SYNOPSIS

TITLE – AUTOMATIC PET FEEDER USING ARDUINO

Aim

The aim of an automatic pet feeder using Arduino is to enhance the well-being of pets and provide convenience and peace of mind to their owners. It focuses on providing a convenient and reliable way to feed pets, particularly when the owners are away or have busy schedules. By leveraging Arduino technology, these feeders can be programmed to dispense food at scheduled times, ensuring that pets receive regular meals even when their owners are away or occupied with other tasks. Additionally, Arduino-based feeders offer the flexibility to customize feeding schedules and portion sizes according to the specific dietary needs of each pet. This not only helps maintain their health and weight but also reduces the risk of overfeeding or underfeeding. These innovative devices is to promote the well-being of pets by providing them with consistent, balanced nutrition while offering convenience and peace of mind to their owners. This project aims in rectifying the problems caused by existing feeders and designing an efficient automatic pet feeders using internet of things.

Objective

The objective of this project is to create an automatic feeding machine for pets feeding.

- This project is designed keeping the view of dairy farms, poultry farms and pets at home it is important to maintain the diet of animals just like human being from keeping them healthy for better production and good quality of milk in the case of dairy farm and eggs and chicken in the case of poultry farm.
- Now a days, everyone can have a pet at home without giving their full commitment to have a healthy pet. With this feeding machine, it will help pet owner to manage their pet diet wheel. When user is at home, it can be controlled by a mobile application through internet. If user is not at home, user can set the timer to feed their pet.
- To make sure that the food does not exceed force sensor will active and detect the exact amount should be in the bowl. Whenever we go to work or are away on vocation. We always end up paying so much money for pet sitters to feed our pets.
- By adapting a pet feeder to an IOT application would not only solve our problem but would also benefit other pet owners.

Introduction

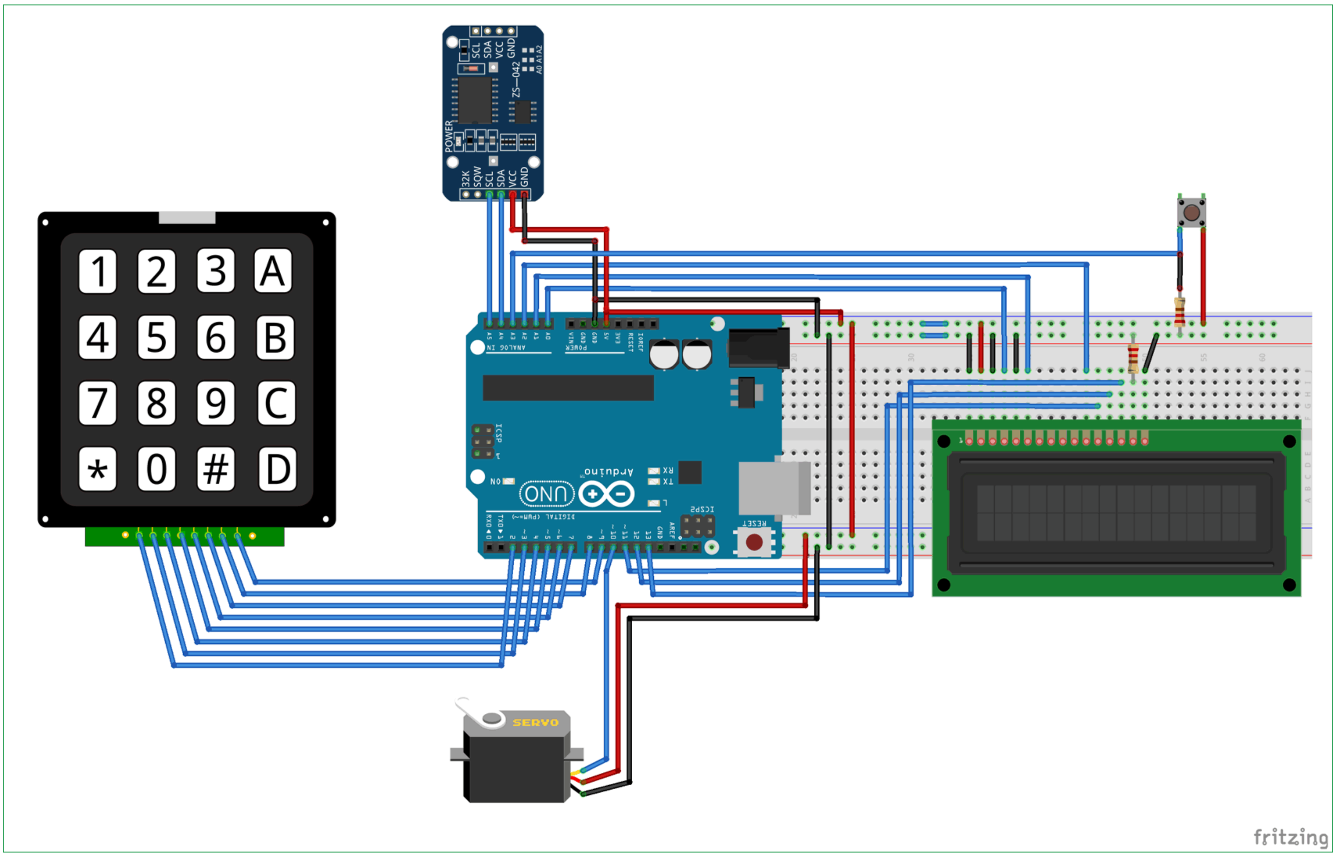
Automatic pet feeder is one of the new technologies for feeding pet. It will help pet owner to take care of their pet while they are not at home. Even the owners are not at home, they still can feed their pet. Automatic pet feeder is built to help pet owner taking care of their pet. IoT pet feeder is one of the pet feeders that will be controlled by a mobile application through internet. The automatic pet feeder will automatically dispense predetermined amount of food and water to the bowls. As pet lovers, user should understand those pets also need a proper diet management. Whether user away from home unexpectedly or simply would like one less chore to worry about, user can feel secure that the beloved pet will be cared for and fed on time every time. The Automatic pet feeder will solve two problems which pet owners face i.e., making sure that each pet has access to a healthy amount of food throughout the day, regardless of the owner's schedule. Making sure that each pet eats only its own food though there are a variety of products on the market which solve the first problem, there are none which address the second. The automatic pet feeder will give pet owners a solution to both problems, thereby improving the lives of both pets and owners by allowing the owner to reliably provide food to a pet at the time the owner wishes and keep the pet from reaching the food stored for later feedings. Many animal feed systems can be designed to function as an automatic device that allow the user to feed whenever he wishes from anywhere through internet. The purpose of having sensors in a system like this is to automate the feed process completely with less human interference.

Description

Arduino based Automatic Pet Feeder which can automatically serve food to your pet timely. It has a DS3231 RTC (Real Time Clock) Module, which used to set time and date on which your pet should be given food. So, by setting up the time according to your pet's eating schedule, the device drop or fill the food bowl automatically. In this circuit, we are using a 16*2 LCD to display the time using DS3231 RTC Module with Arduino UNO. Also, a servo motor is used to rotate the containers to provide the food and 4*4 matrix keypad to manually set up the time for feeding the Pet. You can set the rotation angle and container opening duration according to the quantity of food you want to serve to your pet. The quantity of food may also depend upon your pet whether it's a dog, cat or bird.

Circuit Diagram

In this Arduino based Pet Feeder, for Getting Time and Date, we have used RTC (Real Time Clock) Module. We have used the 4*4 Matrix Keypad to set the Pet's eating time manually with the help of 16x2 LCD. The Servo motor rotates the container and drop the food on the time set by the user. The LCD is used for displaying the Date and Time. Complete working can be found in the Video given at the end.



DS3231 RTC Module

DS3231 is a RTC (Real Time Clock) module. It is used to maintain the date and time for most of the Electronics projects. This module has its own coin cell power supply using which it maintains the date and time even when the main power is removed or the MCU has gone through a hard reset. So once we set the date and time in this module it will keep track of it always. In our circuit, we are using DS3231 to feed the pet according to the time, set up by the Pet's owner, like an alarm. As, clock reaches to the set time, it operates the servo motor to open the container gate and the food drops in the Pet's food bowl.

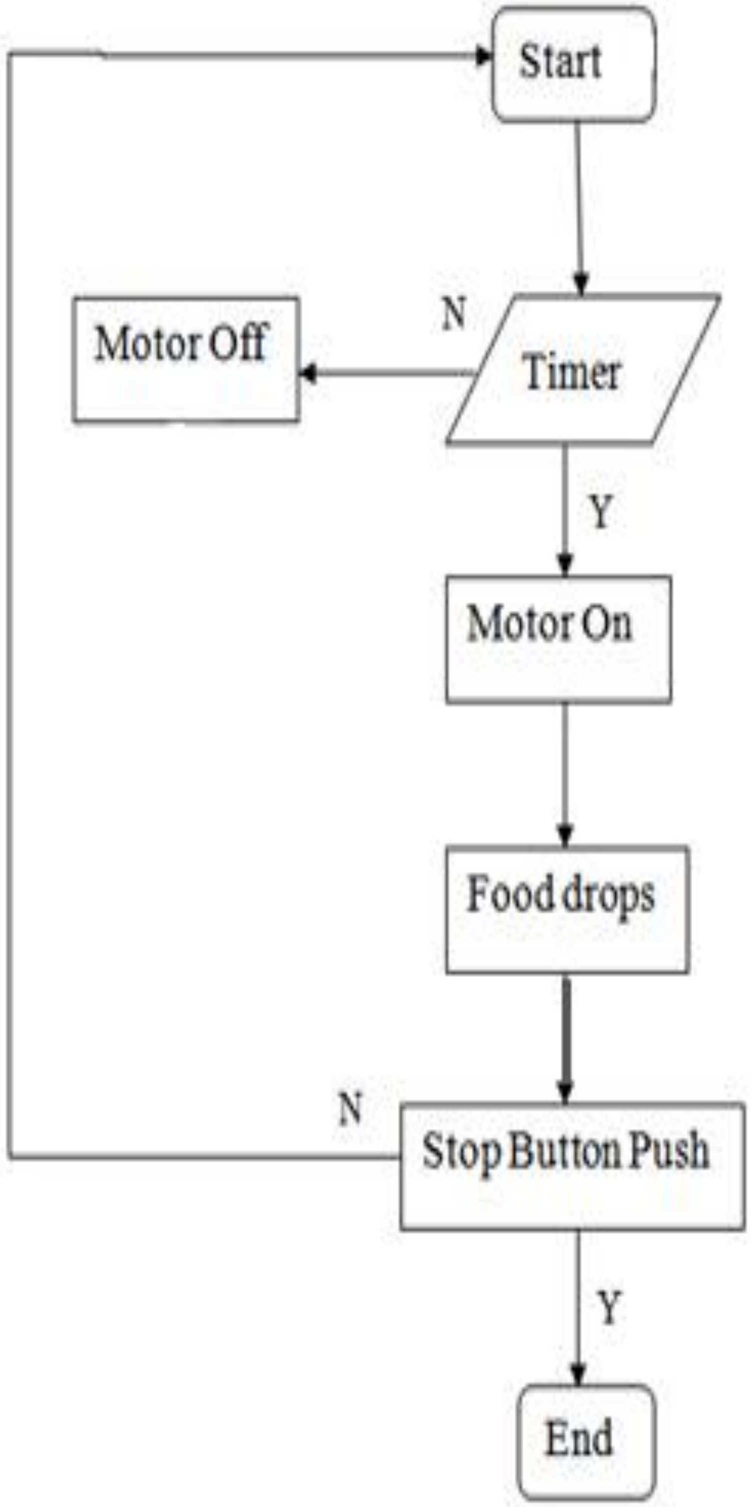
Arduino have default libraries for using the Servo motor and LCD 16*2 with it. But for using DS3231 RTC Module and 4*4 Matrix Keypad with the Arduino, the libraries are given below:

- DS3231 RTC (Real Time Clock) Module Library
- 4*4 Matrix Keypad Library

The DS3231 RTC (Real-Time Clock) module is a highly accurate timekeeping device commonly used in electronic projects requiring time-sensitive functions. DS3231 IC: At the heart of the module is the DS3231, a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. This IC provides highly precise timekeeping, with an accuracy of a few seconds per year.

The DS3231 RTC module offers highly accurate timekeeping, temperature-compensated performance, and convenient interfacing via the I2C protocol. These features make it an ideal choice for applications requiring precise timekeeping, such as clocks, data loggers, timers, and automated systems.

Data Flow diagram for Automatic Pet Feeder



Requirements Specifications

Hardware Components

1. **Arduino UNO:** The Arduino UNO is the microcontroller board that serves as the brain of the automatic pet feeder. It runs the code to control the various components and manage the feeding schedule.
2. **4*4 Matrix Keypad:** The keypad provides an interface for the user to input commands such as setting feeding times or adjusting portion sizes. It consists of a grid of buttons that can be pressed to send signals to the Arduino.
3. **16*2 LCD:** The LCD (Liquid Crystal Display) provides visual feedback to the user by displaying information such as current settings, feeding schedule, or error messages. It enhances the user interface by providing a means of communication between the user and the pet feeder.
4. **Push Button:** The push button(s) can be used for various functions such as starting or stopping the feeding process, confirming settings, or activating manual feeding mode. They provide tactile feedback to the user.
5. **Servo Motor:** The servo motor controls the feeding mechanism of the pet feeder. It rotates a specific angle to dispense the pet food from the hopper into the pet's bowl. The rotation angle can be controlled precisely, allowing for accurate portion control.
6. **Resistor:** Resistors are used to limit the current flow in the circuit and protect the components from damage. They may be used in various parts of the circuit where necessary to ensure proper operation.
7. **Connecting Wires:** Connecting wires are used to establish electrical connections between the different components of the pet feeder, allowing them to communicate with the Arduino and each other.
8. **Breadboard:** The breadboard provides a convenient platform for prototyping and connecting the electronic components of the pet feeder without soldering. It allows for easy testing and modification of the circuit layout.

Software components

1. **Arduino Sketch:** The program written in Arduino IDE that controls the operation of the feeder. It includes code to schedule feeding times, control the feeding mechanism, interact with the user interface, and handle sensor inputs if applicable.
2. **Libraries:** Arduino libraries may be used to simplify interfacing with components such as RTC modules, motors, or sensors.
3. **Timing Functions:** Functions to manage timing and scheduling of feeding times based on the current time provided by the RTC module.
4. **Control Logic:** Logic to control the feeding mechanism based on user-defined settings and sensor inputs if used.
5. **User Interface Handling:** Code to handle user input from buttons, switches, or other interface elements, and display feedback on an LCD display if used.
6. **Error Handling:** Error detection and handling mechanisms to deal with situations such as food jams or low battery levels.
7. **Optional Features:** Additional features such as remote monitoring and control via Wi-Fi or Bluetooth, data logging of feeding activity, or integration with a mobile app for enhanced user experience.

Conclusion

An automatic pet feeder using Arduino offers a practical and efficient solution for pet owners to ensure their pets receive timely and controlled food, even when they are away from home. By integrating components such as Arduino UNO, keypad, LCD, servo motor, and push buttons, this system provides a user-friendly interface for setting feeding schedules and portion sizes. The servo motor precisely dispenses food from the hopper into the pet's bowl, promoting portion control and preventing overfeeding or underfeeding. Additionally, the inclusion of features like resistors, connecting wires, and breadboard facilitates easy assembly and customization of the feeder. Overall, this automated system enhances pet care by providing convenience, consistency, and peace of mind to pet owners, while ensuring the health and well-being of their beloved companions.

For future enhancements of a pet feeder, several avenues can be explored to improve functionality, convenience, and user experience. Here are some ideas:

1.Smart Integration:

- Incorporate advanced connectivity features, such as integration with smart home systems like Amazon Alexa or Google Home, allowing users to control the feeder via voice commands.
- Develop a dedicated smartphone app with features for remote monitoring, scheduling adjustments, and receiving notifications about feeding events or issues.

2.Machine Learning Algorithms:

- Implement machine learning algorithms to analyse pet feeding patterns and adjust feeding schedules or portion sizes automatically based on the pet's behaviour and dietary needs.
- Utilize image recognition technology to identify the pet using the feeder and provide personalized feeding recommendations or alerts to the owner.

3.Health Monitoring:

- Integrate sensors to monitor the pet's health indicators, such as weight, activity level, or eating habits, and provide insights to the owner or veterinarian for early detection of health issues.
- Include features for tracking food consumption over time, allowing users to monitor changes in their pet's eating behaviour and detect potential health concerns.

4.Modular Design:

- Design the feeder with modular components, allowing users to customize and expand its functionality, such as adding additional food compartments for multi-pet households or integrating water dispensers.
- Enable compatibility with accessories or attachments for specialized diets or feeding requirements, such as slow-feed bowls or food dispensing puzzles.

INTRODUCTION

CHAPTER - 1

INTRODUCTION

Introduction to IoT (Internet of Things): The Internet of Things (IoT) refers to the network of interconnected devices embedded with sensors, software, and other technologies that enable them to collect and exchange data over the internet. These devices can range from everyday objects such as household appliances, wearable devices, and industrial machinery to sophisticated instruments like autonomous vehicles and smart city infrastructure.

IoT stands for Internet of Things. It refers to the network of physical devices, vehicles, home appliances, and other items embedded with sensors, software, and connectivity, enabling them to connect and exchange data over the internet. These devices are typically equipped with sensors and actuators that allow them to gather and transmit data, as well as respond to commands. The concept behind IoT is to create a network of interconnected devices that can communicate, collect, and exchange data with each other autonomously, without human intervention. This enables various applications and services, such as smart homes, industrial automation, healthcare monitoring, environmental monitoring, and more.

IoT has the potential to revolutionize many aspects of daily life and business operations by enabling greater automation, efficiency, and insights derived from data collected by these connected devices. However, it also raises concerns about privacy, security, and the ethical implications of ubiquitous data collection and monitoring.

1.1 Overview of the Project

An automatic pet feeder aims to create a device that reliably dispenses pet food at scheduled times, ensuring pets are fed regularly even in the owner's absence. This project typically involves a combination of mechanical and electronic components, including a food container, a motorized dispensing mechanism, a microcontroller (like Arduino or Raspberry Pi), a real-time clock for scheduling, and sensors to monitor food levels. Key considerations include portion control, reliability, safety, ease of cleaning, and power backup. Advanced features might include remote control via a mobile app, voice command integration, and pet monitoring with a camera. The development process involves planning, component selection, prototyping, assembling the final product, software development, and thorough testing. The goal is to create a user-friendly, dependable feeder that meets the specific needs of the pet and the owner.

This pet feeder designed to cater to households with two pets, revolutionizes the way pet owners manage the dietary needs of their furry family members.

This feeder typically features two separate feeding stations, each with its own food compartment and dispenser. This innovative solution not only ensures that each pet receives the right amount of food at scheduled times but also addresses the unique challenges that arise when feeding multiple pets, such as dietary restrictions, competitive eating, and differing nutritional needs.

1.2 Aim

The aim of an automatic pet feeder project is to design and build a reliable, user-friendly device that automatically dispenses a predetermined amount of pet food at scheduled times. This ensures that pets receive consistent and timely meals, promoting their health and well-being, even when their owners are not home. The project seeks to incorporate features such as portion control, ease of use, and safety, with potential enhancements like remote operation via mobile apps, voice commands, and pet monitoring capabilities to provide convenience and peace of mind to pet owners.

1.3 Objectives

1. **Automated Feeding Schedule:** Develop a system that dispenses food at preset times, ensuring pets are fed regularly without human intervention.
2. **Health and Nutrition:** Ensures each pet receives the right amount of food, supports weight management, and adheres to dietary restrictions.
3. **Portion Control:** Ensure the feeder can accurately measure and dispense specific portions of food to prevent overfeeding or underfeeding.
4. **Ease of Use:** Create an intuitive user interface for easy programming and operation by pet owners, including straightforward setup and maintenance.
5. **Safety and Hygiene:** Ensure the feeder is safe for pets to use and easy to clean, preventing contamination and maintaining food hygiene.
6. **Customization and Flexibility:** Allow customization of feeding schedules and portions to cater to different pet types, sizes, and dietary needs.
7. **Cost-Effectiveness:** Aim for a cost-effective solution that balances functionality, quality, and affordability for the average pet owner.

1.4 Key features

1. **User Interface:** Easy-to-use controls, such as buttons and an LCD screen, for setting feeding times and portions, making it user-friendly for pet owners.
2. **Food Level Monitoring:** Sensors to detect and alert when the food level is low, ensuring the feeder is always ready to dispense the next meal.
3. **Ease of Cleaning:** Components that are easy to disassemble and clean to maintain hygiene and prevent contamination.
4. **LCD Display**: Show current time, feeding schedules, and remaining food.
5. **Dispenser Mechanism:** A reliable and robust mechanism (e.g., rotating auger, sliding gate) driven by a motor, designed to handle various types of pet food without jamming.
6. **Programmable Feeding Schedule:** Pet owners can program the feeder to dispense food at specific times, ensuring that pets are fed regular meals even when their owners are away or busy. This helps maintain a healthy eating routine and supports pets' overall well-being.
7. **Water Dispenser Integration:** Combine with a water dispenser to ensure your pet has both food and water.
8. **Real-Time Clock (RTC):** Keep accurate time to manage feeding schedules.

REQUIREMENT ANALYSIS

CHAPTER – 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Introduction

The Software Requirement Specification (SRS) outlines the software requirements for the development of a Pet Feeder system using Arduino. This document aims to define the functional and non-functional requirements necessary for the successful implementation of the software component of the automatic pet feeder.

2.2 Purpose

The purpose of this SRS is to provide a detailed description of the software functionalities, user interactions, and system behaviour required for the automatic pet feeder. It serves as a guideline for software developers, testers, and stakeholders involved in the project, ensuring a clear understanding of the software requirements and objectives.

2.3 Scope

The scope of the software component includes the programming logic, user interface design, scheduling algorithms, and communication protocols necessary to operate the automatic pet feeder. It does not encompass hardware design specifications or physical components.

2.4 Functional Requirements

Functional requirements for an IoT project detail the specific tasks and capabilities the system must fulfil. This includes functionalities related to data collection, processing, communication, and control. These requirements specify the actions the IoT device or system should perform to meet the needs of users or other systems interacting with it.

1. Feeding Mechanism

- ❖ Specify the need for a feeding mechanism capable of dispensing pet food accurately and reliably.
- ❖ Define the requirements for portion control to prevent overfeeding or underfeeding.

2. Portion Control

- ❖ The feeder must dispense a specified amount of food at each feeding time.
- ❖ Users should be able to adjust the portion size according to their pet's dietary needs.

3. Food Storage and Dispensing

- The feeder should include a storage container for dry pet food.
- A mechanism (such as a motorized auger or gravity feed with a timed release) must be in place to dispense food accurately and reliably.

4. User Interface

- The feeder should have an easy-to-use interface for programming the feeding schedule and portion sizes. This could be a simple LCD screen with buttons or a more advanced solution like a mobile app.
- Indicators (LEDs or display messages) should notify the user of important statuses, such as low food levels, feeding schedule confirmation, and errors.

5. Timekeeping

- The feeder must keep accurate time to ensure feedings occur at the scheduled times. This could involve using a real-time clock (RTC) module.

6. Scheduling Functionality

- Outline the requirement for scheduling feeding times based on the dietary needs and routines of pets.
- Specify the ability to set up multiple feeding schedules for different pets, if applicable.

7. Safety Features

- Identify the need for safety mechanisms to prevent food contamination, jamming of the feeding mechanism, or accidental pet access.
- Specify requirements for fail-safe mechanisms to handle power failures or system malfunctions.

2.5 Non-Functional Requirements

Non-functional requirements for an IoT project specify the qualities or characteristics the system must possess. These include attributes like reliability, security, performance, scalability, usability, and maintainability. Non-functional requirements define the overall behaviour and performance of the IoT system beyond its specific functionalities, ensuring it meets the desired standards and expectations.

1. Reliability

- Specify the requirement for the automatic pet feeder to operate reliably under various environmental conditions.

2. Accuracy

- Specify the requirement for accurate portion control and timing of feeding schedules to ensure the well-being of pets.
- Define acceptable tolerances for portion sizes and timing accuracy.

3. Ease of Maintenance

- Define requirements for ease of maintenance and servicing of the automatic pet feeder, including accessibility to components and ease of cleaning.

4. Scalability

- Specify the ability of the feeder to accommodate additional pets or features in the future.
- Define requirements for scalability in terms of hardware and software architecture.

5. Environmental Impact

- The feeder is designed with sustainability in mind, using recyclable materials where possible and minimizing energy consumption.

6. Compatibility

- The feeder is compatible with various types of dry pet food, and possibly wet food, depending on the design.

7. Cost-Effectiveness

- The system should be cost-effective, providing good value for the functionality offered.
- The system should be designed with cost-effective components without compromising quality.

HARDWARE AND SOFTWARE REQUIREMENTS

CHAPTER – 3

HARDWARE AND SOFTWARE REQUIREMENTS

3.1 Hardware Requirements

1. **Arduino Uno**
2. **Jumper Wires**
3. **Breadboard**
4. **USB Cable**
5. **LCD 16*2 Display**
6. **Servo Motor**
7. **4x4 Matrix Keypad**
8. **DS3231 RTC Module**
9. **Reset Button**

3.1.1 Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), and a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your Uno without worrying too much about doing Serial No. Name of the components something wrong, worst-case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE). The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

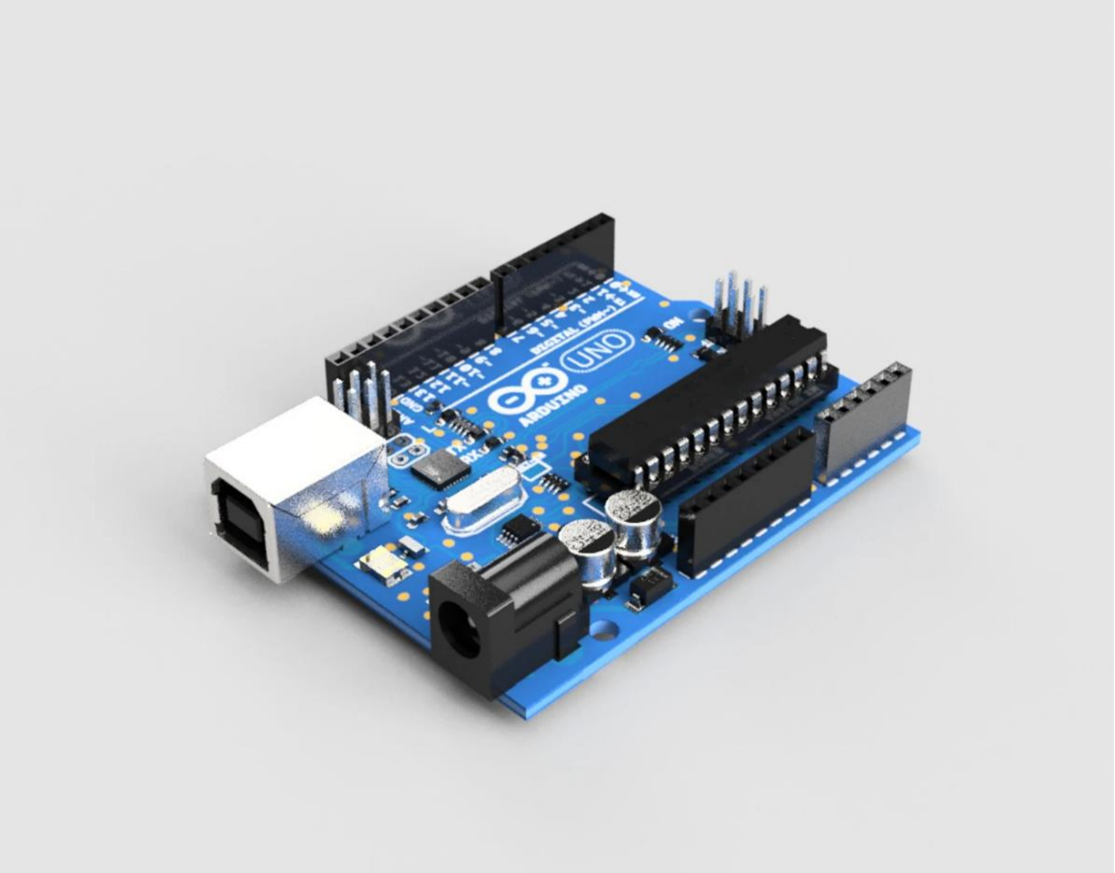


Figure: Arduino Uno

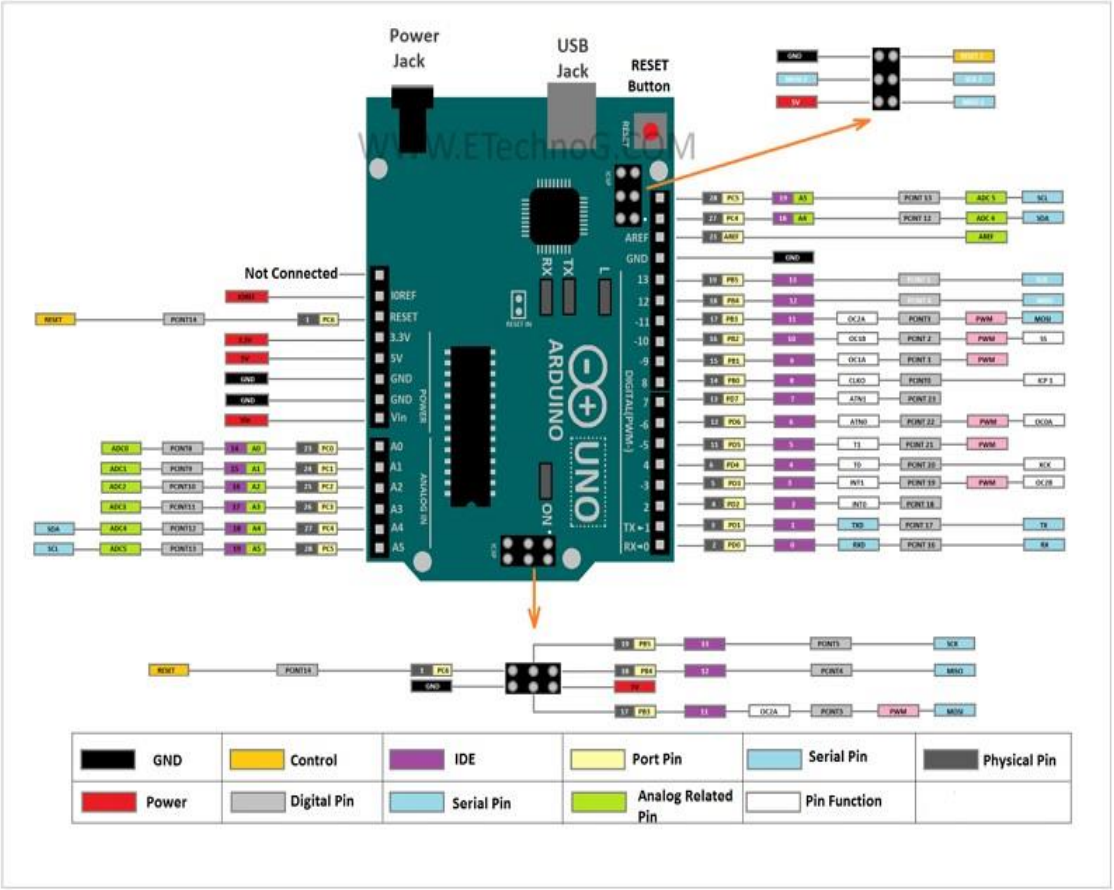


Figure: Arduino Uno pin diagram

PWM pins

These are digital pins marked with a (pins 11, 10, 9, 6, 5 and 3). PWM stands for "pulse width modulation" and allows to make digital pins output "fake" varying amounts of voltage. You'll learn more about PWM later.

TX and RX pins

Digital pins 0 and 1. The T stands for "transmit" and the R for "receive" Arduino uses these pins to communicate with the computer. Avoid using these pins unless you're running out of pins.

LED attached to digital pin 13

This is useful for an easy debugging of the Arduino sketches.

TX and RX pins

These pins blink when there is information being sent between the computer and the Arduino.

Analog pins

The analog pins are labelled from A0 to A5 and are most often used to read analog sensors. They can read different amounts of voltage between 0 and SV. Additionally, they can also be used as digital output/input pins like the digital pins.

Power pins

The Arduino has 3.3V or SV supply, which is useful since most components require 3.3V or 5V The pins labelled as "GND" are the ground pins.

Reset button, when you press that button, the program that is currently being run in your Arduino will start from the beginning. You also have a Reset pin next to the power pins that acts as reset button. When you apply a small voltage to that pin, it will reset the Arduino.

Power ON LED

Will be on since power is applied to the Arduino.

USB jack

Connecting a male USB A to male USB B cable is how you upload programs from your computer to your Arduino board. This also powers your Arduino.

Power jack

The power jack is where you connect a component to power up your Arduino (recommended voltage is 5V). There are several ways to power up your Arduino: rechargeable batteries, disposable batteries, wall-warts, and solar panel.

3.1.2 Jumper Wires

Jumper wires for solder less breadboarding can be obtained in ready to-use jump wire. Ready-to-use jump wires come in different qualities, some even with tiny plugs attached to the wire ends. Jump wire material for ready-made or homemade wires should usually be 22 AWG (0.33 mm) solid copper, tin-plated wire - assuming no tiny plugs are to be attached to the wire ends. The wire ends should be stripped 3⁄16 to 5⁄16 in (4.8 to 7.9 mm).

Differently coloured wires and color-coding discipline are often adhered to for consistency. However, the number of available colours is typically far fewer than the number of signal types or paths. Typically, a few wire colours are reserved for the supply voltages and ground (e.g., red, blue, black), some are reserved for main signals, and the rest are simply used were convenient. Some ready-to-use jump wire sets use the colour to indicate the length of the wires, but these sets do not allow a meaningful color-coding scheme. The insulation on jumper wires safeguards against short circuits, preventing unintended electrical connections.

The variety of colours aids in organizing and identifying connections within complex circuits. Standardized lengths contribute to a neat and organized appearance on a breadboard. Jumper wires play a crucial role in educational settings, where they facilitate hands-on learning of electronics and circuitry. Their flexibility, insulation, and variety contribute to the efficiency and effectiveness of circuit building, making them an integral part of the electronics enthusiast's toolkit.

3.1.2.1 Male to Female Jumper Wires

Male to female jumper wires are essential components in electronics and prototyping, widely used to establish connections between various elements on a breadboard or to connect different electronic modules. These wires have a male pin on one end and a female socket on the other, facilitating versatile connectivity. The male pin, which can be inserted into breadboards, female headers, or other connectors, allows for a stable and reliable connection. Conversely, the female socket can receive male pins from other components, providing flexibility in creating circuits without the need for soldering. This makes male to female jumper wires particularly valuable for experimenting, testing, and developing prototypes, as they can be easily adjusted and reused, enhancing both efficiency and convenience in the design process.

Figure: Male to Female Jumper wires

3.1.2.2 Male to Male Jumper Wires

Male to male jumper wires are fundamental components in electronics and prototyping, serving as connectors to establish direct connections between different elements. These wires feature a male pin on both ends, allowing for easy insertion into breadboards, female headers, or other compatible connectors. This configuration enables the direct linkage of various electronic components, such as sensors, LEDs, or microcontrollers, without the need for additional adapters or connectors. Male to male jumper wires facilitate the creation of circuitry on breadboards, enabling rapid prototyping, testing, and experimentation. They provide a straightforward and flexible solution for creating temporary connections, allowing for quick adjustments and modifications during the design process. As reusable and versatile tools, male to male jumper wires are indispensable for electronics enthusiasts, hobbyists, and professionals alike.

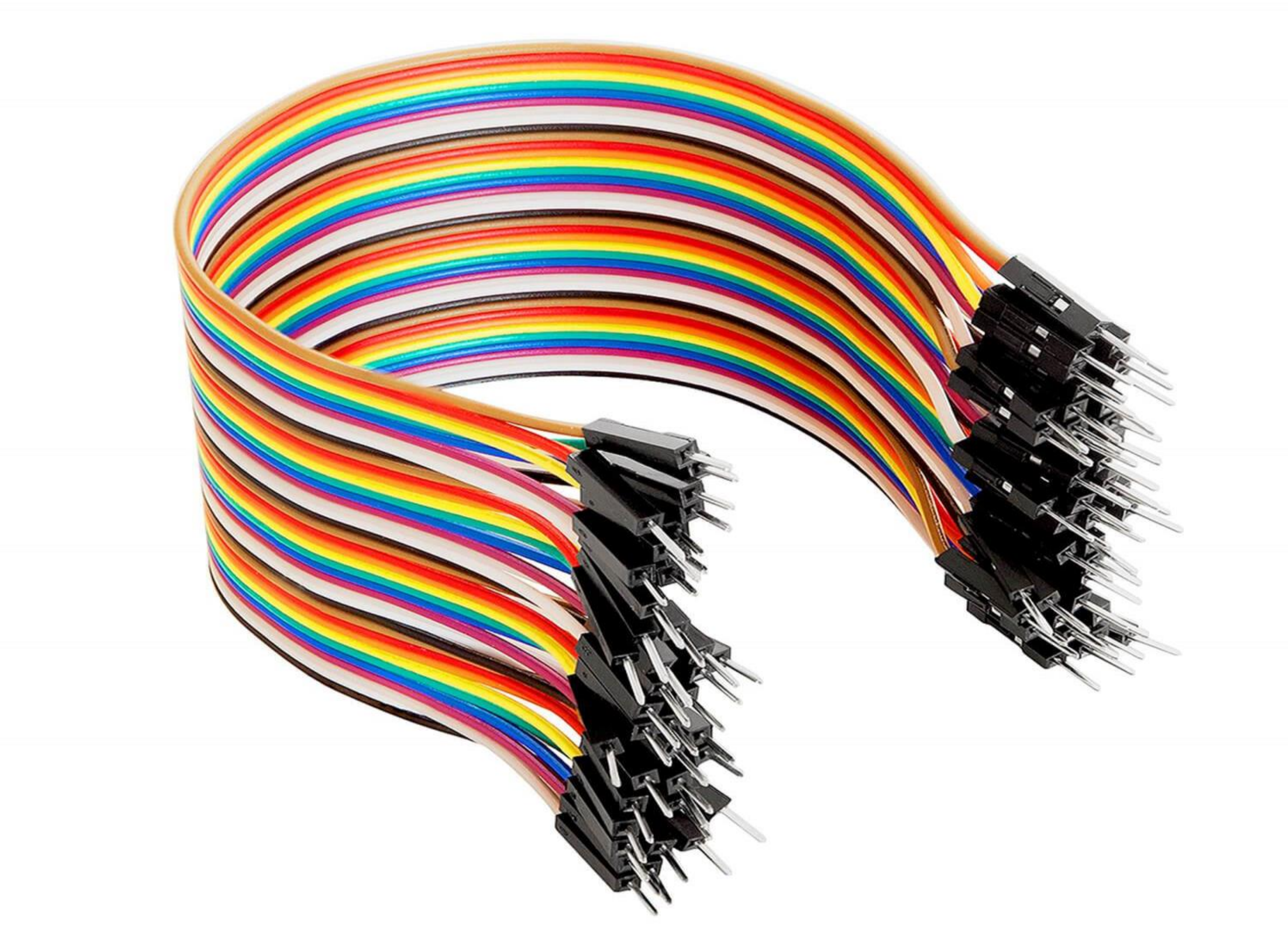


Figure: Male to Male Jumper wires

3.1.3 Breadboard

A breadboard consists of plastic block holding a matrix of electrical sockets of a size suitable for gripping thin connecting wire, component wires or the pins of transistors and integrated circuits (ICs). The sockets are connected inside the board, usually in rows of five sockets. An electronics breadboard (as opposed to the type on which sandwiches are made) is referring to a solderless breadboard. These are great units for making temporary circuits and prototyping, and they require absolutely no soldering. Breadboards are temporary work boards for electronic circuits. The general shape of a breadboard is Compatible with most breadboards, 24-gauge wire is used to connect circuits; solid wire, not stranded. Sometimes, kits may be available with various colours of fixed lengths to specifically fit breadboards. These are a nice convenience. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure. The grid is made up of perfectly aligned spring clip holes that are 0.1″ apart in both the X and Y dimensions. Required for prototyping and organizing connections. This is of most benefit if you are designing the circuit from scratch and must change components often. his is of most benefit if you are designing the circuit from scratch and have to change components often.

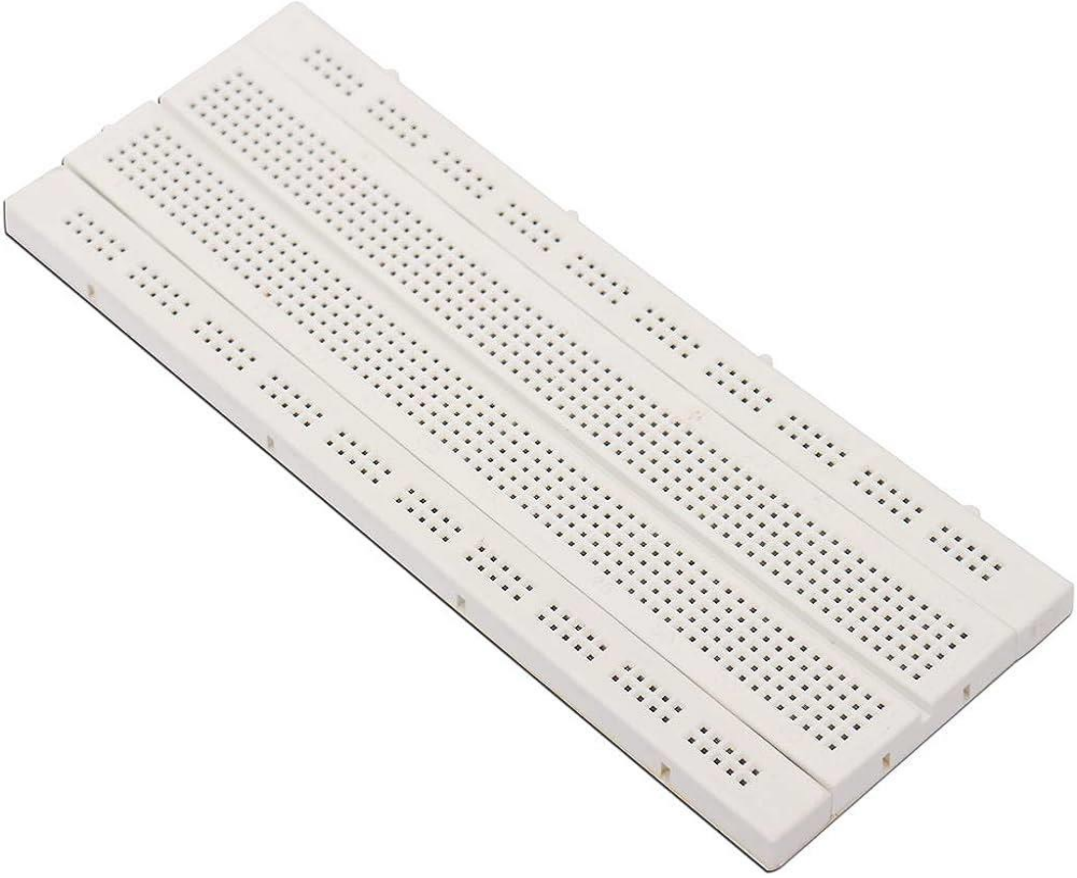


Figure: Breadboard

3.1.4 USB Cable

Needed for connecting Arduino to a computer for programming and power.

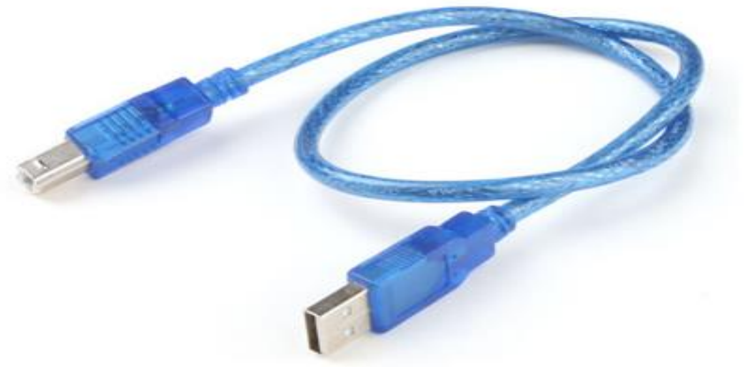


Figure: USB Cable

3.1.5 LCD 16*2 Display

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

A 16x2 LCD display is a widely used electronic component that features a screen with 16 columns and 2 rows, allowing for the display of up to 32 characters at a time. It can show alphanumeric characters, including letters, numbers, and custom symbols, making it versatile for various applications. These displays often come with a backlight for improved visibility in different lighting conditions. Typically, a 16x2 LCD has 16 pins for connections, including power, ground, contrast adjustment, and data/command inputs. The interface can be parallel, requiring multiple data lines for communication, or it can utilize I2C, simplifying wiring by reducing the number of necessary connections. These displays are commonly used with microcontrollers like Arduino, making them ideal for DIY electronics projects, data logging, and user interface displays.

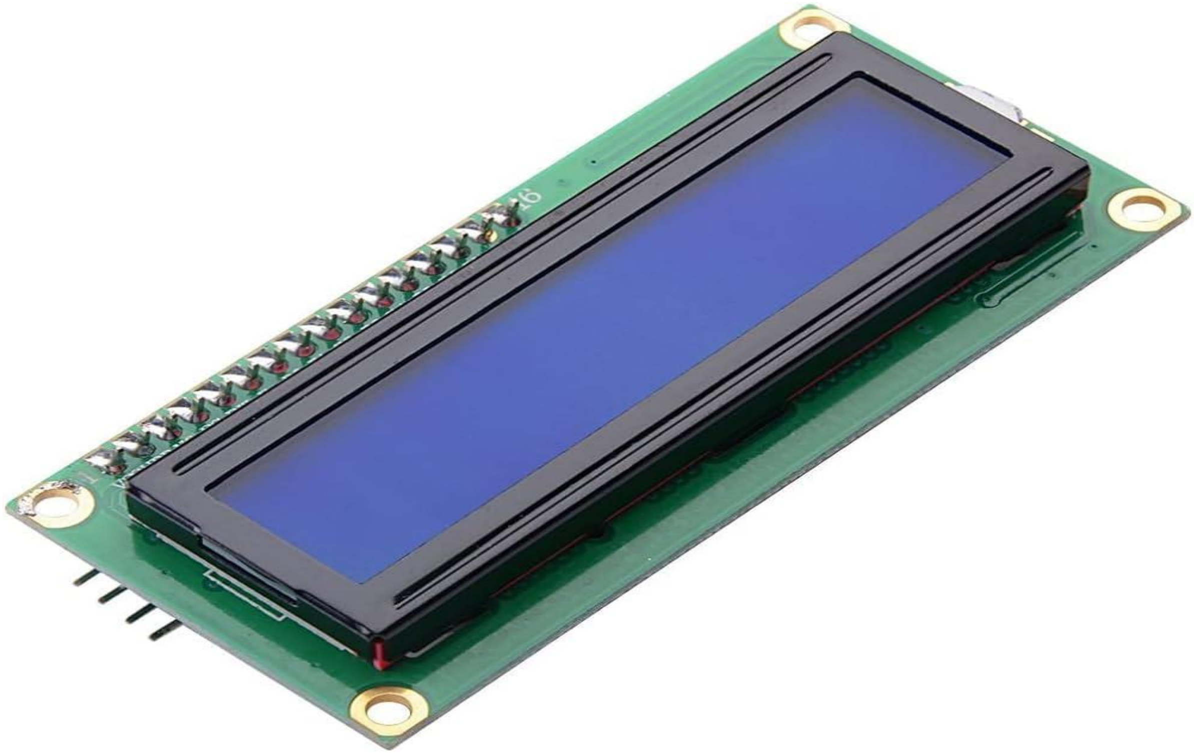


Figure: LCD 16*2 Display

3.1.6 Servo Motor

A servo motor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servo motors. It receives a control signal that represents a desired output position. The motor then adjusts its position, speed, and acceleration to match the desired output as closely as possible, based on feedback from the position sensor. The control circuit compares the actual position of the motor with the desired position and adjusts to reduce any difference between the two (error).

Using Servo Motors in an Automatic Pet Feeder Project: In an automatic pet feeder, servo motors can be used to control the dispensing mechanism with precision. For example, a servo motor could control a rotating flap or auger that dispenses a specific portion of food. The precise control offered by a servo motor ensures that the correct amount of food is dispensed each time, contributing to portion control and the overall reliability of the feeder.

Figure: Servo Motor

3.1.7 4x4 Matrix Keypad

A 4x4 matrix keypad is a common input device used in electronics projects, particularly for applications that require user interaction through multiple button presses, such as entering numbers or characters. It consists of 16 buttons arranged in a grid of four rows and four columns, hence the name 4x4 matrix keypad. The keypad is wired so that each button connects a specific row to a specific column. When a button is pressed, it completes the circuit between the corresponding row and column.

Wiring:

- Rows: Four pins correspond to the four rows.
- Columns: Four pins correspond to the four columns.

A 4x4 matrix keypad is a versatile and cost-effective input device for many electronics projects. It enables users to input data efficiently and can be easily integrated with microcontrollers, making it ideal for projects like security systems, calculators, and more. Understanding its wiring and scanning mechanism is crucial for effective implementation.

Figure:4*4 matrix keypad

3.1.8 DS3231 RTC Module

DS3231 is a RTC (Real Time Clock) module. It is used to maintain the date and time for most of the Electronics projects. This module has its own coin cell power supply using which it maintains the date and time even when the main power is removed or the MCU has gone through a hard reset. So once we set the date and time in this module it will keep track of it always. In our circuit, we are using DS3231 to feed the pet according to the time, set up by the Pet’s owner, like an alarm. As, clock reaches to the set time, it operates the servo motor to open the container gate and the food drops in the Pet’s food bowl.

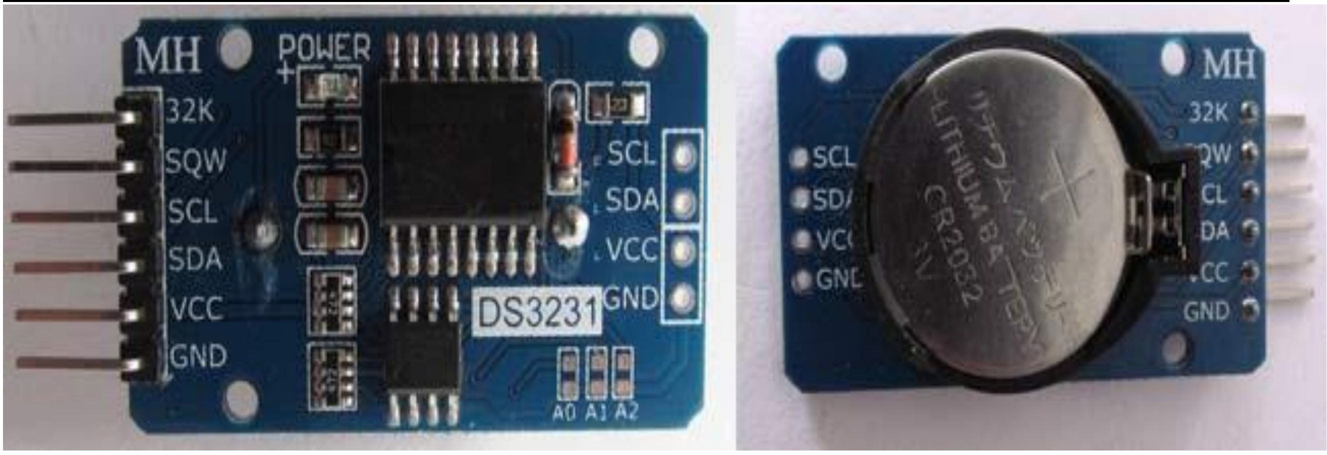


Figure:DS3231 RTC Module

3.1.9 Reset Button

Pushing a button involves applying pressure to a switch or control mechanism, typically with your finger or hand, to activate a function or signal. Buttons are commonly used in various electronic devices, appliances, machinery, and interfaces to initiate actions or commands. The operation of a button usually follows a simple principle: when pressure is applied, it completes an electrical circuit, triggering the intended response. This response can vary depending on the device or system it's a part of, such as turning on a light, starting a machine, sending a signal, or executing a command in a software application. Buttons come in different shapes, sizes, and designs, including physical buttons that you physically press, touch-sensitive buttons that respond to touch, and virtual buttons displayed on screens that you tap or click. They play a crucial role in human-machine interaction, offering a tangible and intuitive way for users to control and interact with technology.

Figure: Reset Button

3.2 Software Requirements

1. **Arduino IDE**
2. **Language: Embedded C**

3.2.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a software platform designed to simplify the process of programming Arduino microcontroller boards. Developed by the Arduino team, the IDE provides a user-friendly interface that caters to both beginners and experienced programmers, enabling them to write, compile, and upload code to Arduino boards with ease.

The key features of the Arduino IDE include:

- A text editor for writing code in the Arduino programming language, which is based on C/C++ and relatively easy to learn. The IDE also offers syntax highlighting and auto-completion to improve coding efficiency.
- A compiler that checks the code for syntax errors and converts it into machine-readable instructions that can be understood by the Arduino board's microcontroller. If any errors are detected, the IDE provides helpful error messages to assist users.
- Tools for uploading the compiled code to the Arduino board via a USB connection. The IDE automatically detects the connected board and allows users to select the appropriate board type and serial port before uploading.

In addition to these core features, the Arduino IDE offers a range of libraries that provide pre-written code for commonly used functions and components, saving users time and effort when building their projects. The IDE also supports the integration of third-party libraries, further expanding its capabilities.

Overall, the Arduino IDE is a powerful yet accessible tool that simplifies the development of projects with Arduino boards. Its intuitive interface, comprehensive features, and extensive community support make it an indispensable resource for hobbyists, educators, and professionals alike.

3.2.2 Language: Embedded C

Embedded C is a specialized programming language used extensively in the development of embedded systems, which are integral to the Internet of Things (IoT). Unlike standard C programming, Embedded C is tailored for programming microcontrollers and other hardware devices that form the backbone of IoT applications. These systems require efficient, reliable, and low-level interaction with hardware, making Embedded C an ideal choice due to its close proximity to hardware functions and its ability to directly manipulate bits and bytes.

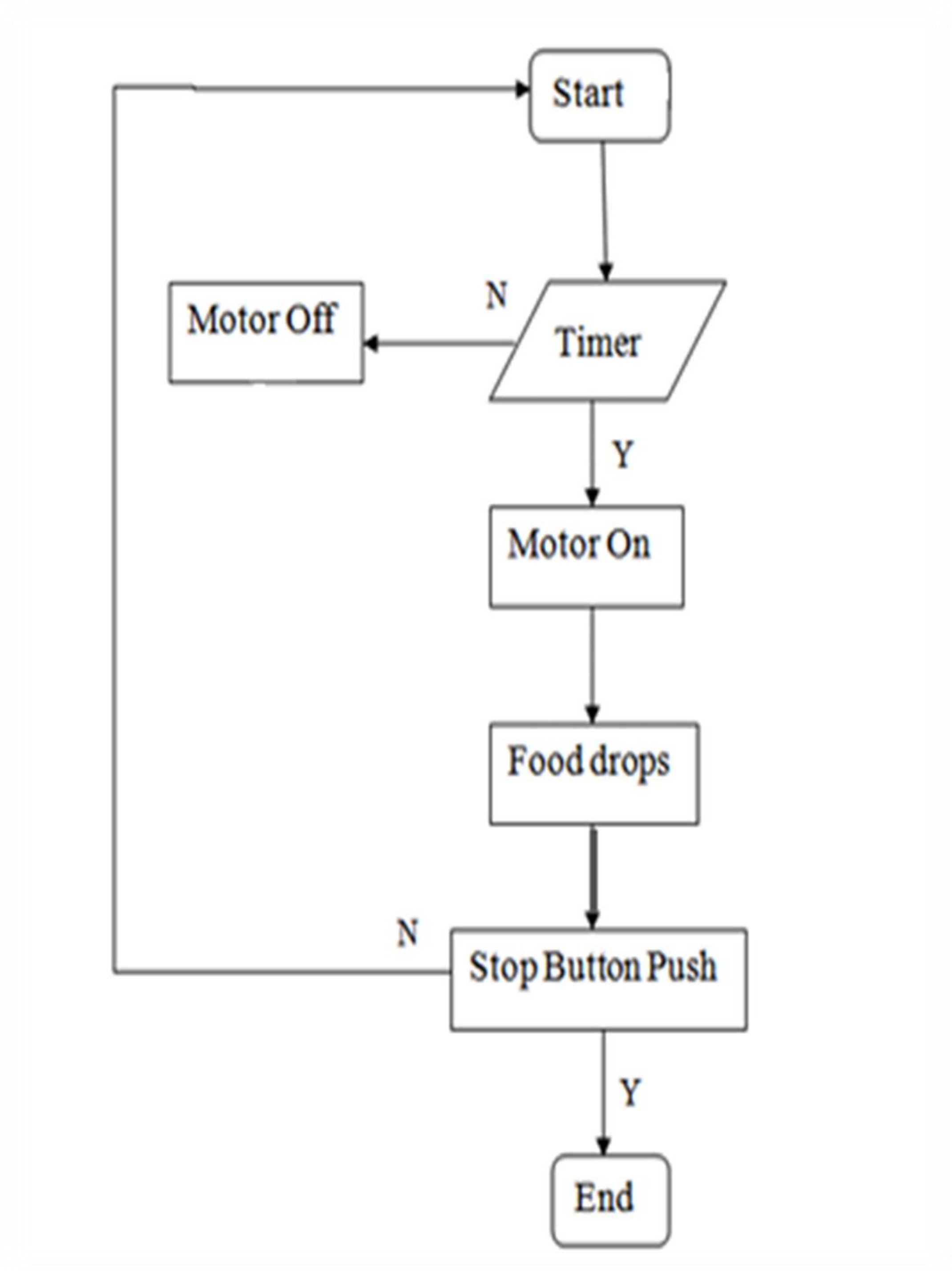
In the realm of IoT, devices range from simple sensors to complex systems that communicate over networks to perform automated tasks. Embedded C facilitates the creation of firmware that can handle these tasks by providing a robust set of features for real-time operations, memory management, and direct hardware access. The language supports the implementation of interrupt service routines, which are crucial for real-time responsiveness in IoT devices. Additionally, Embedded C allows for precise timing operations and low power consumption, which are essential for battery-powered IoT devices.

Developers use Embedded C to write code that is compact and efficient, often optimizing for the limited resources available on microcontrollers. This includes writing drivers for sensors and actuators, managing communication protocols such as I2C, SPI, and UART, and implementing network stacks for wireless communication protocols like Wi-Fi, Bluetooth, and Zigbee. The ability to write deterministic and predictable code makes Embedded C suitable for developing reliable and stable IoT systems.

In summary, Embedded C plays a critical role in the development of IoT devices by providing the necessary tools and functionalities to interface directly with hardware, manage limited resources efficiently, and ensure real-time performance. Its use in writing low-level code for microcontrollers and other embedded systems is fundamental to the creation of sophisticated IoT applications that are both efficient and reliable.

3.3 Data Flow Diagram

Data Flow Diagram (DFD) represents the flow of data within information systems. Data Flow Diagrams (DFD) provide a graphical representation of the data flow of a system that can be understood by both technical and non-technical users. The models enable software engineers, customers, and users to work together effectively during the analysis and specification of requirements.



SYSTEM ANALYSIS AND DESIGN

CHAPTER - 4

SYSTEM ANALYSIS AND DESIGN

The development of a pet feeder for two pets involves several critical phases, among which the Analysis Phase and Design Phase are fundamental in ensuring the product meets the intended requirements and user needs effectively. Here's a breakdown of what each phase typically entails in the context of creating a dual-pet feeder.

Analysis Phase

The Analysis Phase is where the project requirements are gathered, and the needs of the users are thoroughly understood. This phase sets the foundation for a successful product by defining what it needs to achieve and for whom.

Requirement Gathering

User Requirements: Understand the needs of pet owners, focusing on those with two pets. This might involve surveys, interviews, or market research to gather insights on feeding schedules, portion sizes, and specific challenges faced by owners in feeding multiple pets.

Technical Requirements

Define the technical specifications required to meet the user needs. This could include the type of feeding mechanism, portion control accuracy, connectivity options (e.g., Wi-Fi, Bluetooth), and compatibility with various types of pet food.

Feasibility Study

Assess the technological feasibility and cost implications of the proposed features, such as automated scheduling, remote monitoring, and individual pet identification.

Market Analysis

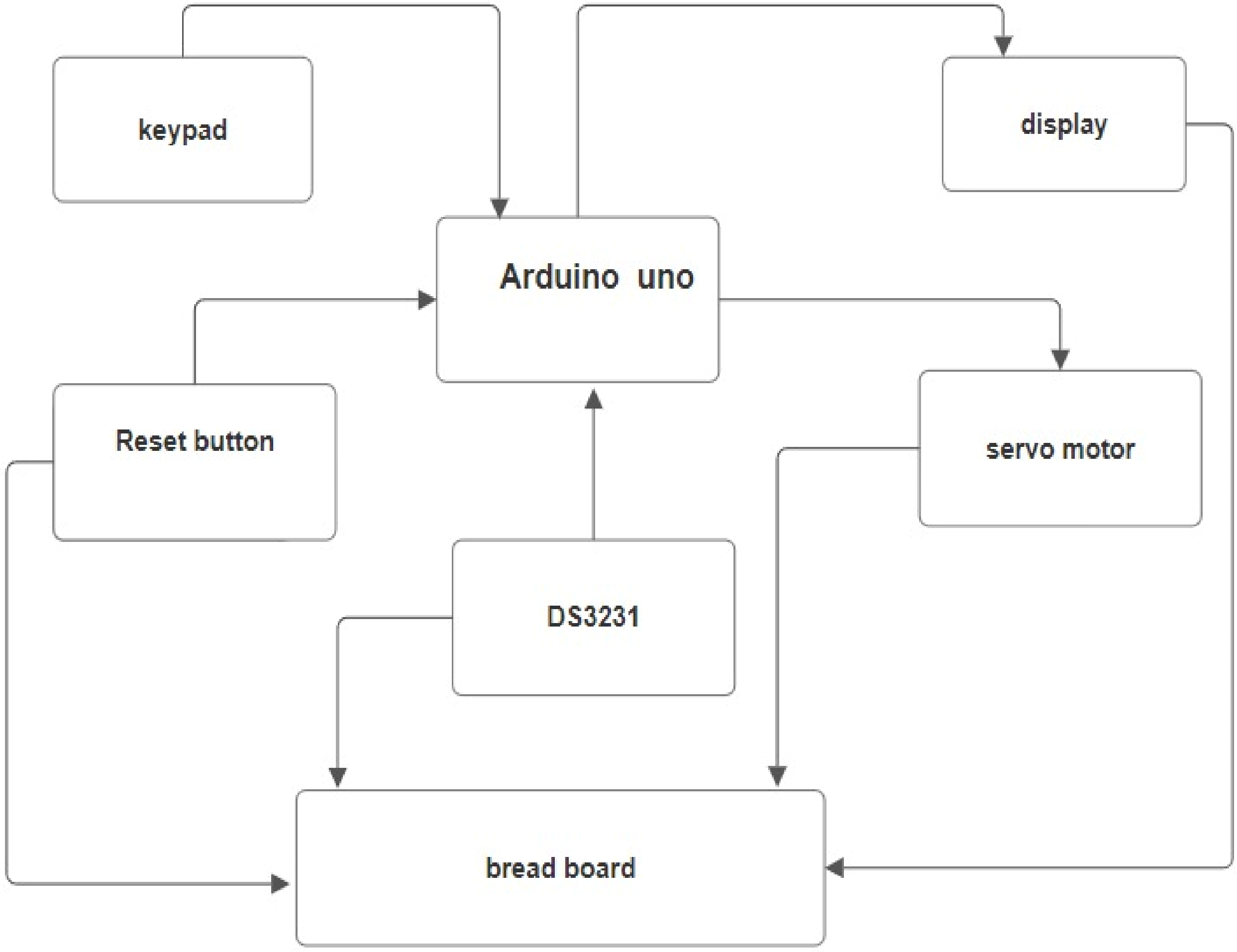
Analyze competitors and similar products in the market to identify gaps and opportunities, setting the stage for differentiation.

User Personas and Scenarios

Develop user personas and scenarios to better understand the context in which the pet feeder will be used. This helps in identifying specific design challenges, such as ensuring the device is tamper-proof from clever pets or user-friendly for less tech-savvy pet owners.

4.1 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. The blocks that give the block diagram its name represent the different elements within a system. The lines and arrows show the relationships between those blocks. These visual elements provide a high-level, functional overview of the system that is easy to digest and understand.



4.2 Design Phase

Following the Analysis Phase, the Design Phase translates the gathered requirements and insights into tangible design solutions. This phase involves conceptualization, prototyping, and user testing to refine the product’s functionality and usability.

Conceptualization

Develop initial design concepts that address the key requirements identified in the analysis phase. This might include sketching, creating CAD models, and deciding on the materials and technology to be used.

Prototyping

Create functional prototypes to explore and test different design solutions. Rapid prototyping methods, like 3D printing, can be particularly useful here for iterating quickly based on feedback.

User Interface and Experience (UI/UX)

Design the user interface for any digital controls or app integration, focusing on simplicity and ease of use. The user experience design should consider how pet owners interact with the device, including setup, daily use, and maintenance.

Engineering Design

Detailed engineering designs are finalized, including mechanical components, electrical circuits, and embedded software. This stage must address practical aspects such as food storage, dispensing mechanisms, power supply, and connectivity.

User Testing

Conduct user testing with prototypes to gather feedback on usability, functionality, and any unforeseen issues. This feedback is crucial for refining the design before moving into production.

Design for Manufacturability

Make necessary adjustments to ensure the product can be manufactured efficiently and at a reasonable cost, without compromising on quality or functionality.

System design can define as a systematic process of formulating a comprehensive plan for the implementation and design of a system. It entails defining the system's structure, components, interfaces, and functionalities to fulfil specific requirements and achieve desired objectives. The scope of system design includes different elements like architecture design, module design, DB design, UI design and the seamless integration of different components. By carefully considering these aspects, system designers lay the groundwork for building a robust and efficient system that aligns with the project's goals.

In the system design phase, the requirements are gathered during analysis phase is transformed into a concrete technical solution. This entails making crucial design choices concerning technologies, frameworks, databases, security measures, performance optimizations, and scalability requirements. The design should aim to establish an efficient, reliable, maintainable, and user-friendly system that abides by the intended objectives. By considering these factors, system designers confirm that the resulting system is well-equipped to handle the desired functionalities while adhering to high standards of performance and usability.

During the system design process, it is frequently encountered to create visual representations, such as architectural diagrams, data flow diagrams, entity-relationship diagrams, and sequence diagrams. These diagrams show the illustrating the structure, components, and interactions within the system.

4.3 System Architecture

System architecture is a basic concept in the world of computing and technology. It refers to the HLD and structure of complex systems, whether it’s a software application, computer networks, hardware systems, or any combination of above elements. The main goal of system architecture is to produce a clear and organized blueprint that outlines how the system's components interact and work together to achieve specific objectives. system architecture forms the backbone of complex technological systems, providing an all-encompassing and genuine essay detailed plan for design and operation. By directing key elements like components, data flow, communication, and security, system architecture lays the groundwork for robust, scalable, and high-performing systems. For any software application, computer network, hardware system, a well-designed architecture confirms that the system proficiently completes its purpose while being adaptable to future needs. Design patterns are the most powerful tool for software developer. It is required to understand design patterns rather than memorizing its classes, methods, and properties. It is also to learn how to apply pattern to specific problem to get desired result. This will be required for continuous practice of using and applying design patterns in day-to-day software development.

4.3.1 Circuit Diagram

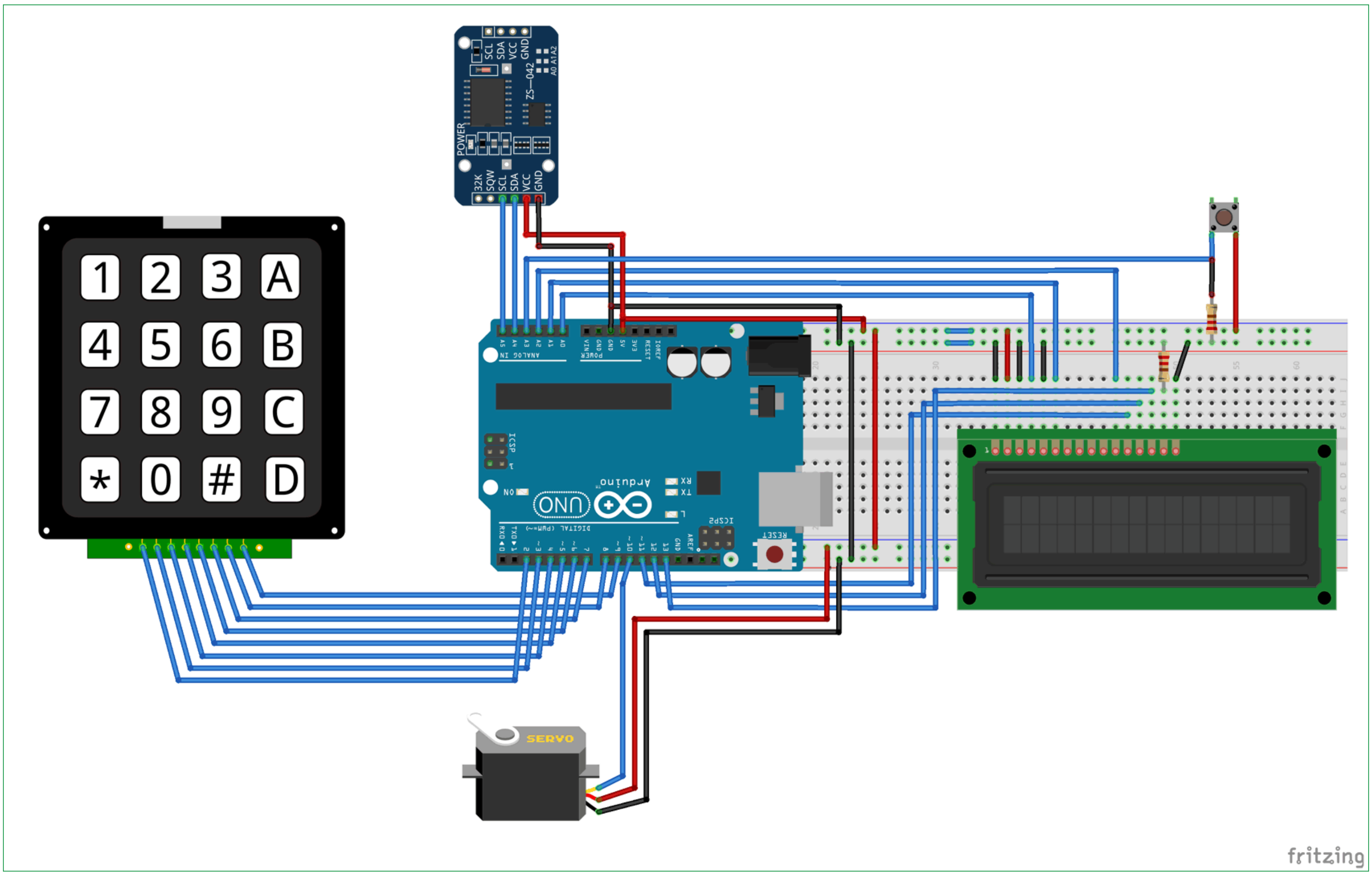


Figure: Circuit diagram

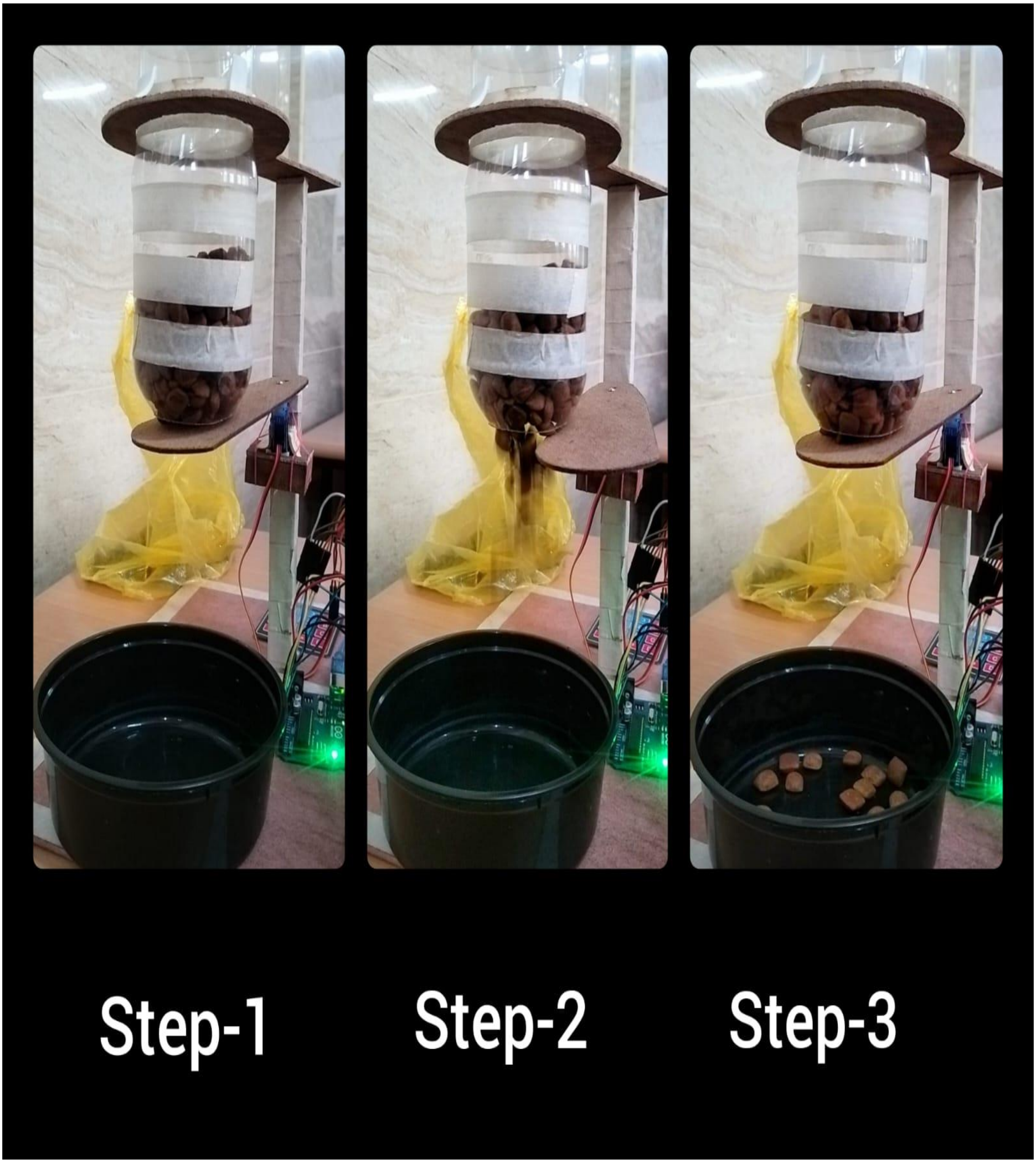
In this circuit, we are using a 16*2 LCD to display the time using DS3231 RTC Module with Arduino UNO. Also, a servo motor is used to rotate the containers to provide the food and 4*4 matrix keypad to manually set up the time for feeding the Pet. You can set the rotation angle and container opening duration according to the quantity of food you want to serve to your pet. The quantity of food may also depend upon your pet whether it's a dog, cat or bird., for Getting Time and Date, we have used RTC (Real Time Clock) Module. We have used the 4*4 Matrix Keypad to set the Pet's eating time manually with the help of 16x2 LCD. The Servo motor rotates the container and drop the food on the time set by the user. The LCD is used for displaying the Date and Time. Arduino based Automatic Pet Feeder which can automatically serve food to your pet timely. It has a DS3231 RTC (Real Time Clock) Module, which used to set time and date on which your pet should be given food. So, by setting up the time according to your pet's eating schedule, the device drops or fill the food bowl automatically.

4.4 Coding

```
#include <DS3231.h>
#include <Servo.h>
#include <LiquidCrystal.h>
#include <Keypad.h>
const byte ROWS = 4; // Four rows
const byte COLS = 4; // Three columns
// Define the Keymap
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
byte rowPins[ROWS] = { 2, 3, 4, 5 };
// Connect keypad COL0, COL1 and COL2 to these Arduino pins.
byte colPins[COLS] = { 6, 7, 8, 9 };
// Create the Keypad
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
DS3231 rtc(A4, A5);
Servo servo_test; //initialize a servo object for the connected servo
LiquidCrystal lcd(A0, A1, A2, 11, 12, 13); // Creates an LC object. Parameters: (rs, enable, d4, d5, d6, d7)
int t1, t2, t3, t4, t5, t6;
```

```
boolean feed = true; // condition for alarm

char key;

int r[6];

void setup() {

  servo_test.attach(10); // attach the signal pin of servo to pin9 of arduino

  rtc.begin();

  lcd.begin(16,2);

  servo_test.write(10);

  Serial.begin(9600);

  pinMode(A0, OUTPUT);

  pinMode(A1, OUTPUT);

  pinMode(A2, OUTPUT);

}

void loop() {

  lcd.setCursor(0,0);

  int buttonPress;

  buttonPress = digitalRead(A3);

  if (buttonPress == 1) {

    setFeedingTime();

  }

  lcd.print("Time: ");

  String t = "";

  t = rtc.getTimeStr();

  t1 = t.charAt(0) - 48;

  t2 = t.charAt(1) - 48;
  t3 = t.charAt(3) - 48;
  t4 = t.charAt(4) - 48;
  t5 = t.charAt(6) - 48;
  t6 = t.charAt(7) - 48;
  lcd.print(rtc.getTimeStr());
  lcd.setCursor(0,1);
  lcd.print("Date: ");
  lcd.print(rtc.getDateStr());
  if (t1 == r[0] && t2 == r[1] && t3 == r[2] && t4 == r[3] && t5 < 1 && t6 < 3 && feed == true) {
    servo_test.write(40); //command to rotate the servo to the specified angle
    delay(2500);
    servo_test.write(10);
    feed = false;
  }
}
void setFeedingTime() {
  feed = true;
  int i = 0;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Set feeding Time");
  lcd.setCursor(0,1);
  lcd.print("HH:MM");
  lcd.setCursor(0,1);
  // Reset feeding time values
  for (int k = 0; k < 6; k++) {
    r[k] = 0;
  }
  while(1) {
    key = kpd.getKey();
    if (key != NO_KEY && key != 'D') {
      lcd.print(key);
      r[i] = key - 48;
      i++;
      if (i == 2) {
        lcd.print(":");
      }
      if (i == 4) {
        break; // Break the loop after setting HH:MM
      }
    }
    delay(0);
  }
}
```

4.4.1 Working Process:



TESTING

CHAPTER - 5

TESTING

Testing a pet feeder is crucial to ensure its reliability, accuracy, and safety. Here's a comprehensive testing plan covering various aspects of the pet feeder:

Functional Testing

- **Feeding Accuracy:** Verify that the feeder dispenses the correct amount of food as per the programmed portion sizes.
- **Feeding Schedule:** Test the feeder's ability to adhere to preset feeding schedules accurately
- **Food Compatibility:** Evaluate the feeder's ability to handle various sizes, shapes, and consistencies of pet food without issues.

User Interface Testing

- **Programming Feeding Schedule**: Verify that users can easily set up and adjust feeding schedules using the interface.
- **Setting Portion Sizes**: Test the functionality for programming and customizing portion sizes according to the pet's dietary needs.
- **User Feedback**: Ensure the interface provides clear feedback to users regarding feeding schedules, portion sizes, and any errors or malfunctions.

Sensor Testing

- **Food Level Sensor**: Validate the accuracy of the sensor used to detect food levels in the storage compartment.
- **Error Detection**: Test sensors for detecting issues such as motor jams, food blockages, or low battery levels.

Power and Connectivity Testing

- **Power Source:** Test the feeder's operation under different power sources (e.g., battery, plug-in adapter) to ensure reliability.
- **Connectivity Features**: If applicable, test the feeder's connectivity via Wi-Fi or Bluetooth for remote monitoring and control.

Safety Testing

- **Pet Safety**: Ensure that the feeder design minimizes any risks to pets, such as preventing access to moving parts or electrical components.
- **User Safety**: Check for any sharp edges or potential hazards in the feeder's design that could pose risks to users during assembly or maintenance.

Environmental Testing:

- **Temperature and Humidity**: Test the feeder's performance under different environmental conditions to ensure it operates reliably.
- **Durability**: Subject the feeder to stress tests to evaluate its durability and longevity, simulating typical usage scenarios.

Field Testing:

- Deploy prototypes or beta units to real users to gather feedback on usability, reliability, and any unforeseen issues.
- Incorporate user feedback into iterative improvements of the feeder design and functionality.

Regulatory Compliance:

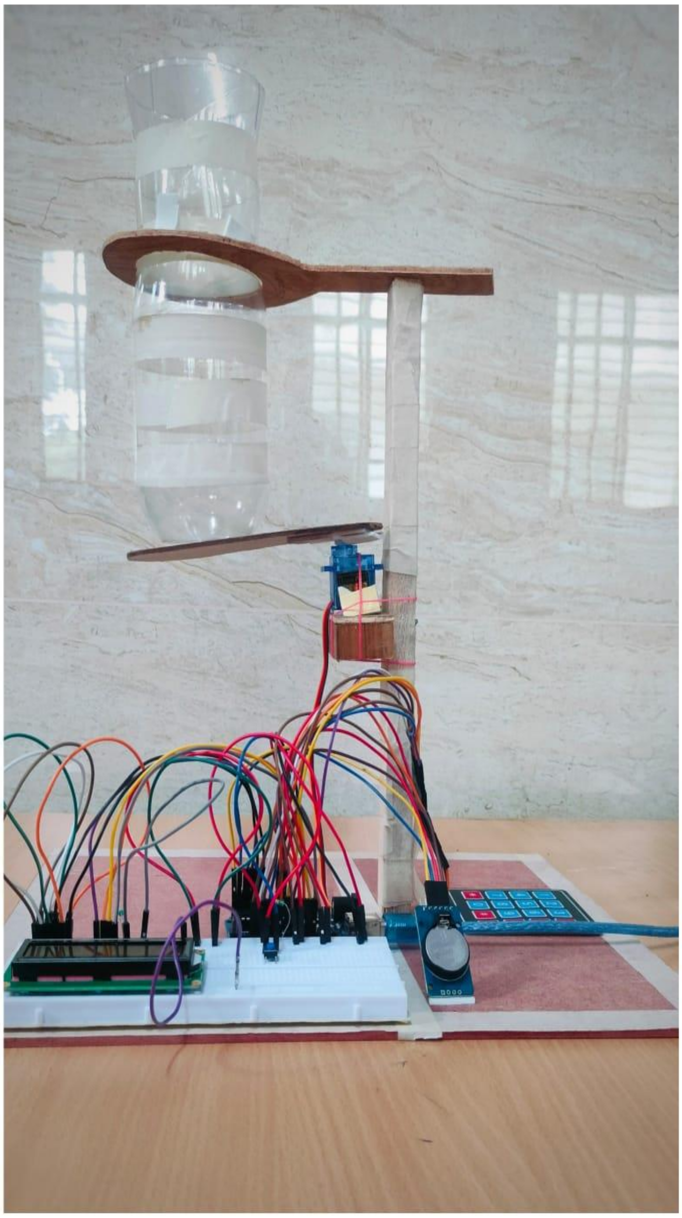
- Ensure that the pet feeder complies with relevant safety and regulatory standards for pet care products in the target market.

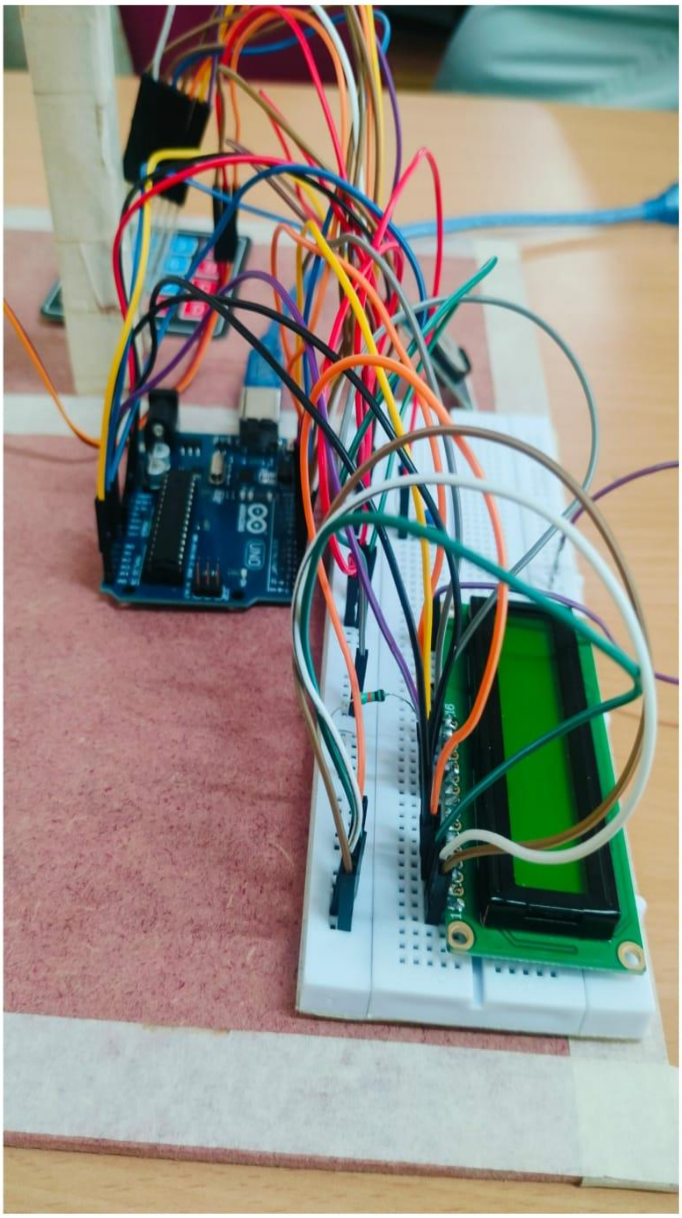
.

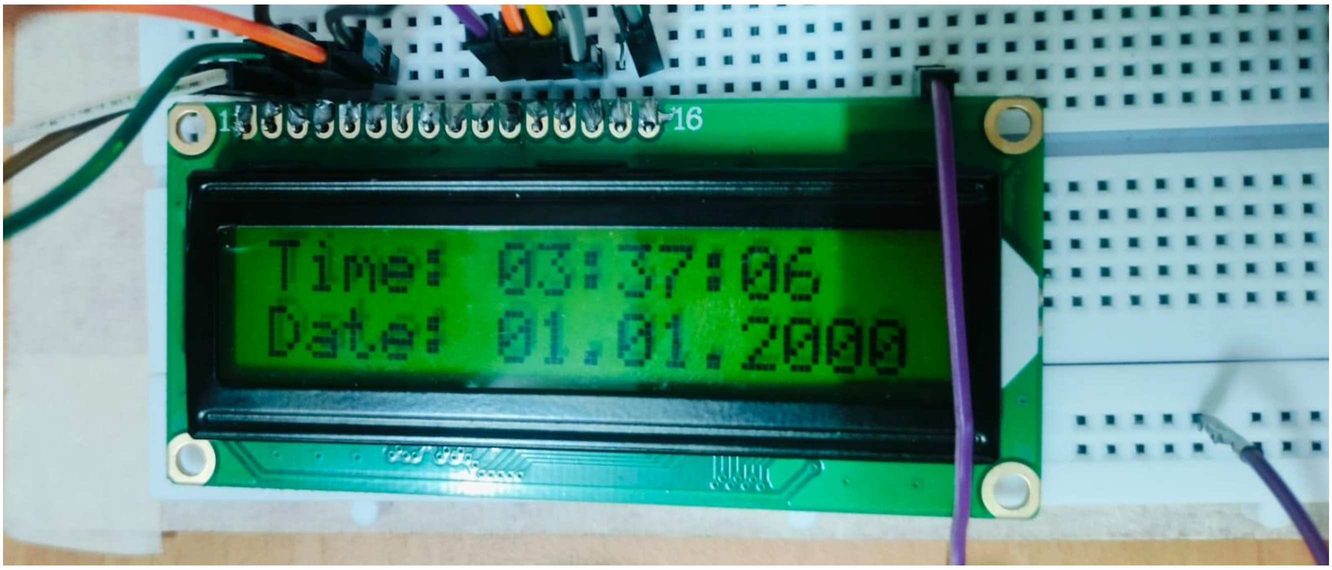
SNAPSHOTS

CHAPTER - 6

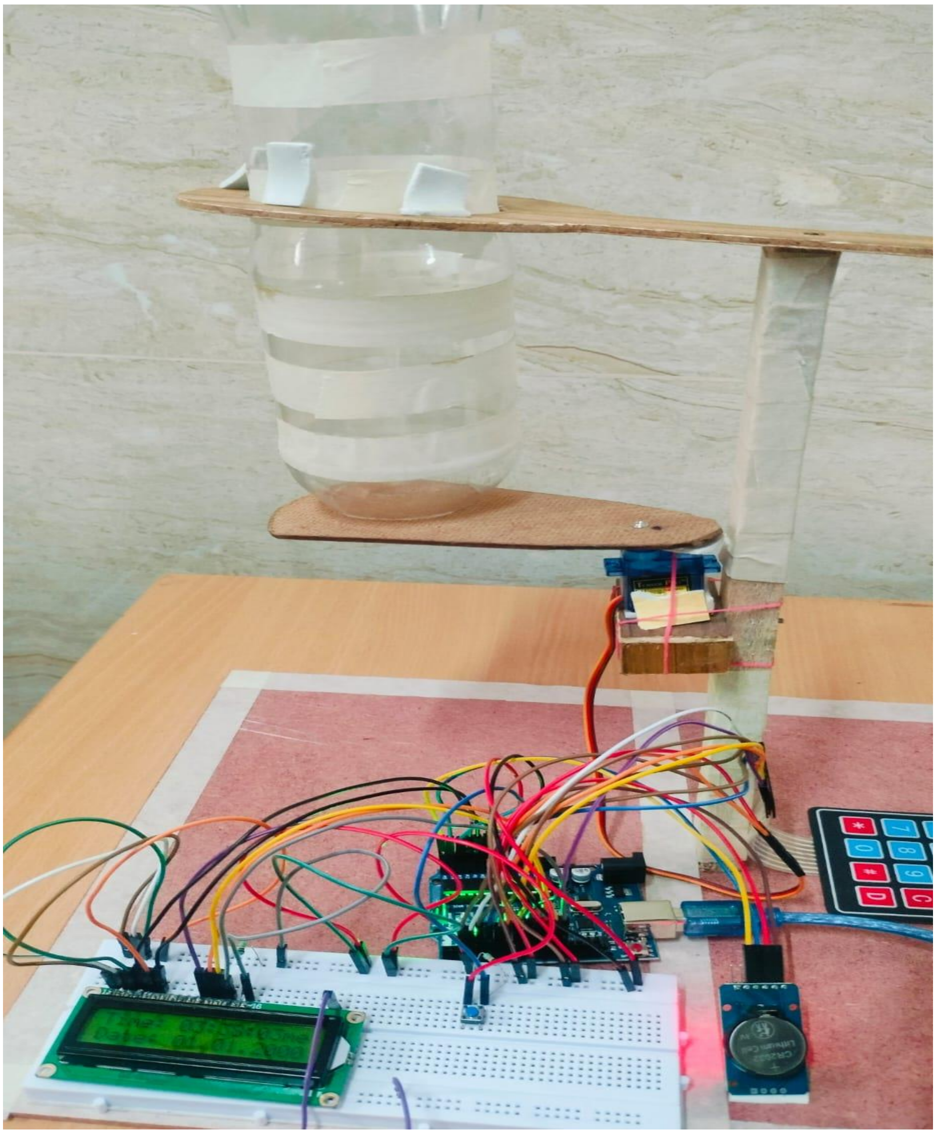
SNAPSHOTS

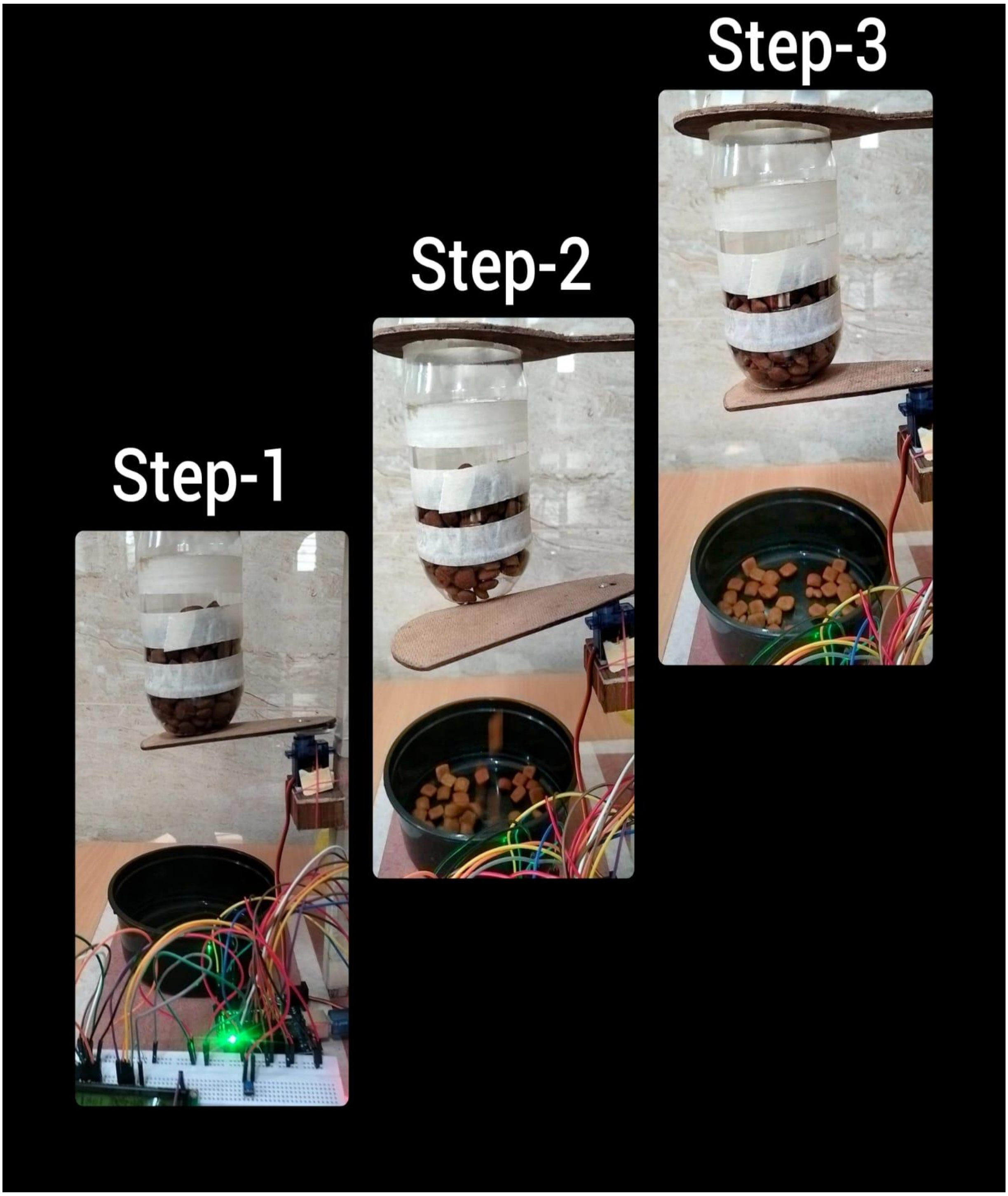






```
Time: 03:37:06
Date: 01.01.2000
```




Step-3
Step-2
Step-1

CONCLUSION

CHAPTER - 7

CONCLUSION

In conclusion, the development and testing of a pet feeder involve a comprehensive process to ensure the device meets the needs of both pets and their owners. Through meticulous design, implementation, and testing, a well-engineered pet feeder can provide convenience, reliability, and peace of mind to pet owners while ensuring the health and well-being of their furry friends.

Key aspects of the pet feeder include precise portion control, accurate feeding schedules, user-friendly interfaces, robust mechanical and electrical components, and adherence to safety standards. Thorough testing across functional, user interface, sensor, power, safety, environmental, and field-testing aspects is essential to validate the feeder's performance, reliability, and safety under various conditions. By prioritizing quality, usability, and safety throughout the development process, manufacturers can deliver a pet feeder that enhances the feeding experience for pets and simplifies the lives of their owners. Continuous iteration based on user feedback and ongoing improvement efforts can further refine the pet feeder's design and functionality, ensuring it remains a trusted and indispensable tool for pet care in households worldwide.

Automatic pet feeders have revolutionized pet care by providing significant convenience for owners and promoting better health for pets. These devices ensure consistent feeding schedules, precise portion control, and reduce pet anxiety by offering regular and predictable meals. Advanced features like health monitoring and smart technology integration enhance their utility, making it easier for owners to manage their pets' dietary needs remotely. By guaranteeing that pets are fed on time even when owners are busy or away, automatic pet feeders contribute to improved pet well-being and peace of mind for owners, marking them as a valuable addition to modern pet care practices.

In conclusion, the development of a pet feeder requires meticulous attention to detail across various stages, from initial design to testing and refinement. By integrating mechanical, electrical, and software components, along with user-friendly interfaces and safety features, a pet feeder can be created to meet the needs of both pets and their owners.

FUTURE SCOPE

CHAPTER - 8

FUTURE SCOPE

For future enhancements of a pet feeder, several avenues can be explored to improve functionality, convenience, and user experience. Here are some ideas:

1. Smart Integration

- Incorporate advanced connectivity features, such as integration with smart home systems like Amazon Alexa or Google Home, allowing users to control the feeder via voice commands.
- Develop a dedicated smartphone app with features for remote monitoring, scheduling adjustments, and receiving notifications about feeding events or issues.

2.Machine Learning Algorithms

- Implement machine learning algorithms to analyse pet feeding patterns and adjust feeding schedules or portion sizes automatically based on the pet's behaviour and dietary needs.
- Utilize image recognition technology to identify the pet using the feeder and provide personalized feeding recommendations or alerts to the owner.

3.Health Monitoring

- Integrate sensors to monitor the pet's health indicators, such as weight, activity level, or eating habits, and provide insights to the owner or veterinarian for early detection of health issues.
- Include features for tracking food consumption over time, allowing users to monitor changes in their pet's eating behavior and detect potential health concerns.

4.Modular Design

- Design the feeder with modular components, allowing users to customize and expand its functionality, such as adding additional food compartments for multi-pet households or integrating water dispensers.
- Enable compatibility with accessories or attachments for specialized diets or feeding requirements, such as slow-feed bowls or food dispensing puzzles.

5.Energy Efficiency:

- Optimize power management features to minimize energy consumption and extend battery life, particularly for battery-powered models.
- Implement solar CSS Sharing capabilities or energy harvesting technologies to supplement power sources and reduce reliance on traditional batteries or electricity.

6.Enhanced Materials and Design

- Explore eco-friendly materials and sustainable manufacturing practices to reduce environmental impact and promote responsible pet ownership.
- Incorporate design elements for ease of cleaning and maintenance, such as removable and dishwasher-safe components or anti-microbial coatings.

7.Diet Management Integration

- Future models could integrate with diet management apps or veterinary databases to provide tailored feeding recommendations based on a pet's age, breed, health condition, and activity level.

8. Interactive Features

- Adding interactive elements like cameras and two-way audio could allow owners to interact with their pets during feeding times, providing reassurance, and enhancing the pet-owner bond even when apart.

BIBLIOGRAPHY

CHAPTER - 9

BIBLIOGRAPHY

9.1 Books

✓ **Arduino Project Handbook**: 25 Practical Projects to Get You Started by Mark Geddes.

✓ **Smart Feeders:** Arduino Innovations for Happy Pets.

✓ **Fur & Functions:** Arduino Pet Feeder Design and Programming.

✓ **Tech Tails:** Arduino Pet Feeders and Beyond.

✓ **Whiskers & Wires:** Arduino Automation for Pet Owners.

✓ **Pet Projects with Arduino:** Feeding Your Furry Friends Automatically.

✓ **Paws & Pixels:** Arduino Pet Feeder Projects Illustrated.

✓ **Arduino Pet Gadgets:** DIY Feeder Edition.

✓ **Hacker Hounds:** Arduino-Powered Pet Feeding Systems Unleashed.

✓ **Feeding Fido:** Arduino-Powered Pet Care Solutions.

9.2 YouTube References

✓ **https://youtu.be/mjiexlL6Cfo?si=9tU195NdmQxNU-aT**

✓ **https://youtu.be/-99P1AsxHzI?si=5AMkR_BMsblLAUie**

✓ **https://youtu.be/EAeuxjtkumM?si=Mjd9RCGHKeLQY-RJ**

✓ **https://youtu.be/Ia7gkgSor78?si=ZWeqwRfpB9QKO6PK**

✓ **https://youtu.be/E6wkvTG2Ofs?si=xVYczAD5DQ1GW3NA**

9.3 Websites

- ✓ https://circuitdigest.com/microcontroller-projects/automatic-pet-feeder-using-arduino
- ✓ https://www.instructables.com/Automatic-Arduino-Pet-Feeder/

9.4 Research Paper

- ✓ Pet Feeding Behavior and Dietary Needs." American Veterinary Medical Association,2023. [Online]. Available: https://www.avma.org/petowners/petcare/pet-nutrition. [Accessed: March 10, 2024].
- ✓ "Arduino Uno Official Documentation." Arduino, 2023. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno. [Accessed: March 10, 2024].
- ✓ Smith, John. "Design and Implementation of an Automatic Pet Feeder System." International Journal of Electronics and Communication Engineering, vol. 5, no. 2, 2020, pp. 112-120.
- ✓ Wayne “Intelligent Food Dispenser (Ifd)” Hari N. Khatavkar, Rahul S. Kini, Suyash K. Pandey, Vaibhav V. Gijare, 20419.
- ✓ A Remote Pet Feeder Control System Via Mqtt Protocol” Wen-sChuan Wu, Ke-Chung Cheng, Peiyu Lin, 2018
- ✓ Iot Based Pet Feeder System” Saurabh A. Yadav, Sneha S. Kulkarni, Ashwini S. Jadhav, Prof. Akshay R. Jain,2018
- ✓ Automatic Pet Feeder Aasavari Kank, Anjali Jakhariye, 2018
- ✓ Pet Feeding Dispenser Using Arduino and Gsm Technology, Smruthi Kumar, 2018.
- ✓ “IEEE Standard for Software Requirements Specifications." IEEE Std 8301998 (Revision of IEEE Std 830-1993), 1998.