MISO/MOSI pins are the dedicated pins for SPI communication. They can be found on digital pins 10-13 of the Arduino Uno and on the ICSP headers.
- **I2C -** SCL/SDA pins are the dedicated pins for I2C communication. On the Arduino Uno they are found on Analog pins A4 and A5.
- **Aref -** Reference voltage for the analog inputs.
- **Interrupt -** INT0 and INT1. Arduino Uno has two external interrupt pins.
- **External Interrupt -** An external interrupt is a system interrupt that occurs when outside interference is present. Interference can come from the user or other hardware devices in the network. Common uses for these interrupts in Arduino are reading the frequency a square wave generated by encoders or waking up the processor upon an external event.

- **Relay Module**

A **relay** is an electrically operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations thereof. Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. Relays were first used in long-distance telegraph circuits as signal repeaters: they refresh the signal coming in from one circuit by transmitting it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations. The traditional form of a relay uses an electromagnet to close or open the contacts, but other operating principles have been invented, such as in solid-state relays which use semiconductor properties for control without relying on moving parts. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called *protective relays*. Latching relays require only a single pulse of control power to operate the switch persistently. Another pulse applied to a second set of control terminals, or a pulse with opposite polarity, resets the switch, while repeated pulses of the same kind have no effects. Magnetic latching relays are useful in applications when interrupted power should not affect the circuits that the relay is controlling.

Relay Module Pinout:

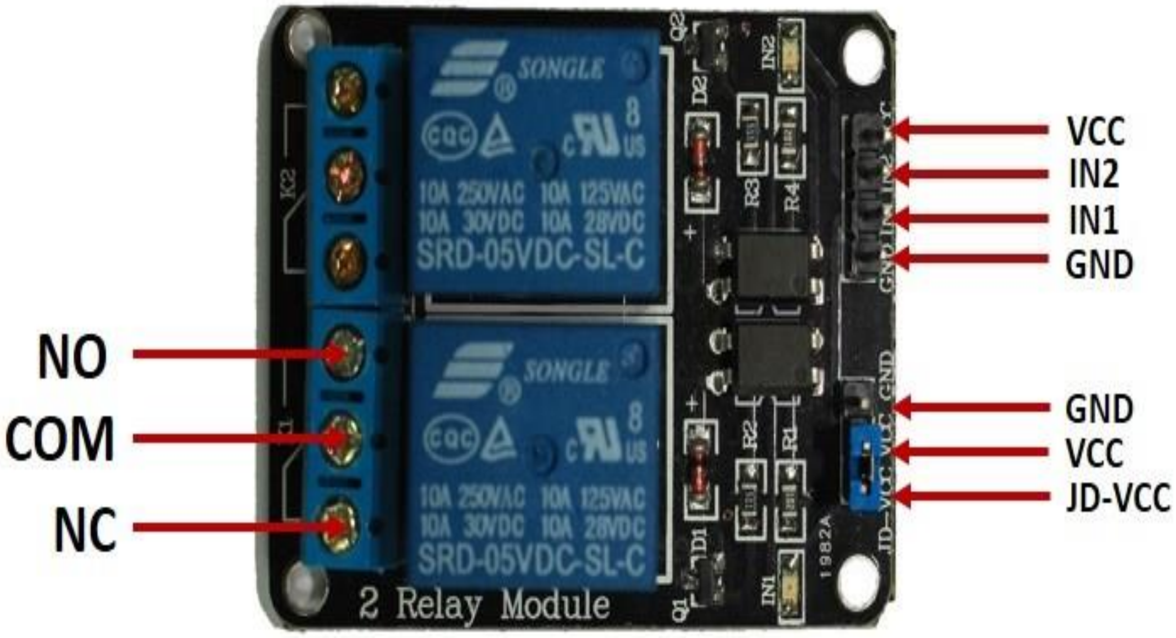


Figure 18: Relay Module Pinout Diagram

- **COM**: common pin
- **NC (Normally Closed):** the normally closed configuration is used when you want the relay to be closed by default, meaning the current is flowing unless you send a signal from the Arduino to the relay module to open the circuit and stop the current.
- **NO (Normally Open):** the normally open configuration works the other way around: the relay is always open, so the circuit is broken unless you send a signal from the Arduino to close the circuit.
- **GND**: goes to ground
- **IN1**: controls the first relay (it will be connected to an Arduino digital pin)
- **IN2**: controls the second relay (it should be connected to an Arduino digital pin if you are using this second relay. Otherwise, you don’t need to connect it)
- **VCC**: goes to 5V
- The set at the right consists of **VCC** and **GND** to power up the module, and input 1 (**IN1**) and input 2 (**IN2**) to control the bottom and top relays, respectively.The second set of pins consists of **GND**, **VCC**, and **JD-VCC** pins. The JD-VCC pin powers the electromagnet of the relay.

LCD Module:

A **liquid-crystal display** (**LCD**) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome.[1] LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and seven-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made from a matrix of small pixels, while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement. For example, a character positive LCD with a backlight will have black lettering on a background that is the color of the backlight, and a character negative LCD will have a black background with the letters being of the same color as the backlight. Optical filters are added to white on blue LCDs to give them their characteristic appearance.

LCDs are used in a wide range of applications, including LCD televisions, computer monitors, instrument panels, aircraft cockpit displays, and indoor and outdoor signage. Small LCD screens are common in portable consumer devices such as digital cameras, watches, calculators, and mobile telephones, including smartphones. LCD screens are also used on consumer electronics products such as DVD players, video game devices and clocks. LCD screens have replaced heavy, bulky cathode ray tube (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and plasma displays, with LCD screens available in sizes ranging from tiny digital watches to very large television receivers. LCDs are slowly being replaced by OLEDs, which can be easily made into different shapes, and have a lower response time, wider color gamut, virtually infinite color contrast and viewing angles, lower weight for a given display size and a slimmer profile (because OLEDs use a single glass or plastic panel whereas LCDs use two glass panels; the thickness of the panels increases with size but the increase is more noticeable on LCDs) and potentially lower power consumption (as the display is only "on" where needed and there is no backlight).

LCD Module Pinout Diagram:

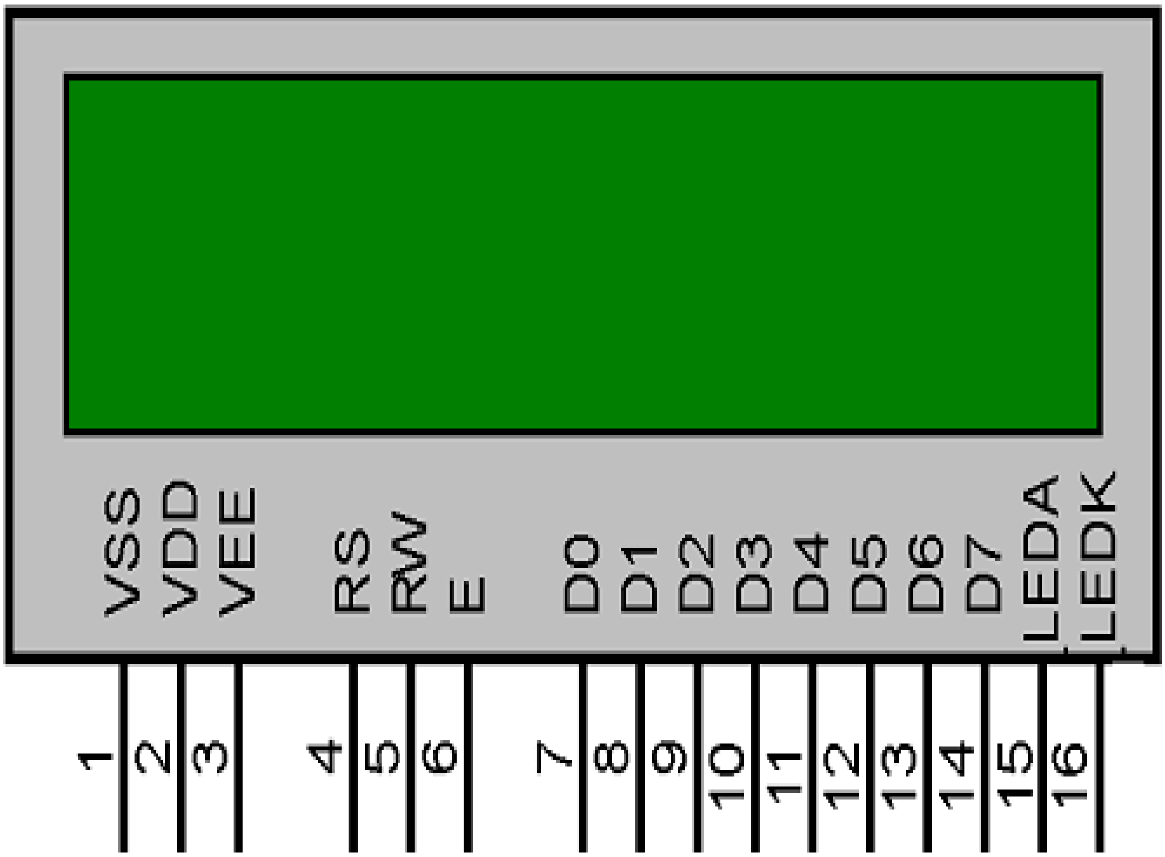


Figure 19:LCD Display Pinout Diagram

Sr. No	Pin No.	Pin Name	Pin Type	Pin Description
1	Pin 1	Ground	Source Pin	This is a ground pin of LCD
2	Pin 2	VCC	Source Pin	This is the supply voltage pin of LCD
3	Pin 3	V0/VEE	Control Pin	Adjusts the contrast of the LCD.
4	Pin 4	Register Select	Control Pin	Toggles between Command/Data Register
5	Pin 5	Read/Write	Control Pin	Toggles the LCD between Read/Write Operation
6	Pin 6	Enable	Control Pin	Must be held high to perform Read/Write Operation

7	Pin 7-14	Data Bits (0-7)	Data/Command Pin	Pins used to send Command or data to the LCD.
8	Pin 15	LED Positive	LED Pin	Normal LED like operation to illuminate the LCD
9	Pin 16	LED Negative	LED Pin	Normal LED like operation to illuminate the LCD connected with GND.

Table 4:LCD Display Pinout

- **Rotary Encoder Module:**

A **rotary encoder**, also called a **shaft encoder**, is an electro-mechanical device that converts the angular position or motion of a shaft or axle to analog or digital output signals.There are two main types of rotary encoder: absolute and incremental. The output of an absolute encoder indicates the current shaft position, making it an angle transducer. The output of an incremental encoder provides information about the *motion* of the shaft, which typically is processed elsewhere into information such as position, speed and distance.Rotary encoders are used in a wide range of applications that require monitoring or control, or both, of mechanical systems, including industrial controls, robotics, photographic lenses,[2] computer input devices such as optomechanical mice and trackballs, controlled stress rheometers, and rotating radar platforms.

Rotary Encoder pinout diagram:

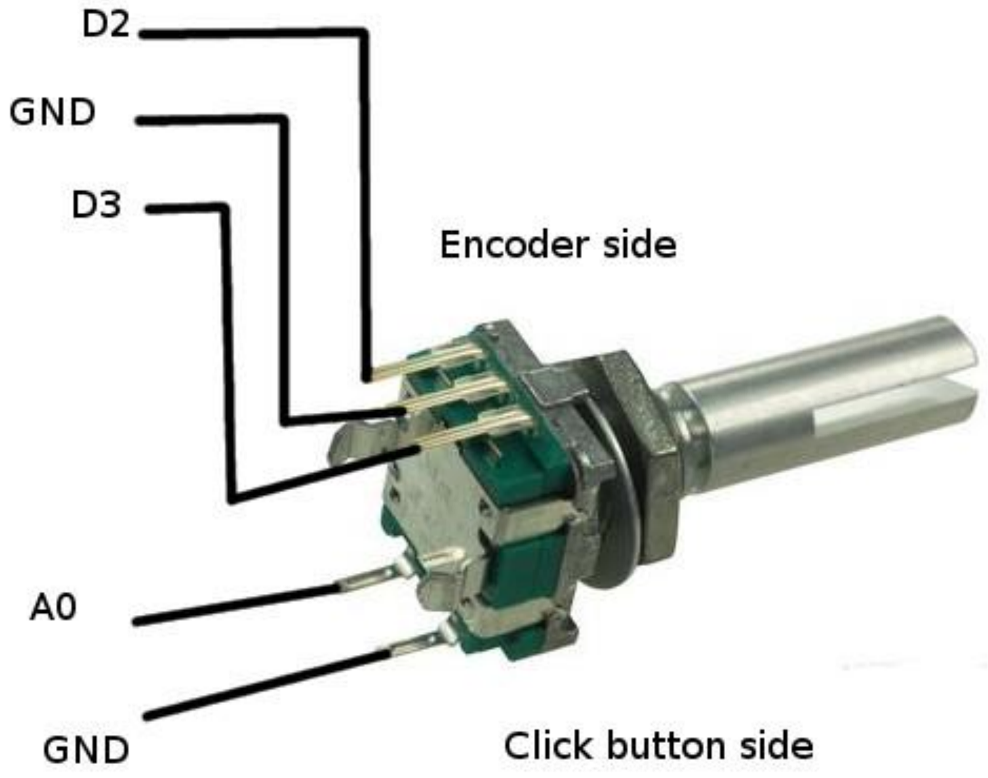


Figure 20:Rotary Encoder Pinout Diagram

Pin Name	Description
GND	Connected to GROUND
+	Connected to +5V
SW	Output of internal button
DT	Contact A output or DATA
CLK	Contact B output or CLOCK

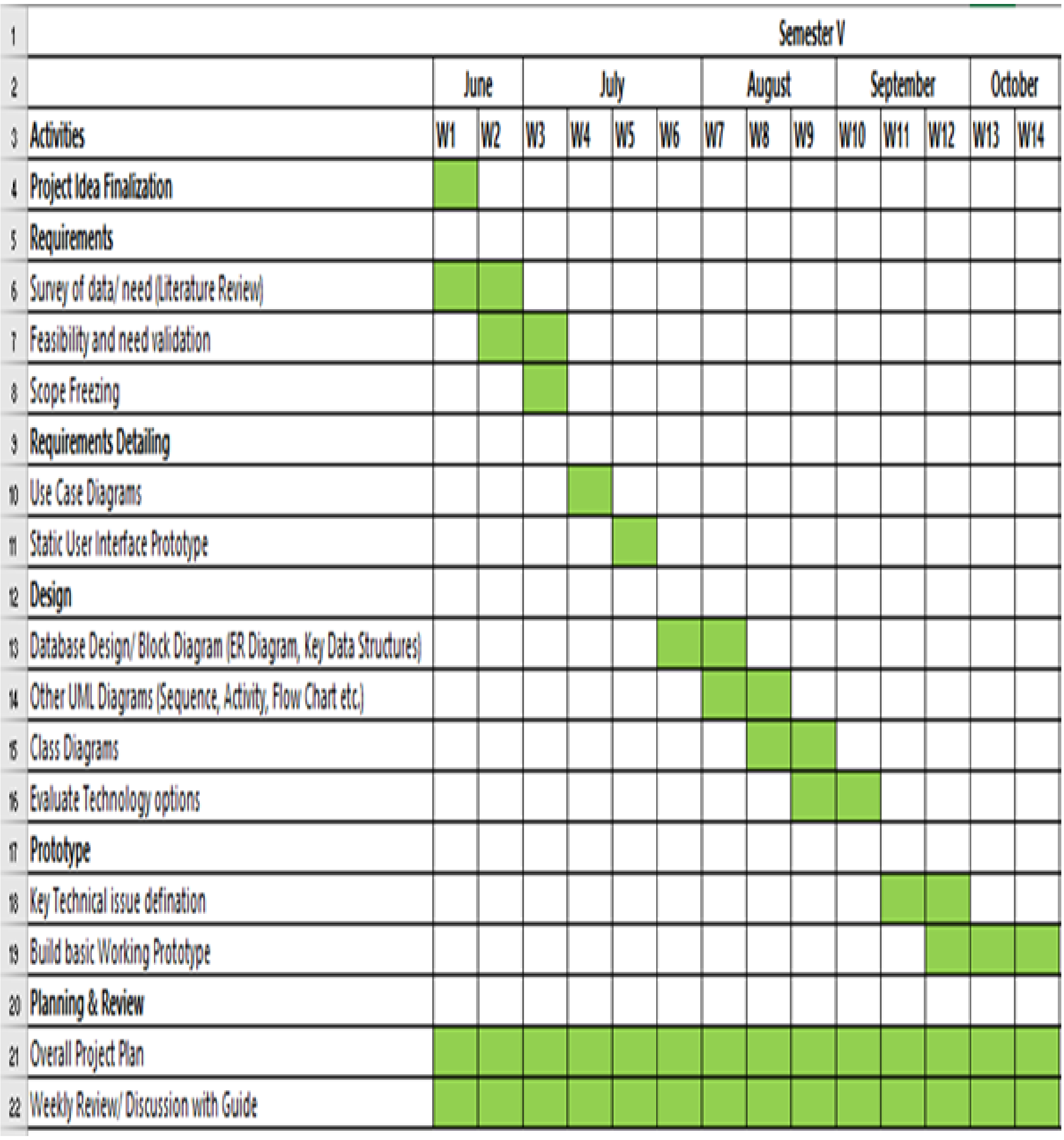
Table 5:Rotary Encoder Pinout

5.1 Implementation Approaches:

Sr.No.	Implementation Approaches	Action	
1.	Module	Arduino Relay Module LCD Display Rotary Encoder	
2.	Percentage Complete	Arduino Module-100% Relay Module-100% LCD Display Module-100% Rotary Encoder Module-100%	
3.	Status	Arduino Module-Completed Relay Module-Completed LCD Display Module-Completed Rotary Encoder Module-Completed	
4.	Day Started	18th December 2019	
5.	Day to be completed	8th March 2020	
6.	Actual Completion Date	15th March 2020	
7.	Module assignment	Arduino Module	Nikhilesh Pednekar
		Arduino Module	Gaurang Supal
		Relay Module	Nikhilesh Pednekar
		Relay Module	Gaurang Supal
		LCD Module	Nikhilesh Pednekar
		LCD Module	Gaurang Supal
		Rotary Encoder	Nikhilesh Pednekar
		Rotary Encoder	Gaurang Supal
8.	Importance of Module	Arduino Module - High Priority Relay Module – Medium Priority LCD Module – Medium Priority Rotary Encoder Module – Low Priority	

Table 6:Implementation Approaches Table

Gantt Chart:



1		Semester V													
2		June		July				August			September			October	
3	**Activities**	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
4	**Project Idea Finalization**	🟩													
5	**Requirements**														
6	Survey of data/ need (Literature Review)	🟩	🟩												
7	Feasibility and need validation		🟩	🟩											
8	Scope Freezing			🟩											
9	**Requirements Detailing**														
10	Use Case Diagrams				🟩										
11	Static User Interface Prototype					🟩									
12	**Design**														
13	Database Design/ Block Diagram (ER Diagram, Key Data Structures)						🟩	🟩							
14	Other UML Diagrams (Sequence, Activity, Flow Chart etc.)							🟩	🟩						
15	Class Diagrams								🟩	🟩					
16	Evaluate Technology options									🟩	🟩				
17	**Prototype**														
18	Key Technical issue defination											🟩	🟩		
19	Build basic Working Prototype												🟩	🟩	🟩
20	**Planning & Review**														
21	Overall Project Plan	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
22	Weekly Review/ Discussion with Guide	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Table 7: Complete Project Gantt Chart

5.2 Coding Details and Code Efficiency

5.2.1 The following code is for Arduino module:

```
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

#define outputA A1
#define outputB A2
#define button A3
#define charger A5
int counter = 0;
int aState;
int aLastState;
int hr=0,minu=0,sec=0;
void setup() {
  lcd.begin(16,2);
  pinMode (outputA,INPUT);
  pinMode (outputB,INPUT);
  pinMode (button,INPUT);
  pinMode (charger,OUTPUT);
  digitalWrite(charger,HIGH);
  Serial.begin (9600);
  aLastState = digitalRead(outputA);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SET TIMER:");
  lcd.setCursor(0,1);
  lcd.print("HRS : MIN  ");
  lcd.print(hr);lcd.print(":");lcd.print(minu);
}
```

```
void loop() {
  aState = digitalRead(outputA);
  if (aState != aLastState){
    if (digitalRead(outputB) != aState) {
      counter ++;
    } else {
      counter --;
      if(counter<0)
      counter=0;
    }
    hr=floor(counter/60);
    minu=counter%60;
    Serial.print("SET TIMER: ");
    Serial.println(counter);
    Serial.print(" HR:");Serial.print(hr);Serial.print(" MIN:");
    Serial.print(minu);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("SET TIMER:");
    lcd.setCursor(0,1);
    lcd.print("HRS : MIN  ");
    lcd.print(hr);lcd.print(":");lcd.print(minu);
  }
  if(digitalRead(button)==1&&counter>0)
  {
   Serial.println("Button pressed");
   start_time();
   delay(50);
  }
  aLastState = aState;
}

void start_time()
 {
 sec=59;
 counter--;
 digitalWrite(charger,LOW);
 while(counter>0||sec>-1)
 {
   if(sec<0)
   {
     counter--;
     sec=59;
   }
   delay(1000);
   hr=floor(counter/60);
   minu=counter%60;
   Serial.print("COUNTDOWN :");Serial.println(counter);
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("COUNTDOWN : ");
   lcd.setCursor(0,1);
   lcd.print(hr);lcd.print(":");lcd.print(minu);lcd.print(":");
   lcd.print(sec);
   sec--;
 }
 setup();
}
```

5.3 Testing Approaches

Testing approaches are based on the Scheme Design. In Scheme Designing phase we consult with the owner/customer to determine project goals and requirements. It work like a policy maker

Testing Approach tells us how to test the entire project.

The different types of testing approaches are:

5.3.1 Unit Testing

Unit testing is done to remove problems/errors from the code by breaking your program into different parts/submodules and testing each part.

5.3.2 Integration Testing

In Integration testing individual units are combined and tested for proper interfacing between different components and to expose faults in the interaction between these integrated units.

5.3.3 **Usability Testing**

There are so many devices of different shape and form factors are used by the users. Moreover, the perception also varies from one user to other. That's why checking usability of the system is very important in IoT testing.

5.3.4 **Compatibility Testing**

There are lots of devices which can be connected though IOT system. These devices have varied software and hardware configuration. Therefore, the possible combination are huge. As a result, checking the compatibility in IOT system is important.

5.3.5 **Reliability and Scalability Testing**

Reliability and Scalablity is important for buidling an IOT test enviorment which involves simulation of sensors by utilzing virtualization tools and technolgoies.

5.3.6 **Security testing**

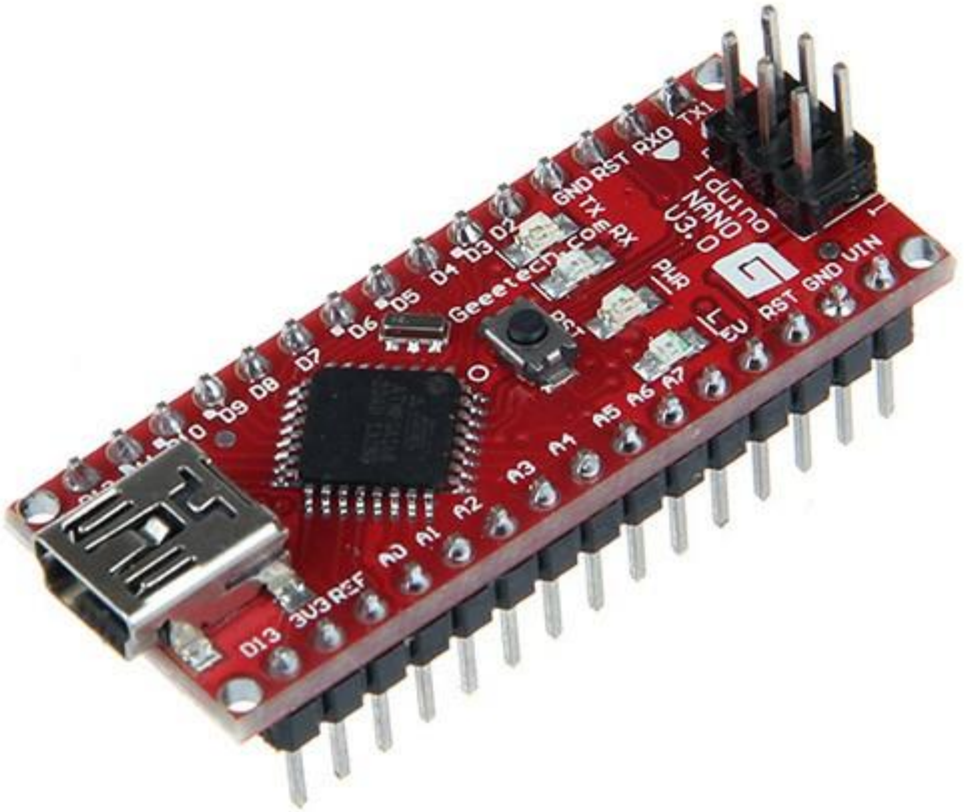
In the IOT environment, there are many users are accessing a massive amount of data. Thus, it is important to validate user via authentication, have data privacy controls as part of security testing.

5.3.7 **Performance Testing**

Performance testing is important to create strategic approach for developing and implementing an IOT testing plan.

5.4 Modification and Improvements

- First we were about to use arduino uno for the project but as it was not functioning properly so we used arduino nono in order to run our project. Arduino nano is another form of arduino uno but only in smaller size. The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.



Specifications:

Microcontroller Atmel ATmega168 or ATmega328

Operating Voltage: (logic level) 5 V

Input Voltage: (recommended) 7-12 V

Input Voltage: (limits) 6-20 V

Digital I/O Pins: 14 (of which 6 provide PWM output)

Analog Input Pins: 8

DC Current per I/O Pin: 40 mA

Flash Memory: 16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader

SRAM: 1 KB (ATmega168) or 2 KB (ATmega328)

EEPROM: 512 bytes (ATmega168) or 1 KB (ATmega328)

Clock Speed: 16 MHz

Dimensions: 0.73" x 1.70"

- We also changed the code according to the deamnds as the previous code that we have used was giving us errors reguarding the lcd display. By using the previous code the LCD Display was not showing any output as the LCD Display required higher input power supply and the code we used had said that the power supply should be directly from the arduino. So in order to fight this defect we had to change the code accoringly

Chapter 6

Results and Discussion

6.1 Test Reports

Its is a set of rules or Conditions to check if the system works according to requirement. It can be termed as set of details which include results, information and soon.

Test Module	Test Case	Expected Result	Test Module
LCD	Provides Details about the times entered in Hours section	The time you have entered in Hours section is valid.	PASS
LCD	Provides Details about the times entered in Minutes section	The time you have entered in Minutes section is valid.	PASS
LCD	Provides Details about the times entered in Seconds section	The time you have entered in Seconds section is valid.	PASS

Chapter 7

Conclusions

7.1 Conclusion

It has been a great pleasure for me and my partner to work on this exciting and challenging project. This project proved good for us as it provided practical knowledge of not only programming in Arduino IDE but also in hardware devices like Arduino and also procedures related to Arduino UNO.

It also provided us knowledge about the latest technology used in developing an application and technology that will be great demand in future.

The wider areas of this project is to charge his phone for a particular time period by setting the timer for charging the device. Every battery has a limit to the number of time it can be charged (known as charge cycles). Also temperature plays an important role in the life of a battery. Higher temperatures might disrupt the chemistry of the battery.

The time for which we have set the charger to charger the device is displayed on the LCD screen. If there is a power failure, the remaining time is stored in the memory and when the power comes back, it will prompt you whether to continue with the countdown or to set a new time.It is also useful for the people who use their phone heavily and for those who have the habit of charging their phones at night.

7.2 Limitation

- **Space requirement for unit Installation:**

  The space required for this system is bigger than the current charger hence it takes up a lot of space as compared to the traditional chargers. It is also not sometimes convenient to carry this charger everywhere we go as it is big to carry. Hence this can act as a challenge to the project to be very feasible.

- **Cannot connect more than one device:**

  We can connect only one device at a time. Hence we cannot charge multiple Devices at a single time and hence at time the charger as well as the switch board will be engaged with the one device that is connected to that device for the given time that is specified for charging.

- **Only Cell Phones can be charged:**

  Only cell phones can be charged by this Project. No othe electronic device other than cell phone can be charged like laptop or othe electronic devices. Hence this is one of the drawback of this project.

7.3 Future Scope of Project

Scope of the this project are as follows :

- Better and efficient charging experience.
- It provides a safe environment for charging phones, as it cuts off the power supply to the charger, thus reducing the risk of overheating and this can help to preserve the battery life and also prevent the cell phone blasts due to overheating of batteries.

7.3.1 References

- “wisegeek.com”: What is a Smart Battery Charger?”. Archived from the original on 18th April 2014. “... The smart batteries contain special microchips that are programmed to communicate with a smart battery charger of the same brand...”
- Modular Approach for continuous cell-level balancing to improve performance of Large Battery Packs, National Renewable Energy Laboratory, September 2014.
- “US Patent 5475294: Charge Controller for battery.” Free patentsonline.com
- Linear & Switching Voltage Regulator Handbook; ON Semiconductor; 2002; HB206/D.
- Saxon Wolfgang (6 July,1993). “Clarenie M Zener, Physicist & Prof at Carnegie Mellon”. The New York Times.
- Beginning C for Arduino: Learn C Programming for the Arduino & compatible Microcontrollers; Jack Purdum;2012.

7.3.2Bibliography

- search?client=firefox-b-d&q=nevon+projects
- /results?search_query=smart+phone+charging+monitoring+system
- search?client=firefox-b-d&q=wikipedia
- index.php/Arduino_Nano
- projecthub/aleksand1975/intelligent-charger-for-9v-nimh-rechargeable-batteries-v1-61cbd6?ref=search&ref_id=charger&offset=0

7.3.3Websites Used

- https://medium.com/
- https://www.mpoweruk.com/
- https://www.sbsbattery.com/
- https://hal.archives-ouvertes.fr/
- https://energicplus.com/
- https://www.elprocus.com/

7.3.4 APPENDICES

- "wisegeek.com": What is a Smart Battery Charger?". Archived from the original on 18th April 2014. "... The smart batteries contain special microchips that are programmed to communicate with a smart battery charger of the same brand..."

- Modular Approach for continuous cell-level balancing to improve performance of Large Battery Packs, National Renewable Energy Laboratory, September 2014.

- "US Patent 5475294: Charge Controller for battery." Free patentsonline.com

- Linear & Switching Voltage Regulator Handbook; ON Semiconductor; 2002; HB206/D.

- Saxon Wolfgang (6 July,1993). "Clarenie M Zener, Physicist & Prof at Carnegie Mellon". The New York Times.

- Beginning C for Arduino: Learn C Programming for the Arduino & compatible Microcontrollers; Jack Purdum;2012.

7.3.5 GLOSSARY

- Arduino UNO : UNO means one in Italy
- IDE : Integrated Development Environment
- AVR : Alf & Vegard's RISC processor