

DESIGN AND DEVELOPMENT OF AN AUTOMATED FEEDER ROBOT WITH WATER AND FOOD DISPENSING CAPABILITY, CAMERA SYSTEM AND WHEELED MOBILITY

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering degree in Mechatronics

By

S.S. ARAVIND KUMAR (Reg. No. 40090001)



**DEPARTMENT OF MECHATRONICS
SCHOOL OF MECHANICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with grade “A++” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF MECHATRONICS

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of S.S. Aravind Kumar (Reg. No. 40090001) who carried out the project entitled "**DESIGN AND DEVELOPMENT OF AN AUTOMATED FEEDER ROBOT WITH WATER AND FOOD DISPENSING CAPABILITY, CAMERA SYSTEM, AND WHEELED MOBILITY**" under our supervision from November 2023 to April 2024.

Internal Guide

Mr. V. JAYAPRAKASH, M.Tech., M.I.E., (Ph.D.),

Head of the Department

Dr. B. KANIMOZHI., M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **S.S. ARAVIND KUMAR** (Reg.No. 40090001), hereby declare that the Project Report entitled “**DESIGN AND DEVELOPMENT OF AN AUTOMATED FEEDER ROBOT WITH WATER AND FOOD DISPENSING CAPABILITY, CAMERA SYSTEM AND WHEELED MOBILITY**” done by me under the guidance of **Mr. V. Jayaprakash, M.Tech., M.I.E., (Ph.D.)**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Mechatronics Engineering.

DATE:

PLACE: CHENNAI



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my sincere thanks to **Dr. B. Kanimozhi, M.E., Ph.D.**, Head of the Department, Department of Mechatronics for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr. V. Jayaprakash, M.Tech., M.I.E., (Ph.D.)**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Mechatronics who were helpful in many ways for the completion of the project.

ABSTRACT

The paper focuses on a time-based technique for pet feeding using a robot, addressing issues related to pet owners' unavailability and hygiene concerns. The project aims to develop an automated pet feeding system for indoor pets, utilizing an Arduino Uno as the CPU along with components like a pump motor, actuator, and servo motor. Named the Smart Pet Feeding Machine, this system operates on a scheduled basis and automatically dispenses food and water when a pet approaches, triggered by an ultrasonic sensor. Its architectural design includes a vertical food storage compartment, a motorized dispenser with a dispensing hole, and a pump motor for water. Control of the dispenser is managed by a servo motor, while sensors are used for pet activity monitoring.

Keywords: Food and water dispensing, Actuators, Mobility

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	v
	TABLE OF CONTENTS	vi
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	5
	2.1 Petcare: a smart pet care IOT mobile application	5
	2.2 Automatic Pet Feeder using IOT	5
	2.3 GSM Controlled Dog Feeder and Automatic Water dispenser using raspberry pi	6
	2.4 Automatic Pet Feeder Using Arduino	6
	2.5. IOT Based Pet Feeder System	7
	2.6 Pet feeding Dispenser using Arduino and GSM Technology	7
	2.7 Implementation of an IoT based Pet Care System	8
3	AIM and OBJECTIVE	9
	3.1 AIM	9
	3.1.1 Benefits to the clients	9
	3.2 SCOPE	9

4	MATERIALS and EXPERIMENTAL METHODS	10
4.1	HARDWARE REQUIREMENTS	10
4.1.1	Arduino	10
4.1.1.1	Pin Configuration	11
4.1.2	Node MCU	13
4.1.2.1	Specifications & Features	13
4.1.3	Water Level Sensor	14
4.1.4	PIR Sensor	15
4.1.5	LCD Display	16
4.1.6	Pump motor	17
4.1.6.1	Specifications	18
4.1.7	Ultrasonic Sensor	18
4.1.8	Servo motor	19
4.1.9	Bluetooth Connectivity	19
4.1.10	DC Motor	21
4.2	SOFTWARE REQUIREMENTS	22
4.2.1	Arduino	22
4.2.1.1	Installed Arduino code	25
4.2.1.2	Installed Node MCU code	28
4.2.2	Proteus	31
4.2.2.1	Start using Proteus	32
5	FINAL PRODUCT VIEW	34
6	RESULT and SUMMARY	37
6.1	RESULT	37
6.2	SUMMARY	37
7	CONCLUSION	38
8	REFERENCES	39
9	APPENDIX	

LIST OF FIGURES

FIGURE No	TITLE	PAGE No.
4.1	Block Diagram	10
4.2	Arduino	11
4.3	Pin configuration of Arduino	12
4.4	Node MCU	13
4.5	Water level Sensor	14
4.6	PIR Sensor	15
4.7	LCD Display	16
4.8	Pump Motor	17
4.9	Ultrasonic Sensor	18
4.10	Servo Motor	19
4.11	Bluetooth Module	20
4.12	DC Motor	21
4.13	USB Cable to connect arduino and computer	22
4.14	Sample Code	23
4.15	Selecting Port	23
4.16	Uploading to Arduino	24
4.17 – 4.22	Installed Arduino Code	25 – 27
4.23 – 4.28	Installed Node MCU Code	28 – 30
4.29	Connection in Proteus Software	33
4.30	Simulation in Proteus Software	33
5.1	Side view	34
5.2	Top View	34
5.3	Cross View	35
5.4	Webcam View	35
5.5	Webcam Demonstration	36
5.6	Lcd Displaying	36

LIST OF SYMBOLS AND ABBEREVIATIONS

Kb	Kilobyte
Gnd	Ground
Vin	Input Voltage
Approx.	Approximate
Kohms	kiloohms
MISO	Master In Slave Out
MOSI	Master Out Slave In
SCK	Serial Clock

CHAPTER 1

INTRODUCTION

The majority of people these days find owning pets to be fascinating. However, sufficient care must be given to these creatures. As kids grow into members of our family, feeding them on time becomes increasingly crucial. However, because of our hectic schedules, we neglect to provide our pets the correct food on time. In order to ensure that pets are fed on a regular basis, this study introduces an automatic pet feeding system in response to the aforementioned problem. The servo motor, dispenser, and food storage are all part of the automatic pet feeding system. Additionally, Arduino is used to automate operation control. Adding cameras and sensors to monitor a pet's actions is another way to make anything more technologically advanced. An automatic pet feeding system has a machine that can feed animals (like dogs) on its own at regular intervals while their owner is not there. By utilising the machine, the master is free to go outdoors and perform other tasks without worrying about his pet, and he also avoids having to stay with the pet every time it needs food. There are several ways to fill a pet feeder dish.

In today's world, the allure of owning a pet has captivated many, yet it comes with the responsibility of ensuring proper care for these beloved companions [1]. It becomes our solemn duty to guarantee that our furry friends are fed regularly once they become part of our household. However, our bustling schedules often hinder us from giving our pets the attention they deserve, especially when it comes to ensuring they receive their meals at the right times [2]. To address this common challenge, this article proposes the implementation of an Automatic Pet Feeding System. This system comprises essential components such as a feeding dish, a dispenser, a servo motor, and a designated food storage area. Additionally, Arduino technology is integrated into the system to automate and streamline its operations. For enhanced functionality, users have the option to incorporate cameras and an audio system to monitor their pets and engage in two-way communication with them [3], [4].

The Automated Pet Feeding System empowers pet owners to ensure their furry companions are fed consistently, even in their absence. This innovative equipment alleviates the worry of constantly being present to feed pets, allowing owners to

engage in external activities while knowing their pets' nutritional needs are being met [5]. Equipped with an LCD screen, the pet feeder facilitates programming of feeding schedules and displays relevant data, all managed by an Arduino UNO. This automated pet feeder, powered by Arduino technology, employs an auger mechanism and offers customizable feeding intervals to cater to the unique requirements of each user [6]. Moreover, it boasts a battery-operated internal clock for seamless operation. In comparison to existing pet feeders available in the UK market, this system stands out with its reliability and effectiveness. Unlike some alternatives that may be cumbersome to program and prone to malfunction, our Automatic Pet Feeding System ensures consistent and hassle-free feeding experiences for both pets and their owners [7].

Automated pet feeders represent a recent advancement in pet care, offering pet owners the ability to monitor and feed their animal companions even in their absence [8]. These devices alleviate the need for constant human presence to ensure pets are adequately nourished, saving pet owners time and effort [9]. One notable example is the Internet of Things (IoT) pet feeder, which can be remotely controlled via a mobile application connected to the internet. With programmable features, such as automatic refilling of food and water bowls at designated intervals, these feeders provide convenience and peace of mind to pet owners [10]. Recognizing the importance of proper nutrition for pets, automated feeders ensure pets receive timely and sufficient meals, even when owners are unexpectedly absent or seeking to simplify their responsibilities [11]. By adhering to preset feeding schedules and preventing access to food intended for future meals, these devices address common challenges faced by pet owners, enhancing the well-being of both pets and their caregivers [12]. Moreover, automation technology, coupled with internet connectivity, enables pet owners to manage feeding schedules remotely, from any location with internet access [13]. Incorporation of sensors further streamlines the feeding process, minimizing the need for human intervention [14]. Overall, automatic pet feeders are a great way to alleviate a lot of stress for pet owners and make sure their dogs are getting the nutrients they need without having to lift a finger [15], [16].

By using any internet-connected mobile device, Luayon et al. [17] aimed to build an IoT-based system that would facilitate the remote management and response to canine needs by pet owners. Aimed squarely at Android 6.0 (Marshmallow), the mobile app streamlines the OS's operation for consumers. Researchers Prasad et al.[18] introduce a groundbreaking pet feeder design that transforms the user experience. Utilizing an interactive remote controller, this innovative system eliminates the need for manual adjustments found in previous iterations. This state-of-the-art design boasts several enhancements compared to its predecessors. The central focus of Prithviraj et al [19] research is the acknowledgment of dogs' specific maintenance requirements. Throughout the project's inception, design, and execution phases, the primary emphasis remains on the development of an intelligent pet system. This study highlights the potential of computer, communication, and control technologies to enhance interactions between humans and animals, primarily through Internet of Things (IoT) integration. The overarching goal is to address the common challenge faced by busy caretakers who struggle to allocate time for hands-on pet care amidst their schedules. Consequently, the system aims to be robust enough to alleviate the burdens associated with pet ownership, particularly in the case of dogs. Asaner et al [20]. The advantages of an IoT system that lets dog owners monitor their dogs' location, activity, and health from a distance were explained. The three primary components of this mandatory Internet of Things (IoT) system are a feeding, water dispenser, and litter box. Wi-Fi modules and Arduino Uno devices are linked to the local network to enable communication.

As per Babu et al. [21], pet owners have the capability to remotely connect with the system via their mobile phones, enabling them to feed their dogs from afar. Utilizing GSM technology, pet owners can send messages to activate the system. Upon receiving the signal, the servo motor responsible for dispensing food rotates, while the solenoid valve opens to allow water flow. After the feeding process concludes, the owner is promptly notified, offering a valuable solution for busy households and ensuring timely pet feeding despite hectic schedules [22]. Utilizing the Global System for Mobile Communications (GSM) enables the receipt of signals from the owner's phone. Upon receiving a message from the owner, both the servo motor and solenoid valve are activated through a gadget installed in the house [23]. The valve opens to allow water flow while the servo motor dispenses food

simultaneously. A confirmation text is sent to the owner immediately after the feeding is completed. This innovative system greatly assists individuals with busy routines in fulfilling their responsibility towards their pets' health and well-being. The development of a pet feeding system integrated Raspberry Pi and GSM technology to address this need effectively [24]. GSM technology streamlines data retrieval from the owner, offering convenience and reliability compared to Wi-Fi. The use of Short Message Service (SMS) for notifications was chosen due to its cost-effectiveness and efficient information delivery on mobile devices [25]. The system's core concept revolves around the GSM module and Raspberry Pi for receiving SMS commands from authorized users [26], ensuring uninterrupted pet care and communication from anywhere.

CHAPTER 2

LITERATURE SURVEY

2.1 Petcare: A Smart Pet Care IoT Mobile Application

Alexis Anne A. Luayon , Gio Francis Z. Tolentino , Van Keith B. Almazan and Patrick Eugene S. Pascual

The project's objective is to establish an Internet of Things (IoT) system enabling pet owners to manage their pet's needs and communicate with them via any internet-connected mobile device at any given time. The mobile application is specifically designed for the Android 6.0 (Marshmallow) operating system, allowing users to operate the system through their device settings. Ideally, the Raspberry Pi microcontroller, connected to the household's Internet connection, remains online continuously. The project aims to create a smartphone app that empowers pet owners to monitor and provide exceptional care for their pets. Successful testing of the Android operating system and Internet connectivity validates the project's capability to fulfill its promises based on the mobile app design. Consequently, it can be concluded that the system has been effectively deployed to achieve its intended purpose.

2.2 Automatic Pet Feeder using IOT

Jayaram Kumar Kondapalli, Venkata Ramana Sanepu, Balakrishna Satyam Kothapalli, Shankar Pattabhi Ram Peketi, Venkata Dattu Naveen Kukatla

The project intends to use IoT to produce an efficient automated pet feeder that solves typical feeding problems. A NODEMCU Micro Controller powers the feeder, which has a DC servo motor, relays, a water pump, and other gear. Microcontrollers are carefully configured to handle switches, motors, and other key operations, while mobile devices use Blynk software to communicate with them over the cloud. This setup lets users operate the feeder remotely. The DC servo motor controls the feeding mechanism and the water pump supplies water from a tank to the automated pet feeder. This ensures dogs have enough food and water on a regular basis. The new IoT-based feeder simplifies pet care for owners, enabling them to care for their dogs even while they're away, improving the whole experience.

2.3 GSM Controlled Dog Feeder and Automatic Water dispenser using raspberry pi

Mark Christian G. Cabaltera, Kurt C. Cuaresma, Allen B. Flores,

Dr. Neil P. Balba

This study claims that the owner's phone sends a signal to the GSM. The solenoid valve and servo motor will turn on when the owner sends a message to the gadget at home. A servo motor will turn, opening a valve to release the food and water. Upon completion of the procedure, the owner will get a text message confirming the successful feeding. The goal was to make feeding pets easier for those with hectic schedules. The health of a pet is one of a pet owner's primary responsibilities. When the owner is away from home, make sure the pet doesn't get hungry. A pet feeding system was created utilising GSM technology and a Raspberry Pi. One of the helpful aspects of the gadget is the ability to utilise a GSM to receive data from the owner. GSM is more easy and dependable to use than Wi-Fi, which is why the majority of pet feeders utilise it. Because SMS is the most affordable and efficient method of providing information via mobile devices, its proponents chose it as the module that alerts users. The concept behind SUN is to utilise a Raspberry Pi and GSM module to receive SMS commands from authorised users from any location.

2.4 Automatic Pet Feeder Using Arduino

Mritunjay Subhashchandra Tiwari , Sahil Manoj Hawal, Nikhil Navanath Mhatre, Akshay Ramesh Bhosale, Mainak Bhaumik

The project involves automating a pet feeding machine for a minimum of eight hours every day when all members of the nuclear family are at work to ensure their survival in major cities and other places. The mobile app does this setup automatically. Pet owners always make sure that their pets, especially dogs and cats, are fed, even while they are busy at work. Pets at home can get food at intervals of 30 minutes, and this can be tracked via a mobile app. When leaving the house, the pet's food was stored in a box. When a pet approaches the box when they are hungry, food will be automatically given to them up to a certain amount. Preventing the pets from becoming hungry when no one is home and they are all preoccupied with their regularly scheduled tasks and work at the office or business is a great idea.

2.5 IOT Based Pet Feeder System

Saurabh A. yadav, Sneha S. kulkarni, Ashwini S. jadhav,
Prof. Akshay R. jain

This study proposes a groundbreaking design for a pet feeder that eliminates the manual settings required in earlier models, opting instead for operation through an interactive remote controller. This design incorporates several improvements over its predecessors. Users can adjust the feed volume, intervals between feeds, and feeding times, enhancing flexibility and customization. Additionally, the design features a safety lock for the feeder container, a refill alert, dual power supply with a battery charger, a message alert system to notify the owner if the pet misses a feeding, a priority feeder with a dual serve option allowing multiple feeding times for the owner's selection and one for the pet, among other features. These elements in the pet feeder's design aim to simplify pet care for both the owner and the animal. To address concerns about overfeeding, the system provides detailed information on the pet's feeding habits, including whether it is eating and how much. By dispensing leftover food first, this arrangement minimizes feed waste. Furthermore, the interactive remote control eliminates the need for manual adjustments to the pet feeder, enhancing convenience and efficiency.

2.6 Pet feeding Dispenser using Arduino and GSM Technology

Smruthi Kumar

As the primary focus of this research, the idea of pet owners being able to remotely feed their animals by sending instructions through their mobile phones is investigated in depth. For the purpose of determining whether or not the pet owner has provided a feeding command, the system makes use of GSM technology. After the system has received the message, it will begin to operate the solenoid valve as well as the servo motor. During this time, the solenoid valve opens to let water to flow more freely, and the servo motor, which is responsible for distributing food, turns in the appropriate manner. A notice is sent to the owner after the feeding procedure has been completed by the animal. Individuals within the family who, as a result of their hectic schedules, may find it difficult to personally feed their dogs are the individuals who are intended to benefit from this novel strategy.

2.7 Implementation of an IoT based Pet Care System

Yixing Chen, Maher Elshakankiri

The integration of the Internet of Things (IoT) enables pet owners to monitor their pets' activities, location, and health remotely, enhancing their caregiving capabilities. This system utilizes IoT technology to connect various components such as a litter box, water dish, and food dish, which are essential for a cat's home environment. These subsystems are interconnected through local network Wi-Fi modules linked to Arduino Uno devices. Furthermore, a dedicated smartphone app processes and presents data collected from sensors, providing pet caregivers with insights into their pet's food and water consumption, defecation patterns, timing, duration, and frequency. This consolidated information is accessible through a user-friendly interface, streamlining pet care management.

Additionally, the app features a control function that empowers pet owners to dispense food at their convenience and from any location, adding convenience and flexibility to the pet feeding schedule.

CHAPTER 3

AIM and SCOPE

3.1 AIM

Every pet owner can benefit from our project's ability to remotely and intelligently feed their pets.

3.1.1. *Benefits to the clients:*

- Pet owners will have the knowledge to provide their animals the right amount of food. Data will be transmitted to the PCB by Bluetooth or the internet, and then signals will be sent to the gates that disperse food.
- The app will track the amount of food that pets eat every day, so owners can see how their pets are doing physically. Every day, sensors will be inserted in the food plate to measure the difference in weight. After PCB processes the data, it will be delivered to the mobile app.

There will be an RFID tag for every pet. A meal plate will be spined at the precise angle to expose the associated food when the RFID receiver determines the type of animal.

3.2 SCOPE

Making an automated feeding machine for animals is the main goal of this project. Dairy farms, poultry houses, and domestic pets are all considered throughout the planning process. Animals, like people, need regular dietary checks to ensure optimal health, which in turn improves milk production and quality on dairy farms and egg and chicken quality on poultry farms. Nowadays, it's easy for anyone to have a pet without really committing to keeping them healthy. Pet owners will find this feeding machine useful for keeping track of their pets' dietary needs. It can be managed remotely through an internet-connected smartphone app while the user is at home. The timer feature allows pet owners to schedule and automate feeding times for their pets, even when they are away from home. The exact amount of food that should be in the bowl will be detected by the force sensor to ensure that it does not exceed. Whether we're at the office or on a business trip. Pet sitters charge us exorbitant amounts of money to feed our pets. Not only would this address our issue, but it would also be useful for other pet owners if we could modify a pet feeder to work with an internet of things app.

CHAPTER 4

MATERIALS and EXPERIMENTAL METHODS

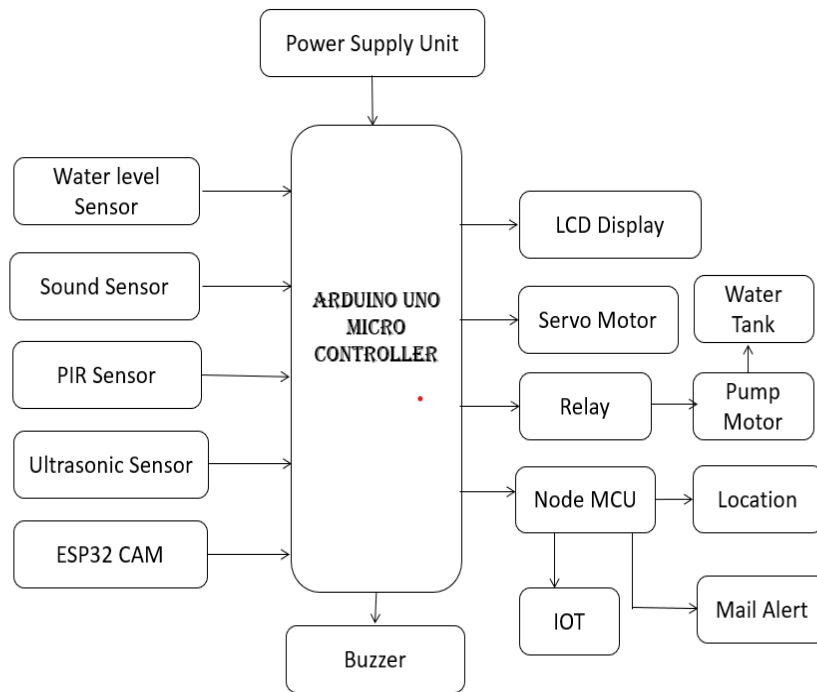


Fig.4.1 Block Diagram

4.1. HARDWARE REQUIREMENTS

4.1.1. ARDUINO

Arduino-based computers are different from regular desktop computers because they can understand and control physical objects. They use a basic microcontroller board and software to do this, and it's all open-source. With Arduino, you can make interactive things by connecting sensors and switches to control lights, motors, and more. You can use Arduino projects alone or connect them to your computer. You can buy ready-made boards or build them yourself, and the software to program them is free.

One type of Arduino board, called Duemilanove, has a 5V Atmel ATmega328 microprocessor with memory for programs and parameters. It's fast and has input/output pins, USB for connecting to a computer, and a power jack for batteries or adapters. Arduino programming utilizes a simplified version of C and C++ languages, which is beneficial for beginners due to its ease of understanding and

learning curve. You can write programs on your computer, transfer them to Arduino, and run them independently. Another popular Arduino board, Uno, has similar features but uses a different method for USB communication. It's also easy to use and comes with everything you need to get started.



Fig.4.2 – Arduino

4.1.1.1. Pin Configuration

The Arduino Uno automatically selects a power source from its USB port or an external power supply. To connect a battery or AC-to-DC adapter like a wall charger to the Arduino Uno board's power connector, use a 2.1mm center-positive plug. You can supply voltage between 6V and 20V, but avoid going below 7V to prevent instability.

The power pins are as follows:

- 5V: This pin provides a stable 5V regulated by the board's regulator. You can power the board through the USB connection (5V), the DC power jack (7-12V), or the VIN pin (7-12V). Avoid directly connecting voltage to the 5V or 3.3V pins to prevent damage.
- 3V3: This pin provides 3.3V with a maximum current draw of 50mA.
- GND: These pins are grounded.

The ATmega328 chip on the Arduino Uno board comes with 32KB of RAM, with half of it reserved for the bootloader, along with 1KB of EEPROM and 2KB of SRAM. In terms of analog inputs, the Uno features six inputs labelled A0 through A5, offering 10-bit resolution for a total of 1024 distinct values. The measuring range of these inputs can be adjusted using the AREF pin and the `analogReference()` function, although they typically measure from ground to 5 volts by default.

For digital operations, the Uno provides 14 digital pins that can be configured as inputs or outputs using the `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. These pins operate at 5V and are equipped with built-in pull-up resistors, capable of handling up to 40mA of current. Some digital pins serve specific purposes:

- Pins 0 (RX) and 1 (TX) are designated for serial data communication.
- Pins 3, 5, 6, 9, 10, and 11 support pulse width modulation (PWM) output.
- Pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK) are dedicated to SPI communication.
- Digital pin 13 includes an integrated LED that illuminates when set to HIGH.

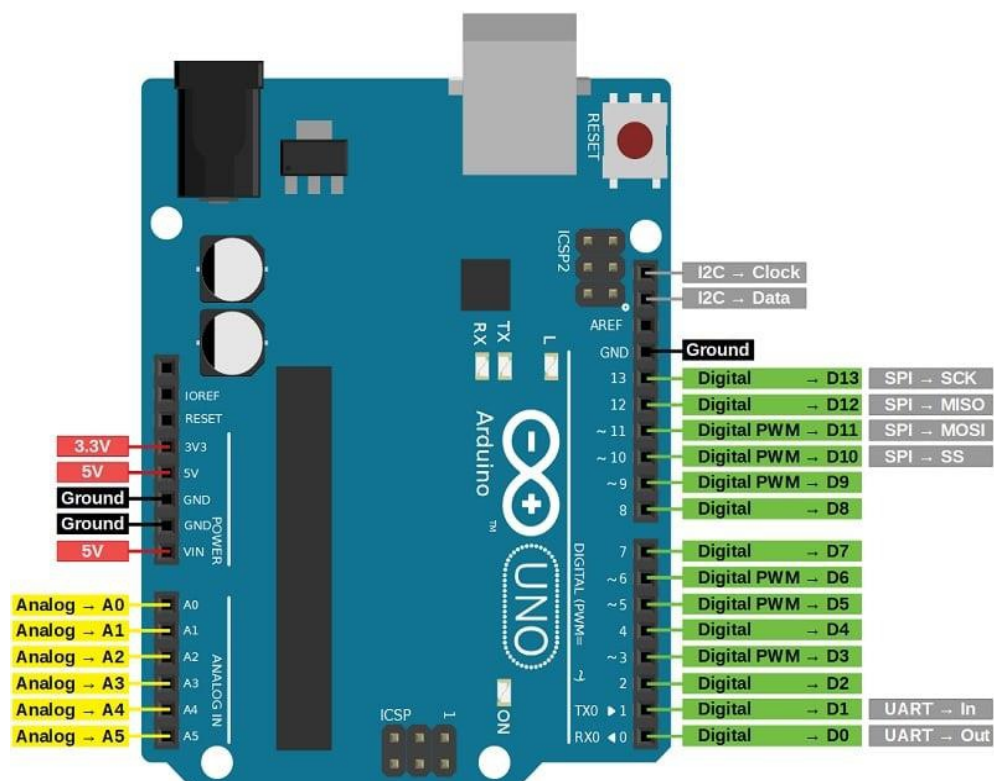


Fig.4.3 – Pin Configuration of Arduino

4.1.2. NODE MCU

The ESP32, combined with a webcam, enables live streaming capabilities for a pet feeder robot, allowing users to remotely monitor their pets in real-time. The principle involves integrating the ESP32 microcontroller with the webcam and setting up a streaming server to broadcast the video feed over the internet. The ESP32 microcontroller provides the necessary processing power and connectivity options, such as Wi-Fi or Ethernet, to establish a connection with the webcam and transmit the video data. The webcam captures live video footage of the pet's activities, which is then processed by the ESP32 and streamed to a designated streaming server or platform.

The streaming server acts as an intermediary, receiving the video feed from the pet feeder robot and broadcasting it over the internet. Users can access the live stream using a web browser or a dedicated streaming application on their devices, allowing them to remotely view their pets' behaviour, interactions, and feeding routines. This setup enhances the pet owner's ability to monitor their pets' well-being and activities, even when they are not physically present. It provides a convenient and interactive way to stay connected with their pets and ensure they are cared for.

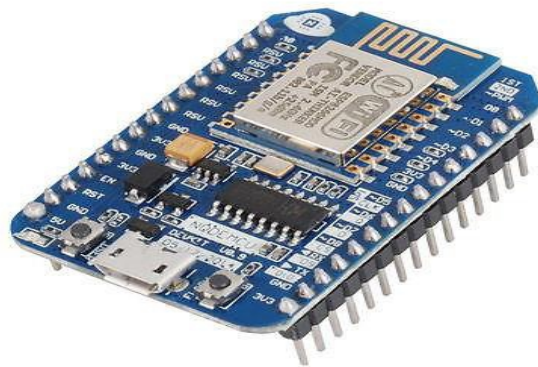


Fig.4.4 - NODE MCU

4.1.2.1. Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V

- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, enabling plug and play
- PCB Antenna
- Small-sized module designed to fit smartly inside IoT projects

4.1.3. WATER LEVEL SENSOR

A water level sensor in this pet feeding robot is designed to detect that the water is dispensed. When integrated into the robot's system, the water level sensor ensures that water is dispensed for a certain time period by signalling the microcontroller to activate the water pump motor to dispense the water. This functionality is often synchronized with an ultrasonic sensor, which measures the distance between the pet and the robot, allowing to dispense the water for a certain amount. By combining these sensors, the pet feeding robot can maintain a consistent water supply for pets while preventing overflows or dry periods, enhancing the overall efficiency and reliability of the feeding system.



Fig.4.5 – Water Level Sensor

4.1.4. PIR SENSOR

A Passive Infrared sensor (PIR) operates by detecting infrared radiation emitted by objects within its field of view. When an object, such as a pet, enters the sensor's detection range, it emits infrared radiation due to its heat. The sensor's lenses are designed to concentrate this incoming infrared radiation onto its elements, which in turn generate an electrical signal proportional to the intensity of the heat. The functionality of a PIR sensor hinges on the contrast in infrared radiation between the detected object and the background environment. This contrast is perceived as a change in the electrical signal produced by the sensor elements. PIR sensors typically employ a dual-element configuration, consisting of two sensors sensitive to infrared radiation. These sensors are positioned slightly apart from each other, allowing the sensor to detect variations in infrared radiation as an object moves across its detection area.

The PIR sensor in a pet feeding robot detects the pet's motion by sensing changes in infrared radiation. When the pet moves within its range, the sensor triggers the feeding mechanism to dispense food and water. This automation ensures that the pet is fed when it approaches the feeding area. PIR sensors come with adjustable sensitivity and time delay settings, allowing customization for specific application needs. Overall, the PIR sensor enhances convenience and ensures timely meals for pets in the feeding robot system.



Fig. 4.6 – PIR Sensor

4.1.4. LCD DISPLAY

A liquid crystal display (LCD) is a type of flat panel display that utilizes liquid crystals to produce text and graphics. The "16x2" standard used to describe LCD dimensions signifies that it has two lines and can display up to 16 characters per line. In a pet feeding robot, the LCD serves as a crucial interface between the operator and the machine, facilitating the transmission of data and instructions. One of the primary functions of the LCD display in a pet feeding robot is to show the current time. This feature allows pet owners to track when their pets were last fed and plan future feeding times accordingly. The time display on the LCD is typically synchronized with the device's internal clock, but users also have the option to manually adjust it if needed.

In addition to displaying the time, the LCD screen on the pet feeding robot can also showcase the preprogrammed feeding schedule. This interface enables users to input precise feeding times and quantities, providing a convenient way to customize feeding routines based on individual pet requirements. The flexibility to adjust feeding schedules according to specific pet needs is a significant advantage for pet owners using the pet feeding robot. It enhances convenience, ensures timely and appropriate feeding, and contributes to the overall well-being of the pets in the care of the robotic system.

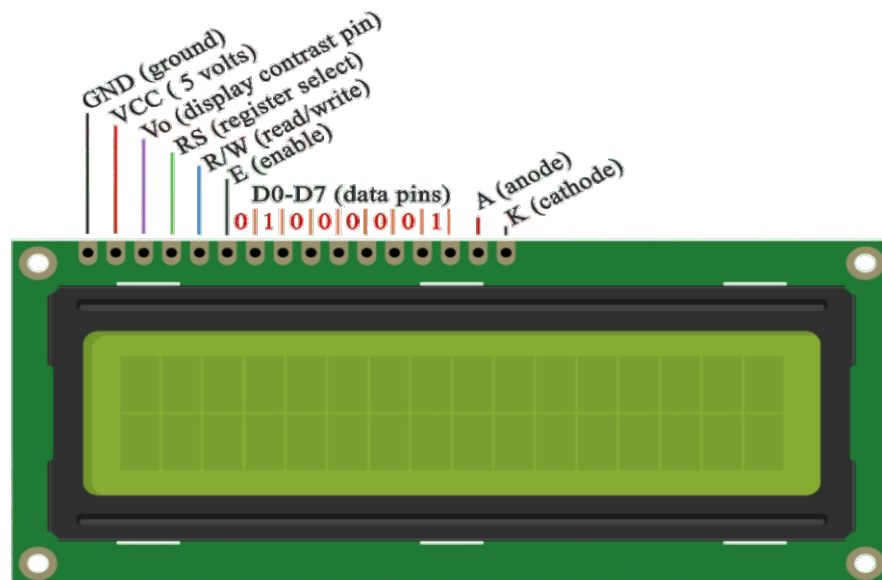


Fig.4.7 – LCD Display (16x2)

4.1.4. PUMP MOTOR

The pump motor works by compressing and releasing a flexible tube create a vacuum that draws water from the tank and then pushes it through the tubing to the water bowl or dispenser. The pump motor is controlled by the robot's microcontroller, which receives signals from sensors or a timer to initiate the water dispensing process. When the pet feeding robot detects that the water level in the bowl is low or based on a preset schedule, it activates the pump motor to start the water dispensing cycle. The microcontroller sends a signal to the pump motor, which begins rotating the rollers that compress and release the tubing. As the tubing is compressed, it creates suction that pulls water from the tank into the tubing. Once the tubing is released, the water is pushed through the tubing and into the water bowl.

The design of the peristaltic pump ensures that the water remains uncontaminated during the dispensing process. Since the water only comes into contact with the inside surface of the tubing, there is minimal risk of contamination from external sources. This is especially important for ensuring the health and well-being of pets, as they require clean and fresh water for hydration. Additionally, the pump motor can be equipped with features such as adjustable flow rates to control the amount of water dispensed per cycle. This allows the pet feeding robot to tailor the water supply to the pet's needs, ensuring that they have access to the right amount of water throughout the day.



Fig.4.8 – Pump Motor

4.1.6.1 Specifications

- Model: R385
- Rated Voltage: DC 6V to 12V (1 amp)
- Working Current: 0.5A to 0.7A (maximum)
- Power: 4W-7W
- Maximum Lift: 3 meters
- Maximum Suction: 2 meters
- Maximum Water Temperature: 80°C
- Pump Size: Approximately 90mm * 40mm * 35mm
- Fluid Temperature Range: 0-100°C
- Input/Output Tube Diameter: Outer 8.5mm, Inner 6mm (approximately)

4.1.7. ULTRASONIC (UV) SENSORS or ULTRASOUND SENSORS

An ultrasonic sensor functions by emitting high-frequency sound waves and then measuring the time it takes for these waves to bounce off nearby objects and return to the sensor. This principle, similar to echolocation used by animals like bats and dolphins, allows the sensor to precisely determine distances to objects in its vicinity, including pets.

In the pet feeding robot, the ultrasonic sensor serves a vital function by detecting the presence of the pet near the feeding area. Once the sensor detects the pet within a specified range, it sends a signal to the robot's control system. This signal then activates the feeding mechanism, allowing food and water to be dispensed from the tank for the pet's feeding.



Fig.4.9 – Ultrasonic Sensor

4.1.8. SERVO MOTORS

Electromechanical servo motors regulate angular or linear position precisely. This pet feeding robot uses a servo motor to open and close a lid or hatch activated by ultrasonic sensors. This procedure trains the servo motor to open or close based on sensor input. The robot's control system receives a signal from the ultrasonic sensor when a pet is detected. After the control system commands the servo motor to spin to a specified degree, the lid opens to reveal the food or water compartment. This technique opens the cover for a certain time, letting the pet feed or drink comfortably.

The control system signals the servo motor to return to its starting position and close the lid after the predetermined time period or when the sensor no longer detects the pet. Its automatic system allows the pet feeding robot to quickly provide food and water to the pet and secure the compartment to prevent contamination or spillage. By merging the servo motor and ultrasonic sensor, the pet feeding robot can provide a clean, efficient, and user-friendly feeding experience.

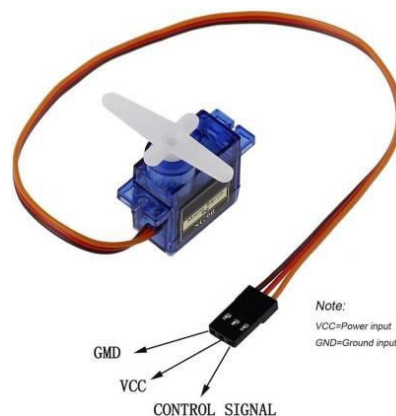


Figure: 4.10 – Servomotor

4.1.9. BLUETOOTH CONNECTIVITY

The pet feeding robot's Bluetooth module permits wireless connection between the robot and a mobile device, giving the user control over the robot's movement (left, right, front, and rear) using a mobile application. The mobile smartphone serves as a remote control interface after the user opens the application and connects to the robot via Bluetooth. Through the Bluetooth module, the programme gives the robot orders that tell it to move left, right, forward, or backward.

The following steps are involved in the principle of operation:

1. Bluetooth Pairing: First, the robot's Bluetooth module and the mobile device must be associated. Usually, the user chooses the robot's Bluetooth module and creates a connection through the settings of the mobile application.

2. Command Transmission: The user may instruct the robot to move by sending movement orders from the mobile application once the Bluetooth connection has been made.

3. Microcontroller Processing: The robot's control system microcontroller decodes the commands into particular actions for the DC motors by interpreting the data packets. As an illustration:

- Front Movement: To advance the robot in the direction of the goal, the microcontroller turns on the DC motors in the front direction.

- Back Movement: To drive the robot backward, the microcontroller flips the polarity of the DC motors.

- Left and Right Movement: The robot may turn left or right by adjusting the speed and direction of the DC motors on either side.

4. DC Motor Control: The DC motors are in charge of converting the microcontroller's orders into motion. The DC motors change their speed and rotation to accomplish the needed motion based on the intended direction (forward, rear, left, or right).

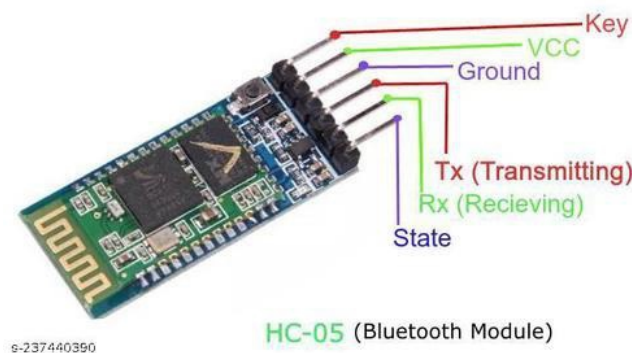


Fig.4.11 – Bluetooth Module

4.1.10 DC MOTORS

The movement system of the pet feeding robot relies on the integration of four DC motors, each connected to a wheel to facilitate locomotion. These motors are controlled by an Arduino microcontroller, which communicates with a Bluetooth module to receive instructions from a mobile device or a remote control interface. The principle of operation involves coordinating the rotations of these motors to achieve different types of movement. For forward motion, the two motors on each side of the robot rotate in the same direction, propelling the robot forward. Conversely, for backward movement, the motors rotate in opposite directions, causing the robot to move in reverse.

When the robot needs to turn left or right, the Arduino microcontroller adjusts the speed and direction of the motor rotations accordingly. For instance, to turn left, the right-side motors may slow down or reverse while the left-side motors continue forward rotation, causing the robot to pivot leftward. The opposite configuration applies for turning right. This intricate coordination of motor rotations allows the pet feeding robot to navigate smoothly and accurately within its environment, ensuring it can reach its destination and perform tasks such as food dispensing or pet monitoring effectively. The Bluetooth module serves as the wireless communication link between the user's commands and the robot's motor control system, providing a convenient and versatile means of interaction.



Fig.4.12 – DC Motor

4.2 SOFTWARE REQUIREMENTS

4.2.1. ARDUINO

The main Arduino microcontroller programming tool is the Arduino IDE. It works on Windows, macOS, and Linux, making it accessible to many users. The IDE helps create, edit, and upload microcontroller code to ensure proper operation.

The Arduino IDE has key features such as:

1. Compatibility with digital and analogue signals, enabling diverse programming.
2. USB connection to the microcontroller for easy code uploading and hardware interaction.
3. Support for C, C++, Java, and others, allowing developers to pick their chosen language for Arduino applications.

The Arduino IDE also has a text editor for writing code, a message box for displaying output and errors, a text terminal for communicating with the microcontroller, a toolbar with commonly used functions, and menus for accessing additional settings and tools.

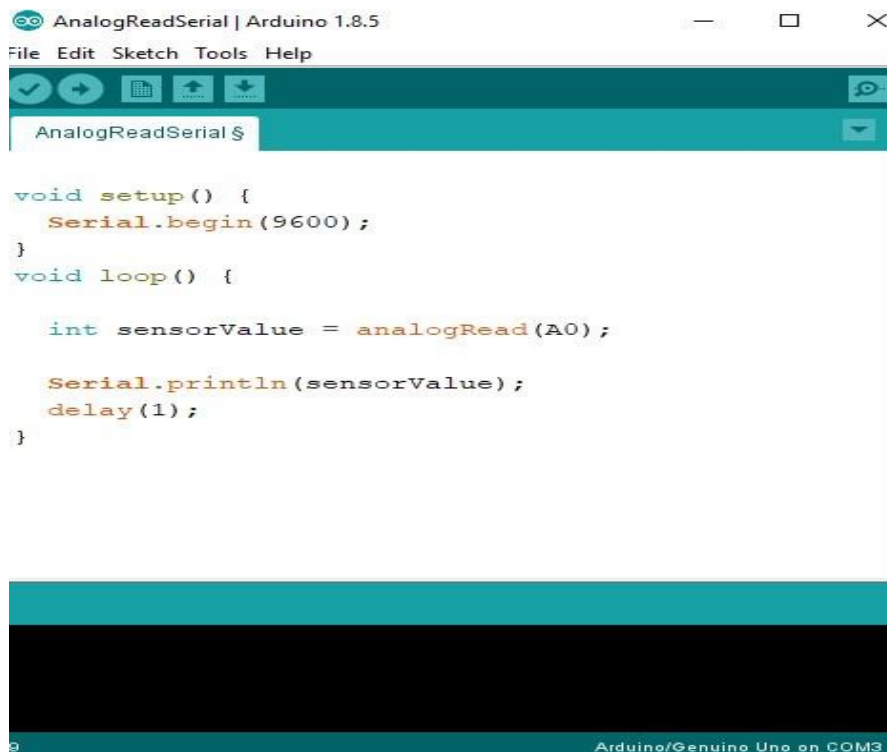
Working:

Step 1. Connecting the Arduino device with the computer using a USB.



Fig. 4.13. USB Cable to connect Arduino and Computer

Step 2. Writing code in selected language.



The screenshot shows the Arduino IDE interface with the file 'AnalogReadSerial.ino' open. The code is as follows:

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  int sensorValue = analogRead(A0);  
  Serial.println(sensorValue);  
  delay(1);  
}
```

The status bar at the bottom indicates 'Arduino/Genuino Uno on COM3'.

Fig.4.14 – Sample Code

Step 3 - Selecting the port from of the available ports.

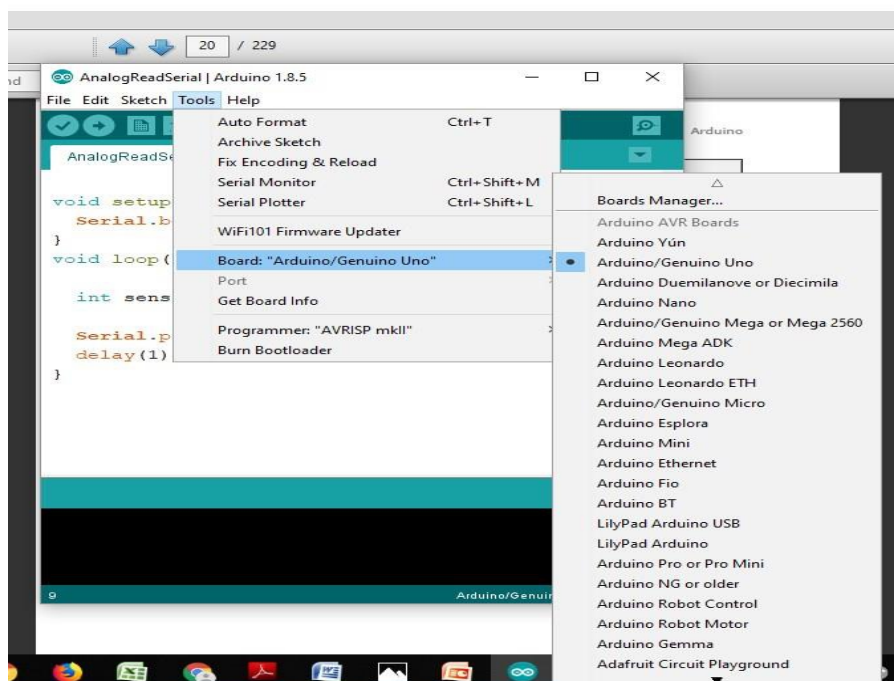


Fig.4.15 – Selecting port

Step 4 - Porting the programme to the Arduino UNO board.

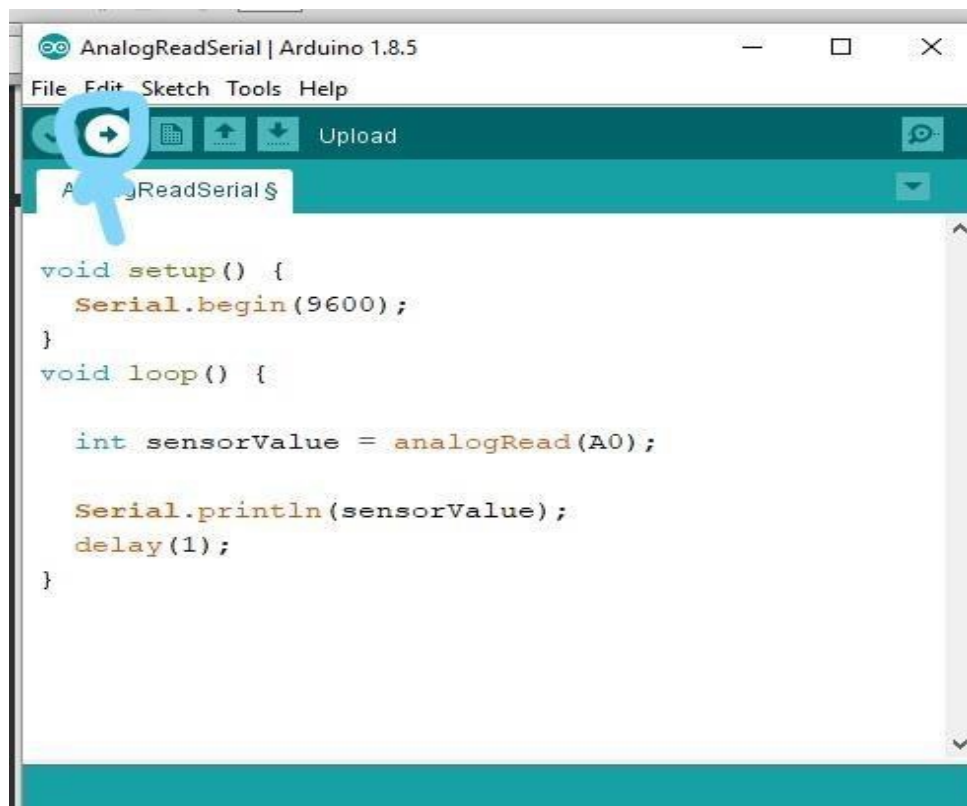
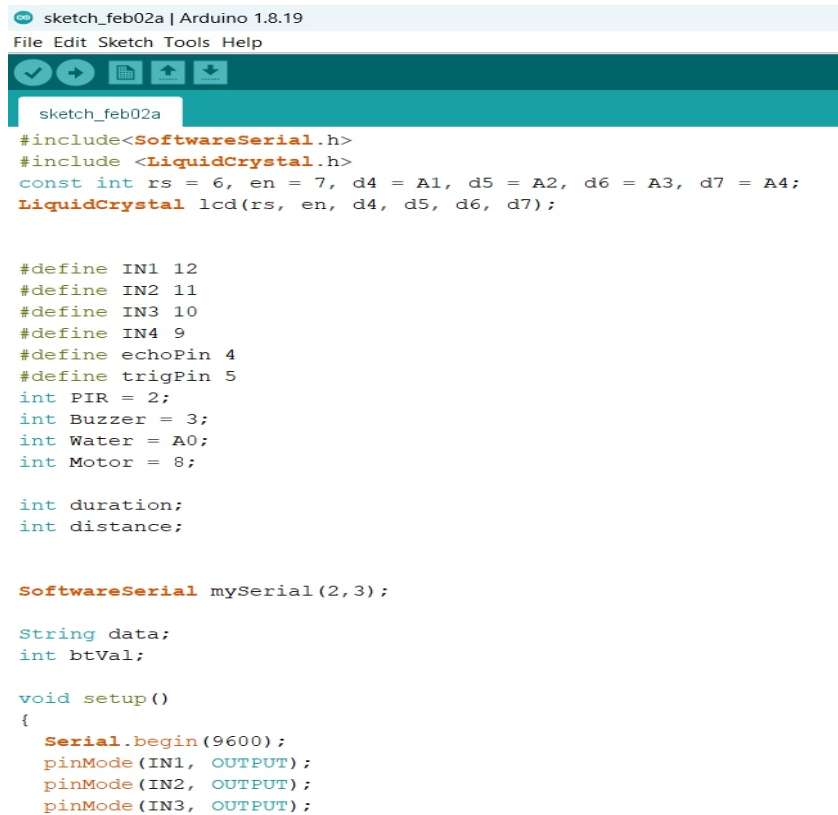


Fig.4.16 – Uploading to Arduino

4.2.1.1. Installed Arduino Code (Fig.4.17 – 4.22) :



```
sketch_feb02a | Arduino 1.8.19
File Edit Sketch Tools Help

sketch_feb02a
#include<SoftwareSerial.h>
#include <LiquidCrystal.h>
const int rs = 6, en = 7, d4 = A1, d5 = A2, d6 = A3, d7 = A4;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

#define IN1 12
#define IN2 11
#define IN3 10
#define IN4 9
#define echoPin 4
#define trigPin 5
int PIR = 2;
int Buzzer = 3;
int Water = A0;
int Motor = 8;

int duration;
int distance;

SoftwareSerial mySerial(2,3);

String data;
int btVal;

void setup()
{
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
```



```
sketch_feb02a | Arduino 1.8.19
File Edit Sketch Tools Help

sketch_feb02a
  pinMode(IN4, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(Water, INPUT);
  pinMode(PIR, INPUT);
  pinMode(Buzzer, OUTPUT);
  pinMode(Motor, OUTPUT);

  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);

  lcd.begin(16, 2);
  lcd.setCursor(1, 0);
  lcd.print("Pet Feeding");
  lcd.setCursor(2, 1);
  lcd.print("Robot");
  delay(1000);

  mySerial.begin(9600);
}

void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
sketch_feb02a | Arduino 1.8.19
File Edit Sketch Tools Help

sketch_feb02a

duration = pulseIn(echoPin, HIGH);

distance = duration * 0.034 / 2;
int PIR = digitalRead(2);
int Water = analogRead(A0);

lcd.clear();
lcd.setCursor(0, 0);lcd.print("D:");lcd.setCursor(3, 0);lcd.print(distance);
lcd.setCursor(8, 0);lcd.print("M:");lcd.setCursor(10, 0);lcd.print(PIR);
lcd.setCursor(0, 1);lcd.print("WL:");lcd.setCursor(4, 1);lcd.print(Water);

Serial.print("Distance :");Serial.print(distance);Serial.print(" ");
Serial.print("Motion Detected :");Serial.print(PIR);Serial.print(" ");
Serial.print("Water Level :");Serial.print(Water);Serial.println(" ");
delay(500);

if (distance < 15)
{
    digitalWrite(Buzzer,HIGH);
    digitalWrite(Motor,HIGH);
}
else if(Water >50)
{
    digitalWrite(Buzzer,HIGH);
}
digitalWrite(Buzzer,LOW);
digitalWrite(Motor,LOW);

while (mySerial.available())
{
    {
```

```
sketch_feb02a | Arduino 1.8.19
File Edit Sketch Tools Help

sketch_feb02a $

    data = mySerial.readStringUntil('\n');
}

    btVal = (data.toInt());

switch (btVal)
{
    case 1:
        forward();
        break;

    case 2:
        reverse();
        break;

    case 3:
        left();
        break;

    case 4:
        right();
        break;

    case 5:
        stoprobot();
        break;
}
}
if (mySerial.available() < 0)
{
    //Serial.println("No Bluetooth Data ");
}
```

```
void forward()
{
    digitalWrite (IN1, HIGH);
    digitalWrite (IN2, LOW);
    digitalWrite (IN3, LOW);
    digitalWrite (IN4, HIGH);
}

void reverse()
{
    digitalWrite (IN1, LOW);
    digitalWrite (IN2, HIGH);
    digitalWrite (IN3, HIGH);
    digitalWrite (IN4, LOW);
}

void left()
{
    digitalWrite (IN1, LOW);
    digitalWrite (IN2, LOW);
    digitalWrite (IN3, LOW);
    digitalWrite (IN4, HIGH);
}

void right()
{
    digitalWrite (IN1, HIGH);
    digitalWrite (IN2, LOW);
    digitalWrite (IN3, LOW);
    digitalWrite (IN4, LOW);
}

void stoprobot()
{
    digitalWrite (IN1, LOW);
    digitalWrite (IN2, LOW);
    digitalWrite (IN3, LOW);
    digitalWrite (IN4, LOW);
}
```

4.2.1.2. Code for NodeMCU (Fig.4.23 – 4.28):

```
webcam | Arduino 1.8.19
File Edit Sketch Tools Help

webcam $
#include "esp_camera.h"
#include <WiFi.h>
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_http_server.h"

const char* ssid = "OPPO F5YOUTH";
const char* password = "Aravind@02";

#define PART_BOUNDARY "12345678900000000000000987654321"

#define CAMERA_MODEL_AI_THINKER
// #define CAMERA_MODEL_M5STACK_PSRAM
// #define CAMERA_MODEL_M5STACK_WITHOUT_PSRAM

// Not tested with this model
// #define CAMERA_MODEL_WROVER_KIT

#if defined(CAMERA_MODEL_WROVER_KIT)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM  -1
#define XCLK_GPIO_NUM    21
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27

#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      19
#define Y4_GPIO_NUM      18
#define Y3_GPIO_NUM       5
#define Y2_GPIO_NUM       4

#elif defined(CAMERA_MODEL_M5STACK_PSRAM)
#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     25
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       32
#define VSYNC_GPIO_NUM    22
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_M5STACK_WITHOUT_PSRAM)
#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     25
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       17
#define VSYNC_GPIO_NUM    22
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21
```

```
webcam | Arduino 1.8.19
File Edit Sketch Tools Help

webcam $
#include "esp_camera.h"
#include <WiFi.h>
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_http_server.h"

const char* ssid = "OPPO F5YOUTH";
const char* password = "Aravind@02";

#define PART_BOUNDARY "12345678900000000000000987654321"

#define CAMERA_MODEL_AI_THINKER
// #define CAMERA_MODEL_M5STACK_PSRAM
// #define CAMERA_MODEL_M5STACK_WITHOUT_PSRAM

// Not tested with this model
// #define CAMERA_MODEL_WROVER_KIT

#if defined(CAMERA_MODEL_WROVER_KIT)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM  -1
#define XCLK_GPIO_NUM    21
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27

#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      19
#define Y4_GPIO_NUM      18
#define Y3_GPIO_NUM       5
#define Y2_GPIO_NUM       4

#elif defined(CAMERA_MODEL_M5STACK_PSRAM)
#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     25
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       32
#define VSYNC_GPIO_NUM    22
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_M5STACK_WITHOUT_PSRAM)
#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     25
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       17
#define VSYNC_GPIO_NUM    22
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21
```

```

webcam | Arduino 1.8.19
File Edit Sketch Tools Help

webcam$

#elif defined(CAMERA_MODEL_AI_THINKER)
    #define FWDN_GPIO_NUM    32
    #define RESET_GPIO_NUM  -1
    #define XCLK_GPIO_NUM    0
    #define SIOD_GPIO_NUM    26
    #define SIOC_GPIO_NUM    27

    #define Y9_GPIO_NUM      35
    #define Y8_GPIO_NUM      34
    #define Y7_GPIO_NUM      39
    #define Y6_GPIO_NUM      36
    #define Y5_GPIO_NUM      21
    #define Y4_GPIO_NUM      19
    #define Y3_GPIO_NUM      18
    #define Y2_GPIO_NUM      5
    #define VSYNC_GPIO_NUM    25
    #define HREF_GPIO_NUM    23
    #define PCLK_GPIO_NUM     22
#else
    #error "Camera model not selected"
#endif

static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary=" PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length: %u\r\n\r\n";

httpd_handle_t stream_httpd = NULL;

static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);

```

```

webcam | Arduino 1.8.19
File Edit Sketch Tools Help

webcam$

res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
if(res != ESP_OK){
    return res;
}

while(true){
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Camera capture failed");
        res = ESP_FAIL;
    } else {
        if(fb->width > 400){
            if(fb->format != PIXFORMAT_JPEG){
                bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
                esp_camera_fb_return(fb);
                fb = NULL;
                if(!jpeg_converted){
                    Serial.println("JPEG compression failed");
                    res = ESP_FAIL;
                }
            } else {
                _jpg_buf_len = fb->len;
                _jpg_buf = fb->buf;
            }
        }
    }
    if(res == ESP_OK){
        size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
        res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
    }
    if(res == ESP_OK){
        res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
    }
    if(res == ESP_OK){
        res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY, strlen(_STREAM_BOUNDARY));
    }
    if(fb){

```

```

webcam | Arduino 1.8.19
File Edit Sketch Tools Help

webcam $
    if(fb){
        esp_camera_fb_return(fb);
        fb = NULL;
        _jpg_buf = NULL;
    } else if(_jpg_buf){
        free(_jpg_buf);
        _jpg_buf = NULL;
    }
    if(res != ESP_OK){
        break;
    }
}
return res;
}

void startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    config.server_port = 80;

    httpd_uri_t index_uri = {
        .uri       = "/",
        .method     = HTTP_GET,
        .handler    = stream_handler,
        .user_ctx   = NULL
    };

    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(stream_httpd, &index_uri);
    }
}

```

```

webcam | Arduino 1.8.19
File Edit Sketch Tools Help

webcam $
void setup() {
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

    Serial.begin(115200);
    Serial.setDebugOutput(false);

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG;

```

4.2.2 Proteus software:

Proteus 8 is the finest simulation programme for several microcontroller architectures. Its main appeal stems from the fact that almost any microcontroller can be found in it. It is therefore a helpful tool for testing software and embedded systems for aficionados in electronics. You may simulate your microcontroller programming with Proteus 8 Simulation Software. Proteus 8 Software is a one-stop shop for hobbyists and students since it enables you to make PCBs immediately after designing your circuit. I think you now know the fundamentals of what proteus software is.

In order to co-simulate whole systems based on microcontrollers, Proteus integrates circuit simulation, microprocessor models, and animated components. This ideal instrument allows programmers to run real-time tests of their microcontroller concepts prior to creating a physical prototype. In this programme, the user may manipulate the layout by means of buttons, switches, and, when connected to a computer, LED and LCD displays. Proteus offers a customisable display, many debugging tools, and single stepping for a clean design prior to hardware development. Use Proteus to simulate the connection between any connected analogue or digital electrical equipment and software running on a microcontroller. Proteus was first created as a system utility that enabled users to edit text and binary files and create CGI scripts on a variety of operating systems, including Windows, Unix, and DOS.

Subsequently, hundreds of more specialised functions for Windows were added to the language, such as the ability to create console apps, network and serial connections, database queries, ISAPI scripting (for IIS), system services, and keyboard emulation. While the interpreter is still available for Linux, the most of these additional capabilities are only available on the Windows version.

Proteus was meant to be easy to use, efficient, and all-inclusive—that is, readable, consistent, and practical. Proteus script readability, string manipulation capabilities, and comprehensive data structure support (including sets, AVL trees, stacks, queues (single or double), arrays, and bit maps) are among its greatest features.

4.2.2.1 Start Using Proteus

1. From the Start menu, choose PROTEUS, then ISIS (blue).
2. Click Devices (P) to bring up a part.
3. Enter the name of the PIC16F84A.
4. After writing an LED, choose the green led.
5. To get a resistor, write 1k and choose the first component.

Analogue, Microprocessor, TTL, CMOS, and ECL ICs are all covered by the pre-built libraries that come with Proteus Capture. All sorts of semiconductors, such as field-effect transistors (FETs), diodes (Diodes), and bipolar semiconductors (Bipolars), are available in the library. A growing collection of over 10,000 manufacturer-specific components from companies including National Semiconductor, Philips, Motorola, Teccor, Texas Instruments, Dallas, and Zetex is also available.

By importing BSDL (Boundary Scan Description Library) files, the creation and packaging of bigger components may be done partially automatically and with little effort. Included with Proteus are libraries that provide specified settings for PCB packaging and simulator models. Because relays can handle both homogeneous and heterogeneous multi-element devices, you can draw their connections and coil in different places. Splitting up the connecting pins is another option. Common editing tools are used to make device alterations on the design itself. You can build gadgets out of arcs, circles, lines, boxes, and unique pin objects. It is possible to draw and utilise symbols that are not electrical in simplified mechanical or block diagrams.

Simulation in PROTEUS Software

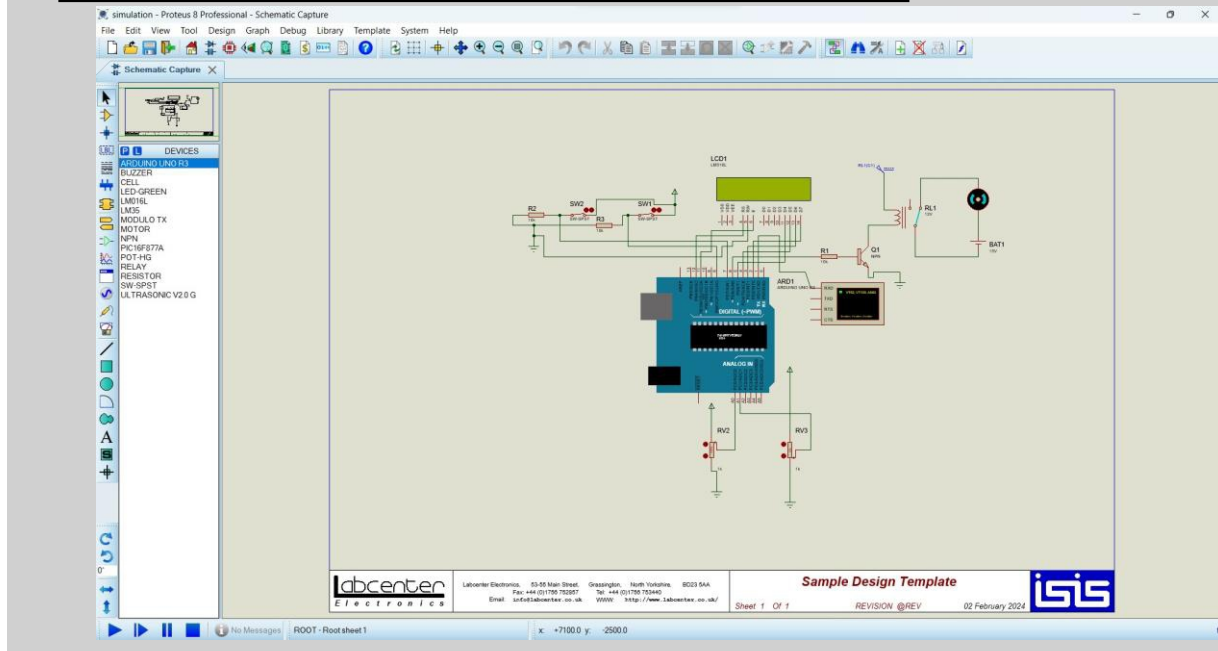


Fig.4.29 -Connection in Proteus software

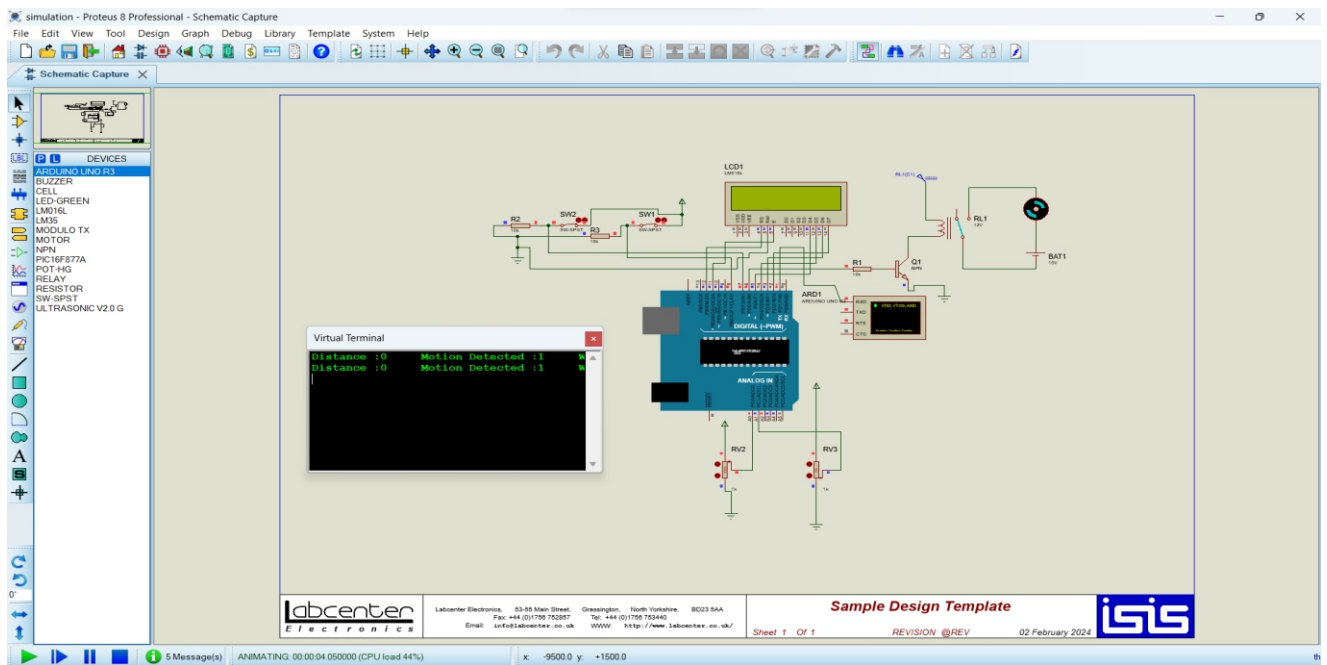


Fig.4.30 - Simulation in Proteus software

CHAPTER 5

FINAL PRODUCT VIEW



Fig. 5.1 – Side View

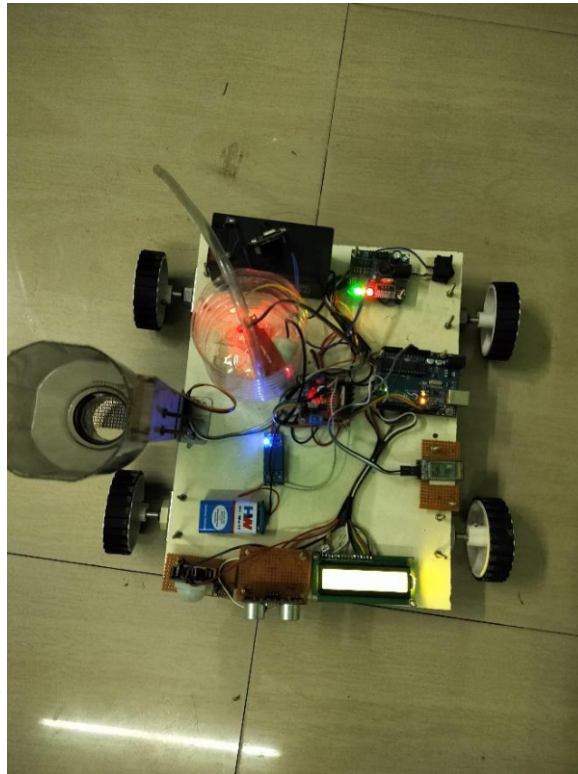


Fig.5.2 – Top View

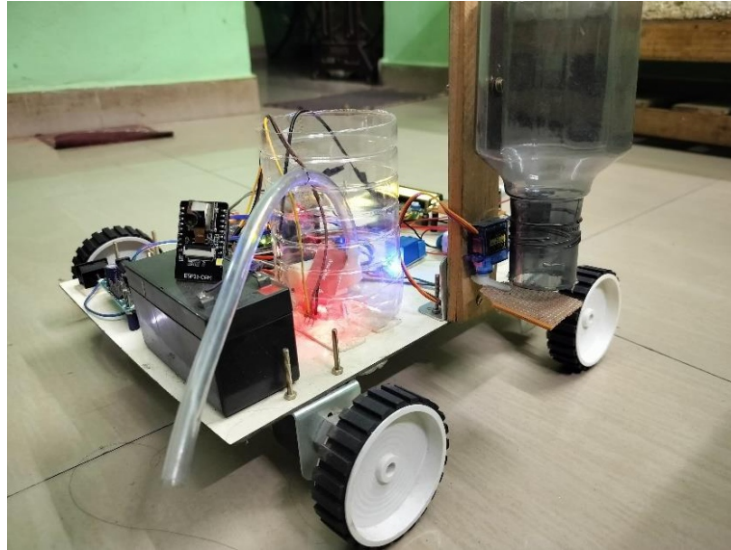


Fig.5.3 – Cross View

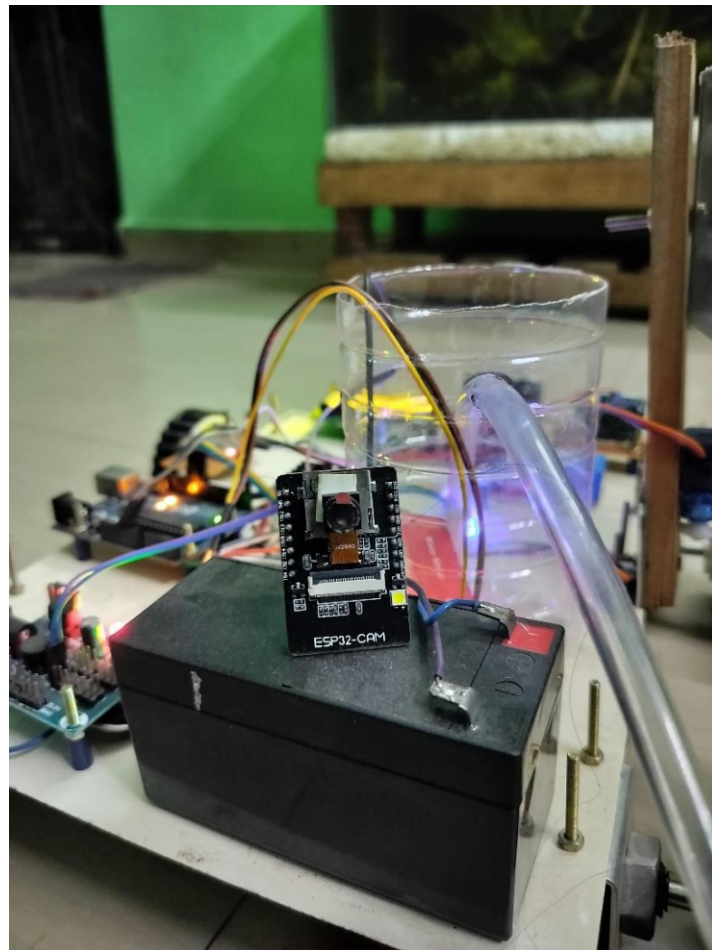


Fig.5.4 – Webcam View

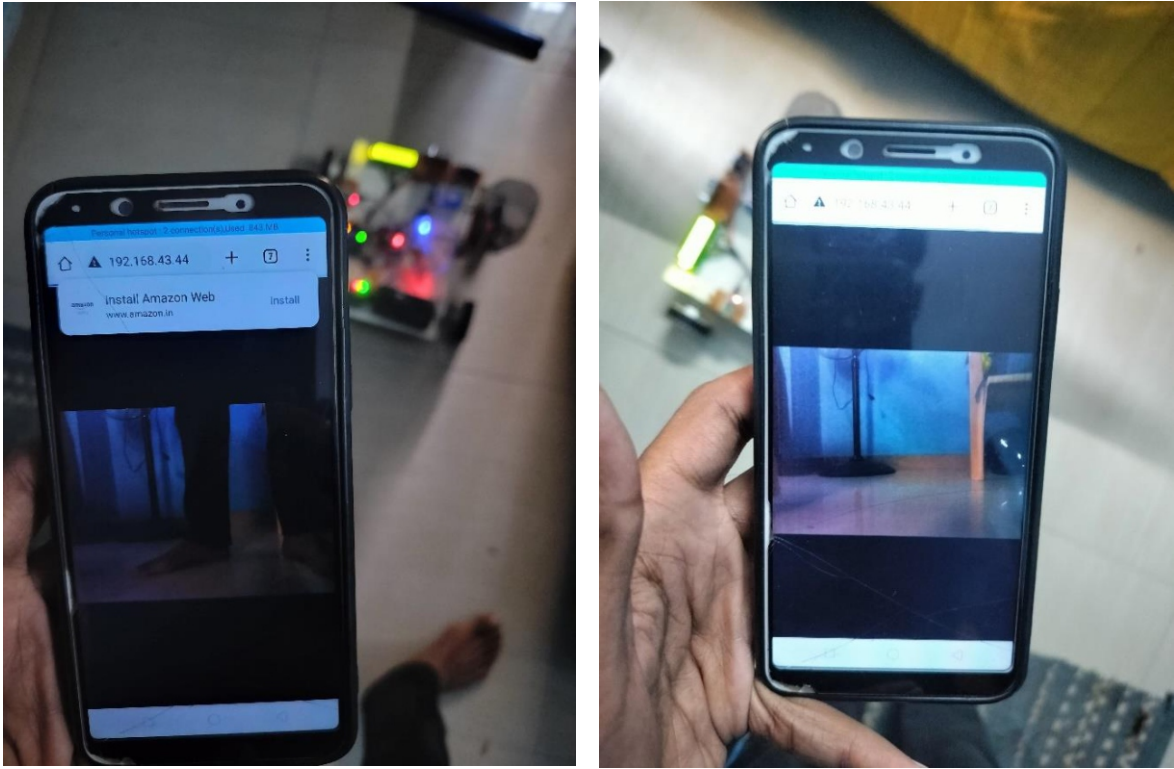


Fig.5.5 : Web cam Demonstration



Fig.5.6 – LCD displaying

CHAPTER 6

RESULT and SUMMARY

6.1. RESULT

The automatic pet feeder efficiently fulfils its objective of feeding pets in the absence of their owners. It operates on a 9V DC power supply and uses a servomotor to rotate the propeller, delivering food into the pet's plate as programmed in the NodeMCU. The entire system is connected to a mobile phone via the internet using an online cloud portal known as the IoT cloud. If there are any interruptions in the cloud app, they are communicated to the microcontroller unit through the internet. For example, when the user presses the "on" button on the IoT app, a signal is sent to the NodeMCU. The NodeMCU subsequently activates the servo motor, which propels the food into the bowl.

6.2. SUMMARY

The pet feeder robot with wheeled mobility, camera, and food and water dispensing capabilities is an advanced solution for pet care. It moves around effortlessly, captures live video feed for remote monitoring, and dispenses food and water as needed. Controlled via a mobile app, it offers convenience and peace of mind to pet owners. With integrated sensors for navigation and obstacle avoidance, it operates autonomously within the home environment. This robot takes a more all-encompassing approach to pet care by keeping dogs well-hydrated, fed, and watched over even when their owners aren't there.

CHAPTER 7

CONCLUSION

7.1. CONCLUSION

There has been a recent uptick in research on human-device interactions, with a particular emphasis on developing more organic and intuitive means of service request processing. The integration of pet control and IoT technology presents exciting opportunities for future developments, including smart-home technologies like smart pet doors and pet feeders. These advancements not only improve pet monitor systems using IoT but also cater to the demands of pet owners. The IoT concept opens up new operational methods, device connections, and even a clean-slate approach to technology. While the full operational definition is being finalized, there are numerous research opportunities to explore.

The next step involves integrating other pet care devices such as litter boxes and pet cams into the system to meet diverse owner needs and cover aspects like health monitoring and pet entertainment. Additionally, as IoT expands globally, the challenge of connecting numerous networking devices around the world arises. Future research will focus on studying IoT gateways and long-distance detection of pets, further enhancing the capabilities and reach of IoT technology in pet care and monitoring.

REFERENCES

- [1] Asaner, U. B., & Elibol, A. (2018). Low-cost IoT design and implementation of a remote food and water control system for pet owners. *Hittite Journal of Science and Engineering*, 5(4), 317-320.
- [2] Chen S., Zhong K, Cai Y.L. (2011). The Design and Application of the Water Temperature Control System for Large Aquaculture Pond. *Measuring Technology and Mechatronics Automation (ICMTMA), 2011 Third International Conference on Measuring Technology and Mechatronics Automation*, Volume: 3 Publication Year: 2011, Page(s): 737-739
- [3] Constantin D. O. (2011). GSM Infrastructure Used for Data. Transmission The 7th international symposium on advanced topics in electrical engineering
- [4] Francisco A.A., Vignaud G.A. (2013). Using opensource platform for trajectory control of DC motors, *IEEE*
- [5] Guifen G., Guili P. (2010). The Survey of GSM Wireless Communication System *International Conference on computer and Information Application (ICCIA 2010)*. Pp121-124
- [6] H. Ning and H. Liu, "Cyber-Physical-Social Based Security Architecture for Future Internet of Things," *Advanced in Internet of Things*, Vol. 2, No. 1, 2012, pp. 1-7. doi:10.4236/ait.2012.21001
- [7] I. F. Akyildiz, W. Su, Y. Sankar Subramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, Vol. 38, No. 4, 2002, pp. 393-422. doi:10.1016/S1389-1286(01)00302-4
- [8] Jayarathne, D. L. S. T. "Design And Implement of IoT-Based Pet Food Feeder Robot."
- [9] Luayon, A. A. A., Tolentino, G. F. Z., Almazan, V. K. B., Pascual, P. E. S., & Samonte, M. J. C. (2019, January). PetCare: a smart pet care IoT mobile application. In *Proceedings of the 10th International Conference on E-Education, E-Business, E-Management and E-Learning* (pp. 427-431).
- [10] M. F. Shaari, M. E. I. Zulkefly, M. S. Wahab, F. Esa (2011). Aerial Fish Feeding System. *International Conference on Mechatronics and Automation*, *IEEE*
- [11] M. Kranz, P. Holleis and A. Schmidt, "Embedded Interaction Interacting with the Internet of Things," *IEEE Internet Computing*, Vol. 14, No. 2, 2010, pp. 46-53. doi:10.1109/MIC.2009.141 40

- [12] Prasad, D. D., Reddy, K. S. R., Kumar, P. R., Satyanarayana, B. V. V., Pathi, A. M. V., & Kumar, G. P. (2023, November). Feeding Pets using Automation Technology and Google Assistant. In 2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 224-229). IEEE.
- [13] Prithviraj, V., & Sriharipriya, K. C. (2022, April). Smart Pet Feeder System Based on Google Assistant. In 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1-6). IEEE.
- [14] Radhika G, Shoba K., Terry S., and Maryam K. (2013). A Competition-Based Approach for Undergraduate Mechatronics Education Using the Arduino Platform
- [15] Rozita T., Salah A. A., Kok W. C., Mok V. H. (2013). Smart GSM Based Home Automation System Conference on Systems, Process & Control (ICSPC2013), IEEE, pp306-309
- [16] R. Kazemi, R. Mosayebi, S.M. Etemadi, M. Boloursaz and F. Behnia (2012). A lower Capacity Bound of Secure End Data Transmission via GSM Network 6th International Symposium on Telecommunications (IST'2012), pp1015-1020
- [17] S. J. Yeoh, F. S. Taip, J. Endan, R. A. Talib and M.k. S. Mazlina (2009). Development of Automatic Feeding Machine for Aquaculture Industry. *Pertanika J. Sci. & Technol.* 18 (1): 105-110 (2010)
- [18] Shih A. L., Chung W. W., Yi H. C., China H. L., Min H. Y., Yi C. L., Ming H. H., Ching C. W. (2012). Servo motor Controller design for robotic Manipulator. International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2012), IEEE
- [19] Stanley J. B., James D. I. (2006). Aquaculture Feed Buoy Control-Part 1: System, IEEE
- [20] Sutam, B., Maungmeesri, B., & Maneetham, D. (2021). A smart pet monitoring and feeding based on feedback control system. *International Journal of Engineering Trends and Technology*, 69(4), 10-15.
- [21] Babu, B. R., Kumar, P. P., & Kuppusamy, P. G. (2019). Arduino Mega based PET feeding automation. *IOSR Journal of Electronics and Communication Engineering*, 14(4), 13-16.
- [22] Vrishanka, P. N., Prabhakar, P., Shet, D., & Rupali, K. (2021, December). Automated Pet Feeder using IoT. In 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC) (pp. 1-5). IEEE. 41

- [23] Nogueira, R., Araújo, H., & Prata, D. (2019). Robot chow: Automatic animal feeding with intelligent interface to monitor pets. *International Journal of Advanced Engineering Research and Science*, 6(4), 262-267.
- [24] Donde, O. O. (2017). Wastewater management techniques: a review of advancement on the appropriate wastewater treatment principles for sustainability. *Environmental Management and Sustainable Development*, 6(1), 40-58.
- [25] Imomqulov, U. B., Imomov, M. H., & Akbaraliyev, X. X. (2020). Theoretical Justification of Some Parameters of the Metering Device. *International Journal of Advanced Research in Science, Engineering and Technology*, 7(11), 15879-15884.
- [26] Bano, N., Amin, A., Boghani, H., Tariq, H., Bakhtawar, S., Waggan, I., & Younas, T. (2019, December). Radio controlled beach cleaning bot. In *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)* (pp. 1-6). IEEE.

APPENDIX

RE-2022-223858-plag-report

ORIGINALITY REPORT

14%

SIMILARITY INDEX

10%

INTERNET SOURCES

6%

PUBLICATIONS

10%

STUDENT PAPERS

PRIMARY SOURCES

1

sist.sathyabama.ac.in

Internet Source

3%

2

Submitted to University of Malaya

Student Paper

1%

3

Alexis Anne A. Luayon, Gio Francis Z. Tolentino, Van Keith B. Almazan, Patrick Eugene S. Pascual, Mary Jane C. Samonte. "PetCare", Proceedings of the 10th International Conference on E-Education, E-Business, E-Management and E-Learning - IC4E '19, 2019

Publication

1%

4

C. Jeeva, Giraja Shankar Chaurasia, Marshal A, Raghavendra R P, Sabarikrishnan S, Dharani Kumar P. "Digital Holographic Based Food Supply Post Flood Assessment System", 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2022

Publication

1%

www.ijsr.net

Submission Notification to co-author | BMS-ENG-2024-HT56-5178-7

Inbox xUnsubscribe x

Recent Patents on Engineering <admin@bentham.manuscriptpoint.com>

to me

15:05 (2 minutes ago) ☆ 😊 ↶ ⋮

Dear Dr. Aravind Kumar S S,

This is with reference to an article entitled: "Patented Automated Feeder Robot: Integrating Food and Water Dispensing, Camera System, and Wheeled Mobility" which has been submitted for possible publication in Recent Patents on Engineering, and in which you are listed as one of the co-authors.

The manuscript has been submitted to the journal by Anish M who is listed as the main author and who will be authorized to track the status of the paper after login.

If you have any objections to this submission, then please contact the editorial office as soon as possible by replying to this email. If we do not hear back from you within one week, we will assume you agree with your co-authorship.

Thank you very much.

Editorial Office
Recent Patents on Engineering
Bentham Science Publishers

MPS

Powered by Eureka

Aravind Kumar
Online

User Portal

Dashboard

New Submission

Email Notifications

Support

User Manuals

Welcome to your Dashboard

Author / Dashboard

ASSOCIATE AUTHOR SUBMISSIONS

Journal	Reference No.	Date	Title	Status
No Submission				

NEWLY SUBMITTED ARTICLES

Show 5 entries

Search:

Journal	Reference No.	Date	Title	Status
ENG	BMS-ENG-2024-HT56-5178-7	2024-03-26	Patented Automated Feeder Robot: Integrating Food and Water Dispensing, Camera System, and Wheeled Mobility	Awaiting Editorial Approval

Showing 1 to 1 of 1 entries

Previous1Next