IOT BASED PET CARE ROBOT

MHB4244 DESIGN PROJECT II REPORT

Submitted by

K .MAHIDHAR (21130036)
G.ANIRUDH (21130030)
V.SRI NAVEEN (21130038)
V.KUSHIN (21130043)

Under the guidance of

Dr. S. JANAKI RAMAN

Mr. V. RAM KUMAR

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

MECHATRONICS

DEPARTMENT OF MECHATRONICS ENGINEERING

SCHOOL OF MECHANICAL SCIENCES

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

PADUR - 603 103

MAY 2023

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

BONAFIDE CERTIFICATE

Certified that this project report titled **IOT BASED PET CARE ROBOT** is the Bonafide work of **Mr. K. MAHIDHAR (21130036), Mr. G. ANIRUDH (21130030) Mr. V.SRI NAVEEN (21130038) AND Mr. V. KUSHIN (21130043)** who carried out the project work under my supervision during the academic year 2022-2023.

HEAD OF THE DEPARTMENT

Dr. RM KUPPAN CHETTY

Professor and HOD(i/c)

Department of Mechatronics Engineering

Hindustan Institute of Technology and Science

Padur, Chennai-603103

SUPERVISORS

Dr. S. JANAKI RAMAN

Assistant Professor(S.G)

Mr. V. RAM KUMAR

Assistant Professor

Department of Mechatronics Engineering

Hindustan Institute of Technology and Science

Padur, Chennai-603103

INTERNAL EXAMINER

Name: _____

Designation: _____

EXTERNAL EXAMINER

Name: _____

Designation: _____

Project Viva-voce conducted on _____

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for his presence and immense blessings throughout the project work. It's a matter of pride and privilege for me to express my deep gratitude to the management of HITS for providing me the necessary facilities and support.

We are highly elated in expressing my sincere and abundant respect to the Vice-Chancellor **Dr. S. N. Sridhara**, for giving me this opportunity to bring out and implement my ideas in this project.

We wish to express my heartfelt gratitude to **Dr. RM. Kuppan Chetty**, Professor and HoD (i/c), **Department of Mechatronics Engineering** for much of his valuable support and encouragement in carrying out this work.

We would like to thank my guides, **Dr. S. JANAKI RAMAN**, Associate Professor(S.G), **Mr. V. RAM KUMAR**, Assistant professor, **Department of Mechatronics Engineering** for continually guiding and mentoring us by giving valuable suggestions to complete the project work.

We are exceptionally obligated to our project coordinators **Mr. V. Ramkumar** and **Ms. N. Seenu** for their direction and steady supervision throughout our project. We would like to thank all the technical and teaching staff, **Department of Mechatronics Engineering** for all the support.

Last, but not least, my thanks and thanks additionally go to my partners in building up the venture and individuals who have energetically gotten me out with their capacities. We are deeply indebted to my parents who have been the greatest support throughout the project.

K .MAHIDHAR (21130036)
G.ANIRUDH (21130030)
V. SRI NAVEEN (21130038)
V. KUSHIN (21130043)

ABSTRACT

A pet care robot based on the Internet of Things (IoT) that uses a variety of sensors and actuators to remotely monitor and care for pets is presented in this paper. The robot is made to give pet owners an automated way to make sure their pets are healthy, safe, and comfortable. A mobile robot with a camera, microphone, temperature sensor, and motion sensor makes up the system. This allows the robot to interact with pets and collect data in real-time. After that, the collected data are sent to, where they are analyzed and processed to produce useful insights and alerts. Additionally, the pet care robot has a feeding system that distributes food at predetermined intervals to ensure that the pet receives adequate nutrition even when the owner is not present. The proposed system is affordable, simple to use, and adaptable to various pet requirements. The aftereffects of the review exhibit the viability of the IoT-based pet consideration robot in observing and dealing with pets, in this way upgrading the prosperity of pets and diminishing the responsibility of animal people.

Keywords:

- IOT Based Robot
- Pan-tilt assembly
- Dispenser system
- Mobility of the robot
- Web page controlling

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction:

In the field of pet care, the application of robotics and the Internet of Things (IoT) technology has been steadily gaining traction. The need for automated solutions to ensure the health, safety, and comfort of pets while their owners are away has grown in importance as the number of pet owners grows. A promising solution has emerged in the form of IoT-based pet care systems that make use of sensors, actuators, and cloud computing to remotely monitor and interact with pets. The IoT-based pet care robot, a cutting-edge technology that provides pet owners with an automated pet care solution, is the subject of our in-depth investigation in this paper. The pet care robot can interact with pets and collect data in real-time thanks to a variety of sensors and actuators. After that, the data are sent to a platform in the cloud for analysis and processing, which results in useful insights and alerts for pet care. The pet consideration robot likewise includes mechanized taking care of framework that apportions food at explicit stretches, guaranteeing that pets are very much taken care of in any event when proprietors are away. This paper highlights the advantages of the IoT-based pet care robot, including enhanced interaction between pet owners and improved pet health. Overall, the IoT-based pet care robot has the potential to change the pet care industry by offering pet owners a low-cost, user-friendly, and customizable way to ensure their pets' well-being. Customers are interested in automatic devices for the purpose of ease of use and time savings, and businesses are attempting to meet the demand. As a result, the automation industry is becoming stronger and more developed every day. Automation is a method of controlling and operating procedures in an automatic manner with the assistance of electronics and software that will be programmed and implemented with machine learning technologies. Automation is not very new; it has been present in the market since the introduction of the first ATM machine in the 1960s. With the assistance of such machinery, the process became People love their pets, but there are times when you want to leave them alone for long periods of time, and this is often a problem. A doggy daycare robot that keeps an eye on and

feeds the dogs or cats in an extremely timely manner is what we propose here. Our project is an Internet of Things-based automatic pet feeding and monitoring system. The reason this was chosen as the title is to initially give up on finding a solution to a problem that many of us who keep pets face. It is difficult for humans to interfere with taking care of pets when they are busy. As a result, our system is effective enough to overcome the challenges that humans face when taking care of pets. This pet care system could be a complete apparatus for controlling the pet's freedom and monitoring all activities, especially those of a dog. Moreover, the venture is partitioned into a few modules every one of which has their special component. The system can feed the animals, keep track of their movements, and inform the owner of any changes. Many people find it difficult to feed their pets in a responsible and intelligent way.

the matter ends up being particularly clear when the proprietors have an intensely involved individual life. Owners will fill the feeder before leaving if they don't have time to feed them on time. His or her pets will almost always develop pathological conditions as a result of their unhealthy diet. According to recent research, obesity, and overeating are among the top health issues. Younger pets may continue to eat until there is nothing left to eat. Indeed, even grown-up pets can have an indistinguishable propensity, which causes a way more limited life expectancy for the pets. Our project is an Internet of Things-based automatic pet feeding and monitoring system. It is difficult for humans to interfere with taking care of pets when they are busy.

1.2 Objective of The Project:

An IoT-based pet care robot is intended to offer pet owners an automated means of ensuring their animals' health, safety, and comfort while they are away. The pet care robot monitors and interacts with pets remotely using a variety of sensors and actuators ,collecting real-time data on their behavior, health, and environment. The data is then analyzed and processed to produce useful insights and alerts that enable pet owners to take the necessary precautions for their animals' well-being. Moreover, the pet consideration robot includes computerized taking care of framework, which apportions food at explicit spans, guaranteeing that pets are very much taken care of in any event when proprietors are away. Overall, the IoT-based pet care robot aims to improve the quality of life for both pets and their owners by offering a low-cost, user-friendly, and customizable pet care solution.

CHAPTER 2

LITERATURE REVIEW

The design of an Internet of Things (IoT) that can improve pet care and monitoring is the goal of this study. Keeping pets has turned into a well known among individuals at the present and quite possibly of the most ideal buddy that that anyone could hope to find at home. The majority of pet owners struggle due to their busy schedules because taking care of their animals requires effort and time. This study will be able to feed and monitor the pets when the owner is away using a microcomputer and the internet. We modify a food dispenser prototype with a webcam, raspberry pi, and servo motor. A web application is created to control the servo engine and webcam by conveying message to the raspberry pi. To close, from the exploration we figured out 79% respondents unequivocally concur that pet taking care of framework are helpful and 61% need to purchase the framework. We hope that the feeding system can one day be used with animals in the zoo, pet store, wildlife department, etc. in the future [1].

A product called an automated pet feeder would take the place of the manual feeding method and could be set to feed at a predetermined rate and time. Development of pet feeder is utilized to defeat careless of the pet person to take care of their pet and to abstain from additional spending costs by leaving them at a pet inn. The creation of a pet feeder mechanism is the goal of this project. In order to control the mechanism and operate the system within a predetermined amount of time in this project, an Arduino is used as the microcontroller. The maximum stress that a designed mechanism can withstand is examined using finite element analysis. In addition, experimental and theoretical prototype testing of the mechanism was carried out, and the analysis's outcome is analyzed. The analysis is showing the way to ensure that the pet feeder's final concept is realized[2].

Customers are interested in automatic devices for the purpose of ease of use and time savings, and businesses are attempting to meet the demand. As a result, the automation industry is becoming stronger and more developed every day. Automation is a method of controlling and

operating procedures in an automatic manner with the assistance of electronics and software that will be programmed and implemented with machine learning technologies. Automation is not very new; it has been present in the market since the introduction of the first ATM machine in the 1960s. With the assistance of such machinery, the process became Individuals love their pets furthermore, contrariwise, yet there are times you wish to leave your pets gathering for long spans alone and this is frequently a risky issue. A doggy daycare robot that keeps an eye on and feeds the dogs or cats in an extremely timely manner is what we propose here. Our project is an Internet of Things-based automatic pet feeding and monitoring system. The weight of picking this is in light of the fact that the title is on the grounds that, at first to give up an arrangement to an issue looked at by a lot of people of us who keep pets. It is difficult for humans to interfere with taking care of pets when they are busy. As a result, our system is effective enough to overcome the challenges that humans face when taking care of pets [3].

A low-cost, wearable, and unobtrusive intelligent accelerometer sensor for observing human activity is the subject of this paper's hardware and software design and implementation. To elevate solid ways of life for seniors for a functioning, free, and sound maturing, as well as concerning the early identification of psychomotor irregularities, the movement observing is acted in an all-encompassing way in similar gadget through various methodologies: 1) a categorization of the level of activity that enables the formation of behavior patterns; 2) a classifier for daily living activities that uses a straightforward method to decouple the motion and gravitational acceleration components to distinguish between activities like climbing or descending stairs; and 3) an estimation of the user's metabolic expenditure that is unaffected by the activity they engage in or their anthropometric characteristics. The prototype and proposed algorithms' viability have been demonstrated by the results of experiments[4].

Time synchronization methods for low-power sensor modules are in high demand due to the rise of IoT-based home automation systems. The obliged application convention (CoAP) was as of late normalized for sensor networks by IETF and is turning out to be generally taken on

for home computerization frameworks by ETSI, OMA, and oneM2M. The organization time convention (NTP) isn't relevant to home robotization frameworks because of its restricted registering assets. A light time synchronization algorithm for CoAP-based home automation system networks is proposed in this paper. The CoAP choice field and a shim header are utilized to incorporate time-stamps in the home robotization framework. As a result, both IP-based and non-IP-based home automation systems can benefit from the proposed plan. The proposed method outperforms the ideal NTP service with an average error of 1 millisecond and a network overhead reduction of 17% when tested on a variety of household devices with non-IP communication interfaces [5].

The profoundly divided and non-normalized scene of the Web of Things industry brings about compelling both IoT designers and end-clients to need to pick their restrictive buyer hardware by an organization, at last turning into a hindrance to fabricating an unfragmented IoT environment. This paper proposes an oneM2M norms agreeable gadget programming stage for customer hardware in light of the Web of Things, called &Cube. It works with oneM2M service platforms through a standardized resource model and REST APIs, resulting in interoperability with various IoT consumer electronics built on the &Cube. Manufacturers and developers will be able to develop novel products and entirely new services thanks to the growing use of the &Cube in consumer electronics[6].

The number of consumer appliances and devices used in the home has increased as a result of recent advancements in smart home systems. In any case, large numbers of these gadgets and apparatuses show specific level of heterogeneity and don't adjust towards joint execution of activity. As a result, it's hard to work together, especially to deliver the services that home users want. For the execution and federated coordination of smart home systems, we propose a brand-new intelligent interoperability framework in this paper. The Simple Object Access Protocol (SOAP) platform-independent interoperability of heterogeneous systems is the foundation of the framework's core. We have tested the interoperability framework with a number of home devices to show that they work well together. The framework's performance was

evaluated in a Local Area Network (LAN) setting, and it was found to be reliable in smart home setting [7].

The need for the pet feeding robot is great because keeping pets takes a lot of time, and we want to make it easier and more efficient for users to feed their pets from anywhere in the world. Keeping pets takes a lot of time and effort. This includes keeping them company, expressing your concerns, and, of course, providing them with food on time and correctly. But not everyone is an expert on pets; It can be difficult and time-consuming to take care of your pet's diet. Overeating and obesity are among the most common health issues that pets face. They typically are content with whatever they are given, especially when they are younger. Numerous grown-up pets are taken care of informally that later may cause short life expectancy. Feeding pets can be difficult when people travel for work or pleasure, so it's possible that they won't always be home. Being preoccupied with personal plans while also caring for a hungry child at home is a constant source of concern for users. The third worry that we need to manage is the way that there hasn't been any item in the market right now that can apportion nourishment for pets checked by its client in realtime. Nonetheless, pets themselves could not be guaranteed to perceive the potential medical issues of eating the wrong food. Pet net, Auto Pet Feeder, and Automation Pet Feeder are examples of products that can be programmed to dispense food at predetermined intervals but lack real-time monitoring and mobility. As a result, we intend to address the issue of feeding by developing a phone- or laptop-controlled, real-time, semiautomatic pet feeder that can dispense the food that the user specifies based on live camera feedback[8].

A low-cost, wearable, and unobtrusive intelligent accelerometer sensor for observing human activity is the subject of this paper's hardware and software design and implementation. To elevate solid ways of life to seniors for a functioning, free, and sound maturing, as well concerning the early identification of psychomotor irregularities, the movement observing is acted in an all encompassing way in similar gadget through various methodologies: 1) a categorization of the level of activity that enables the formation of behavior patterns; 2) a classifier for

daily living activities that uses a straightforward method to decouple the motion and gravitational acceleration components to distinguish between activities like climbing or descending stairs; and 3) an estimation of the user's metabolic expenditure that is unaffected by the activity they engage in or their anthropometric characteristics. The prototype and proposed algorithms' viability have been demonstrated by the results of experiments[9].

There is a growing body of scientific evidence that nonhuman animals can feel emotions. Owners of companion animals readily attribute emotions to their animals and demonstrate a strong connection and attachment to them. The relationship between companion animal owners' attribution of emotions to their companion cat or dog and their attribution of mirrored emotions is examined in this paper. The consequences of a web-based survey, finished by 1,023 Dutch-talking feline or potentially canine proprietors (principally in the Netherlands and Belgium), recommend that proprietors quality a few feelings to their pets. All posited basic (anger, joy [happiness], fear, surprise, disgust, and sadness) and complex (shame, jealousy, disappointment, and compassion) emotions were attributed to their companion animals by respondents, with a general trend toward basic emotions being attributed more frequently than complex emotions (with the exception of sadness). All pet owners demonstrated a strong bond with their companion animal(s), with cat and dog owners' levels of bonding significantly influenced by gender and education level. On the Pet Bonding Scale (PBS), owners who attributed human characteristics to their dog or cat also scored higher. Lastly, the average PBS score of pet owners was higher than that of those who did not like to pet their pets. All of the emotions, including joy, sadness, surprise, shame, disappointment, and compassion, were significantly correlated with owners' attributions of mirrored emotions and their level of attachment to dogs[10].

Pet ownership is on the rise right now. The animal may indeed be considered a friend, companion, or even family member by its owner. The prevalence of pets in the web-based entertainment persuades individuals to claim pets permitting them to demonstrate the way that well they can connect with pets. Possessing a pet has demonstrated to be useful to both physical

and psychological wellness of the proprietor. However, the pet is neglected or not given the proper care because of the owner's busy schedule and unpredictable activities. Robotic pets have been developed by a number of businesses in response to the reduced risk of owning a pet. Even though these robotic pets require minimal care, they still serve the same purpose as real pets. There are a few robotic pets on the market, but they are too expensive and only serve as companions or a few other functions. The creators have recently proposed the primary form of a robot with adjustable capabilities (CoFiBot) permitting the client to select two modes: pet and fighting fires. The development of the second version (V2) of CoFiBot was motivated by the requirement for such additional functions and the recent high demand for the integration of IoT technology. CoFiBot V2 features customizable home monitoring functions and fire detection capabilities. It can simultaneously roam freely within the owner's property and provide the owner with online information about its surroundings. CoFiBot V2, a mammal-like robotic pet based on the Internet of Things, has been shown to be visually appealing and to perform well in functional and endurance tests. A correlation result between CoFiBot V2 and other exploration level or market-prepared mechanical pets with regards to cost and usefulness demonstrates its appeal [11].

The review introduced in this paper investigated individuals' discernments and perspectives towards the possibility of a future robot ally for the home. Surveys and trials of human-robot interaction were used to collect data from 28 adults in a human-centered manner. According to the findings, the majority of participants were in favor of having a companion robot and perceived the potential role as that of an assistant, machine, or servant. Hardly any maintained that a robot buddy should be a companion. Child and animal care tasks were favored over household chores. Humanlike correspondence was alluring for a robot sidekick, though humanlike way of behaving and appearance were less fundamental. The implications of the findings for the direction of future research into the creation of robot companions are discussed[12].

This review inspected preschool kids' thinking about and social connections with quite possibly of the most exceptional mechanical pet as of now on the retail market, Sony's automated canine AIBO. Eighty kids, similarly split between two age gatherings, 34-50 months and 58-74 months, partook in individual meetings that included play with and a meeting around two curios: A stuffed dog and AIBO. Children's reasoning about the two artifacts was similar, but their behavioral interactions were different, according to the findings. Conversation centers around how mechanical pets, as illustrative of an arising mechanical sort in HCI, might be (a) obscuring fundamental ontological classes, and (b) affecting youngsters' social and moral turn of events. In a broader sense, the findings contribute to our comprehension of the human-robot relationship[13].

Pet ownership is on the rise right now. The animal may indeed be considered a friend, companion, or even family member by its owner. The prevalence of pets in the web-based entertainment persuades individuals to claim pets permitting them to demonstrate the way that well they can connect with pets. Possessing a pet has demonstrated to be useful to both physical and psychological wellness of the proprietor. However, the pet is neglected or not given the proper care because of the owner's busy schedule and unpredictable activities. Robotic pets have been developed by a number of businesses in response to the reduced risk of owning a pet. Even though these robotic pets require minimal care, they still serve the same purpose as real pets. There are a few robotic pets on the market, but they are too expensive and only serve as companions or a few other functions. The creators have recently proposed the primary form of a robot with adjustable capabilities (CoFiBot) permitting the client to select two modes: pet and fighting fires. The development of the second version (V2) of CoFiBot was motivated by the requirement for such additional functions and the recent high demand for the integration of IoT technology. CoFiBot V2 features customizable home monitoring functions and fire detection capabilities. It can simultaneously roam freely within the owner's property and provide the owner with online information about its surroundings. CoFiBot V2, a mammal-like robotic pet based on the Internet of Things, has been shown to be visually appealing and to perform well in functional and endurance tests. A correlation result between CoFiBot V2 and

other exploration level or market-prepared mechanical pets with regards to cost and usefulness demonstrates its appeal[14].

It is now widely accepted that having a pet improves one's health, sense of psychological well-being, and longevity due to extensive media coverage. However, while some researchers have found that interacting with animals has positive effects, others have found that pet owners' health and happiness are no better or worse than those of non-pet owners. I argue that the existence of a generalized "pet effect" on human mental and physical health is currently not a fact but an unsubstantiated hypothesis, and I discuss some of the reasons why studies of the effects of pets on people have produced conflicting results[15].

A quickly expanding homeless creature populace in Bangkok has caused concern with respect to transmission of vector-borne and zoonotic sicknesses. The goal of this study was to find out if stray animals in Bangkok could be a source of Hepatozoon, a genus of parasites transmitted by ticks that has received little attention in Thailand. In 42 Bangkok metropolitan districts, stray companion animals near monasteries provided blood samples. There were samples taken from 26 districts for both dogs and cats—four districts for dogs alone and twelve for cats alone. There were a total of 308 dogs and 300 cats from which samples were taken. These samples were examined for evidence of Hepatozoon infection with light microscopy and an 18 S rRNA gene-based PCR assay. By microscopy, gamonts were found in the blood smears of 2.6% of dogs and 0.7% of cats. Hepatozoon was found in buffy coats of 11.4% of dogs and 32.3% of cats tested by PCR. The commonness of disease was similar among male and female canines or felines, and PCR-positive canines and felines were seen as in 36.6% and 36.8% of the locale overviewed, separately. There was a relationship between the rates of PCR-positive canines and felines in regions where both host species were tested. Groupings of delegate amplicons were nearest to those announced for H. canis. The first molecular proof that H. canis is native to Thailand is provided by these findings. It is necessary to investigate the role that stray cats play in the epizootiology of hepatozoonosis given the unexpectedly high prevalence of Hepatozoon among them[16].

In recent years, feline vector-borne diseases (FVBDs) have emerged, demonstrating a wider geographic spread and rising global prevalence. Notwithstanding their veterinary significance, homegrown felines assume a focal part in the transmission patterns of a few FVBD specialists by going about as repositories and sentinels, a situation that requires a One Wellbeing approach. The current study's objective was
to evaluate the risk factors associated with feline vector-borne bacteria and protozoa with veterinary and zoonotic significance in cats from southern Portugal[17].

Numerous businesses are being pressured to develop robotic pets as an alternative to living pets. Robotic pets don't need to be fed or played with. One of such robots, Sony's Aibo robot canine, was demonstrated to show pet qualities the manner in which the living canines do. In a review directed among preschool kids, Sony Aibo had the option to show pet-human associations with kids like those of living canines. Additionally, the children were able to form a human-like bond with the robot dog. The objective of the project described in this paper is to construct a quadrupedal dog-like robot that is both a companion for humans and a functioning robot. The current robot is the first version of the modular pet robot known as CoFiBot (customizable function robot). Aside from an engaging look and superior velocity, the CoFiBot V2 is outfitted with the usefulness and the effortlessness ordinarily found in bug-like robots. The firefighting feature that was present in the previous version is eliminated while the fire-detection procedure is maintained. This permits CoFiBot V2 to work as a playable pet when the proprietor is available and a portable home-checking framework when it is let be. Finally, a comparison is made between CoFiBot V2 and other robotic pets that have been developed. This is finished to show that the previous starts to lead the pack in term of cost to usefulness proportion[18].

The population is rapidly aging as a result of the explosive combination of a low birth rate and a long life expectancy. The expenses of medical services in the dark society are expanding decisively, and soon there will be insufficient assets and individuals for care. In this setting, innovative concepts for elderly care solutions that gradually reduce the proportion of human-

based care are required. These requirements are the focus of research on robot-based solutions for active aging and elderly care. According to an overall point of view, mechanical technology has the ability to reshape the scene of medical services both in its construction and its activity totally. In point of fact, digital health technologies like artificial intelligence, 3D printing, and robotics could be used to power automation that would help ensure the sustainability of healthcare systems over the long term. The last option could take over dreary work from medical services laborers, which would permit them to zero in additional on patients and to have lesser responsibility. Robots may be utilized in senior consideration with a few distinct points. (i) Robots have the potential to act as caregivers, which means that they can help the elderly, ii) they can give restraints and instructions for activities of daily life and safety, and/or they can help their caregivers with everyday tasks; iii) they can assist with observing their way of behaving and wellbeing; also (iv) give friendship, including diversion and side interests, memory and social contact. The utilization of Robots with human subjects/patients bring up a few delicate issues. First and foremost, robots have the potential to serve as information hubs and collect a staggering amount of information about their subjects and surroundings. Truth be told, they record propensities, for example, dozing, working out, third people going into in the house, arrangements. It is possible to continuously record communications. They can also store medical data by connecting to medical devices. On the one hand, this is a very powerful tool for collecting information about a single subject (precision medicine), about a disease (so that artificial intelligence can eventually find new signs and symptoms through machine learning and deep learning), and about the person's environment. On the other hand, the subjects' and their caregivers' privacy may be significantly enhanced by this potent instrument. As a result, robotics is a complex area of ethics. Care robotics run the risk of reducing human contact, increasing the objectification and loss of control of the elderly, and jeopardizing the individual's privacy and freedom (particularly when robots may perform restrictive interventions). Additionally, the use of robots in the care of the elderly may increase the likelihood that the elderly person will be deceived and treated as a child[19].

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter explains about the step-by-step process starting from design to Prototype of the project.

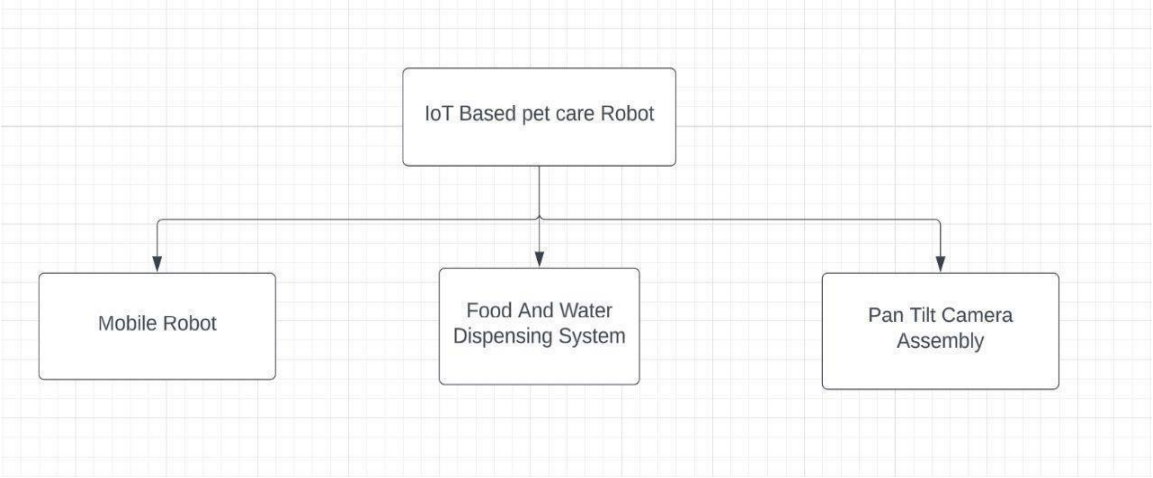

Fig 3.1 methodology flow chart of the iot based pet care robot

3.2 Chassis Design In Solid Works:

A set of designs were designed for the chassis and at last a design was finalized. Then the chassis of the rover was designed as separate pieces and fixed together in the solid works software and the material properties were added to it.

3.3 Circuit Designing:

The circuit designing was done in proteus the circuit consists of an ARDUINO UNO which act as a controller. 4 DC motors [200 R.P.M] are used to run the motor , L298N driver is used to control the motors, 5v buck converters are used to transfer the 12v supply to 5v, ESP cam 32 is used to steer the robot, SD90 servo motor is used to tilt the camera position, we installed the food dispenser system were it contains propeller, doppler these are useful to dispense the food for the dog, we installed the water pump motor to dispense the water.

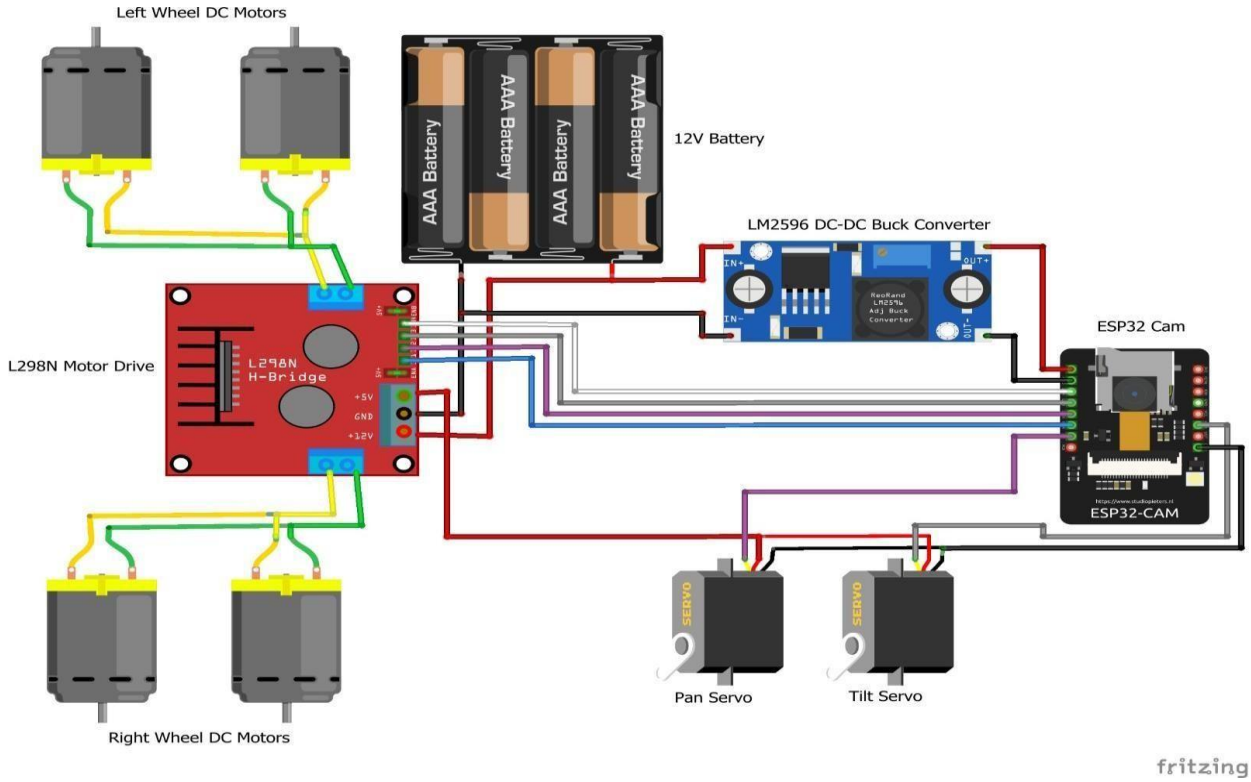

Fig 3.2 Circuit design for the robot movement and pan tilt servo.

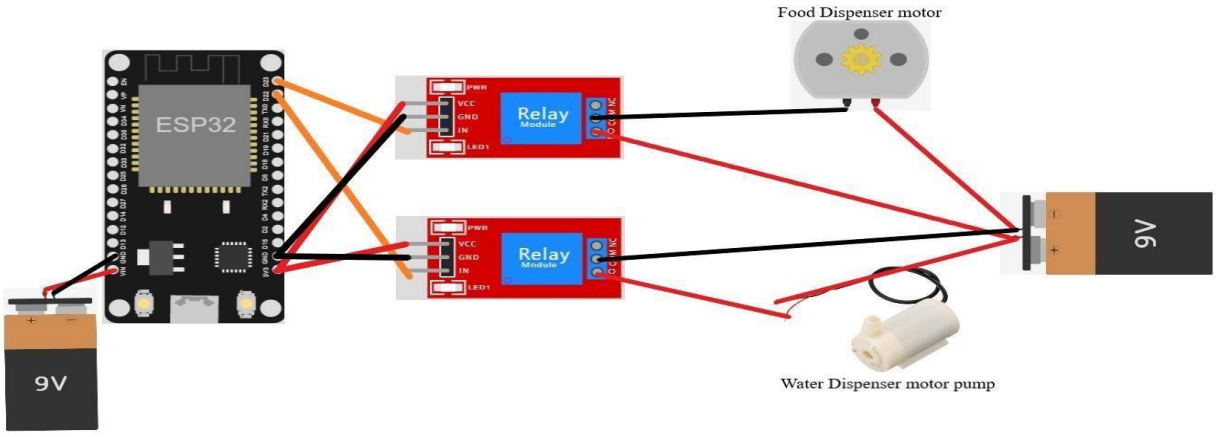

Fig 3.3 Circuit design of the dispenser system

3.4 Prototype :

Acrylic sheet was used for the frame structure because of its availability in the market and because it is cost-efficient. The frame was laser cutted together according to the dimensions done in solid works. Motor housing was done with the help of basic mechanical machinery such as a lathe and drilling machine. The center was taken and the hole was done with a depth of 5mm and diameter of 32mm, drilling was done inside thcounter bore to fix and screw the motor in it. Banner glue was used on the top of the housing for attaching it to the acrylic sheet. For the food dispenser we have used the L-bend PVC pipe the diameter of the PVC pipe matched with the propeller diameter, the food container is kept above the L-bend PVC pipe. A dropper is kept inside the food container which directs the food to the propeller. The Propeller is attached to 45 RPM geared motor which turns the propeller dispense the food. For the water dispenser small water pump motor is kept inside the water container to dispense water.

For pan-tilt camera assembly servo case and esp32 camera case had been 3D printed and kept on the upper plate of the robot.

CHAPTER:4
HARDWARE COMPONENTS

4.1 ESP32 CAM MODULE:

The ESP32-CAM module is a little estimated, minimal expense camera module that incorporates a Wi-Fi and Bluetooth-empowered ESP32 microcontroller and an OV2640 camera sensor. It is ideal for surveillance systems, video streaming, and Internet of Things projects because it offers a low-power, low-latency method for streaming high-quality video and taking pictures. The ESP32-CAM module can record video at up to 60 frames per second (fps) and has a maximum resolution of 2 megapixels (1600 x 1200). Additionally, it supports JPEG compression for effective image transmission and storage. The module supports external SD card storage for extended recording times and has a built-in flash memory of 4 MB. The ESP32-CAM module can be customized utilizing the Arduino IDE, making it simple for fledglings to get everything rolling with. It also works with MicroPython and Lua, two programming languages. The module can speak with different gadgets utilizing Wi-Fi and Bluetooth, and it upholds well known IoT conventions like MQTT and HTTP. Additionally, the ESP32-CAM module comes equipped with a number of peripherals, such as GPIO, I2C, SPI, and UART interfaces, that can be utilized for the connection of sensors, displays, and other devices. It also supports audio streaming and recording and has a microphone built in refer (4.1)

Fig 4.1 ESP32-CAM module

4.2 ESP32 MODULE:

Espressif Systems, the same company that created the well-known ESP8266 SoC, offers the inexpensive ESP32 System on Chip (SoC) Microcontroller. The 32-bit Xtensa LX6 Microprocessor by Tensilica is a replacement for the ESP8266 SoC and features built-in Wi-Fi and Bluetooth. It is available in single-core and dual-core versions. The advantage of ESP32 is that it has inbuilt RF components such a power amplifier, a low-noise receiver amplifier, an antenna switch, filters, and an RF balun, similar to ESP8266. As a result, it is very simple to construct hardware around the ESP32 since minimal external components are needed. The fact that ESP32 is produced utilizing TSMC's ultra-low-power 40 nm technology is another crucial information to be aware of. Therefore, employing ESP32 should make it very simple to create battery-powered applications like as wearable's, audio equipment, baby monitors, smart watches, etc. refer (fig 4.2)

Fig 4.2 ESP32 module

4.3 Servomotor:

The SG90 servo engine is a little, lightweight engine that is regularly utilized in leisure activity and instructive undertakings. A feedback mechanism is used to control its position, speed, and acceleration in this type of DC motor. Due to its low price, high torque, and small size, the SG90 servo motor is a popular choice. The SG90 servo motor can be controlled with a pulse width modulation (PWM) signal, has a rotational range of 0 to 180 degrees, and operates on a voltage range of 4.8V to 6V. It can produce a torque of up to 1.8 kg/cm at 4.8V. The PWM signal decides the place of the engine by controlling how much time the engine is turned on during each cycle. The SG90 servo motor is frequently utilized in RC planes, robotics, and other hobbies. It can be utilized to direct the motion of robot wheels, arms, and other mechanical components. In RC planes, it is also used to control the rudder, elevators, and ailerons. There are various versions of the SG90 servo motor with a variety of gears and bearings. Metal gears and bearings are included in some versions, which improve precision and durability. Different forms are furnished with plastic cog wheels and direction, which are more affordable yet may break down more rapidly refer fig (4.3)

Fig 4.3 SG90 servo motor

4.4 DIRECT CURRENT MOTOR :

DC engines are ordered in light of their speed, force, and power rating. The construction of a DC motor and the voltage that is applied to it determine its speed. The motor will rotate more quickly the higher the voltage applied to it. A DC motor's torque is the amount of rotational force it can produce and is determined by the motor's current. The maximum amount of electrical power that a DC motor can handle is known as its power rating, and it is typically expressed in watts. Batteries, power supplies, and motor controllers are some of the power sources that can be used to power a 200 RPM DC motor. It can be controlled by varying the voltage or current it receives or by controlling its speed with pulse width modulation (PWM) techniques. In projects related to robotics and automation, the 200 RPM DC motor is frequently used. It very well may be utilized to drive wheels, gears, and other mechanical parts, and it can likewise be utilized to control the development of automated arms and different members. It is additionally utilized in robots and quadcopters to control the speed and bearing of the propellers refer fig (4.4).

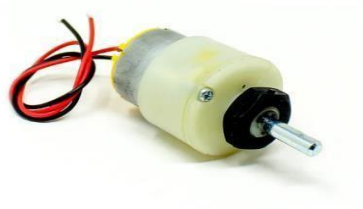

Fig 4.4 200 RPM DC motor

4.5 Lithium Ion Cells:

Lithium Ion batteries are noted for their performance, reliability, and affordability. Three 3.7v lithium ion cells is connected in series to acquire the 12v. The battery pack is 12v, 2500mah and 3c rated refer fig (4.5).

Fig 4.5 Lithium ion cells

4.6 Motor Driver:

A well-known IC for controlling the speed and direction of DC motors is the L298N motor driver. It is usually utilized in advanced mechanics and different applications where exact engine control is required. Two H-bridges are included in the L298N motor driver, and their purpose is to direct the flow of current to the motor. Each H-bridge has four transistors that can be independently turned on and off to control the motor's current flow. The L298N motor driver is suitable for controlling DC motors of a small to medium size because it can handle a maximum current of up to 2 amps per channel.The L298N motor driver's versatility is one of its advantages. It is capable of controlling a wide variety of DC motors, including those that require varying voltages and power. Moreover, it is moderately simple to utilize and can be incorporated into different circuits and frameworks refer fig (4.6)

Fig 4.6 L298N Motor Driver

4.7 Channel Relay Module:

An electronic device for controlling the power supply to electrical appliances is known as a two-channel relay system. A sort of switch can be worked electronically, and is usually utilized in home robotization frameworks, modern control frameworks, and different applications that require controller of electrical gadgets. An electromechanical switch and a control circuit make up a relay. The switch is activated by the control circuit, which in turn controls the electrical device's power supply. In a 2channel transfer framework, there are two free transfers that can be controlled independently. A microcontroller or other electronic device typically sends a digital signal to each channel in a two-channel relay system. The relay turns on the switch and lets power flow to the electrical device when it receives the digital signal. The two channels can be utilized to control different electrical gadgets or to give repetitive control to a solitary gadget. The control of lights, fans, and other electrical devices is one common use for a two-channel relay system in home automation systems.

A home automation controller, for instance, can be connected to a relay system and accessed remotely using a smartphone or other device. This makes it more convenient and secure for homeowners to turn on and off lights and other electrical appliances from anywhere in the world refer fig (4.7)

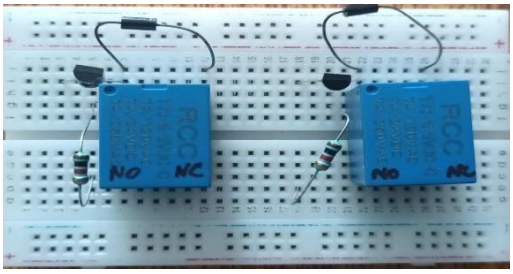

Fig 4.7 2 channel relay system

4.8 DC to DC Buck Convertor:

An electronic circuit known as a DC to DC buck converter transforms a DC signal of a higher voltage into a DC signal of a lower voltage. It is a type of DC-DC converter that controls the voltage output through switching techniques. A control circuit turns on and off the buck converter at a high frequency with a switch, typically a MOSFET. A capacitor is charged by current that flows through an inductor when the switch is turned on, resulting in a higher voltage. The inductor discharges and the voltage across the capacitor decreases to produce a lower voltage when the switch is turned off. The high efficiency of a buck converter is its primary advantage. The converter can function at high efficiency even when there is a significant voltage difference between the input and output because the inductor stores energy during the switch's on time and releases it during the switch's off time. Buck converters are frequently utilized in a wide variety of applications, such as voltage regulators for automotive applications, renewable energy systems, and power supplies for electronic devices. They are particularly valuable when the information voltage is excessively high for the objective gadget, as they permit the voltage to be diminished to the proper level with negligible energy misfortune refer fig (4.8).

Fig 4.8 DC to DC Buck Convertor

4.9 Water Pump Motor

An electronic device known as a 5V water pump uses a low voltage DC power supply to move or circulate water. It is frequently utilized in hydroponic systems, aquariums, and other small water features where a low flow rate is required. Most of the time, the 5V water pump has a small motor that moves an impeller or propeller to move water. A 5V DC power supply, typically found in the form of a USB port or a battery pack, powers the pump. The model and design of a 5V water pump can affect its flow rate. Depending on the pump, the flow rate can range from a few liters per hour to several hundred liters per hour. The low power consumption of a 5V water pump is one advantage. It uses less power than other types of water pumps because it operates at a lower voltage, making it more energy-efficient. Moreover, it is not difficult to introduce and work, and can be controlled utilizing a basic on/off switch refer fig (4.9)

Fig 4.9 Water pump

CHAPTER: 5

SOFTWARE USED

5.1 SOLID WORKS

SolidWorks is a 3D CAD (computer-aided design) software used in product design, architecture, engineering, and other fields. Dassault Systèmes was the company that developed it, and it debuted in 1995. For designing 3D models of parts and assemblies, SolidWorks provides a user-friendly interface that enables users to create intricate geometries and carry out in-depth simulations.

The parametric modeling capabilities of SolidWorks make it possible for users to create parts and assemblies with defined dimensions and relationships between components. This empowers simple adjustment of plans as changes can be made to the elements of one part, and the whole model will refresh in like manner .

SolidWorks likewise incorporates a scope of devices for performing reproductions and investigation of plans, like pressure examination, movement reenactment, and liquid elements recreation. Before making physical prototypes, engineers can test their designs virtually and improve their performance, saving time and money.

One more significant element of SolidWorks is its capacity to produce 2D designing drawings from 3D models. Dimensions, tolerances, and other annotations can be added to these drawings to meet industry standards.

Using tools like version control and document management, SolidWorks also gives teams a place to work together on designs at the same time. Other software tools, such as CAM (computer-aided manufacturing) and PLM (product lifecycle management) systems, can also be integrated with it.

5.2 WEB-PAGE DEVELOPMENT USING HTML:

We designed a web page to control the robot with the help of our mobile and with the help of the WIFI we can control the robot by seeing through the camera which had been installed on the top of the robot[14], as we can control the locomotion of the robot and we can dispense the food with the help of google assistance also which had been created with the help of sinric pro software.

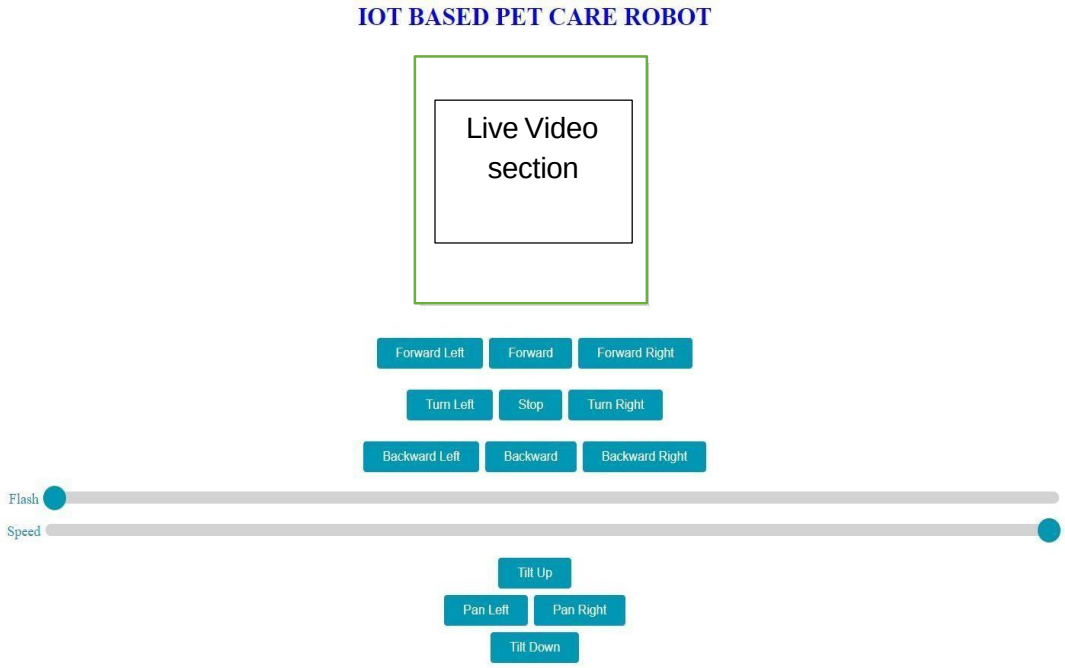

Fig 5.1 web page to control the robot

CHAPTER 6

CAD MODEL OF THE ROBOT

6.1 Chassis Design:

The images given below are the orthographic view of the chassis done in solid works.

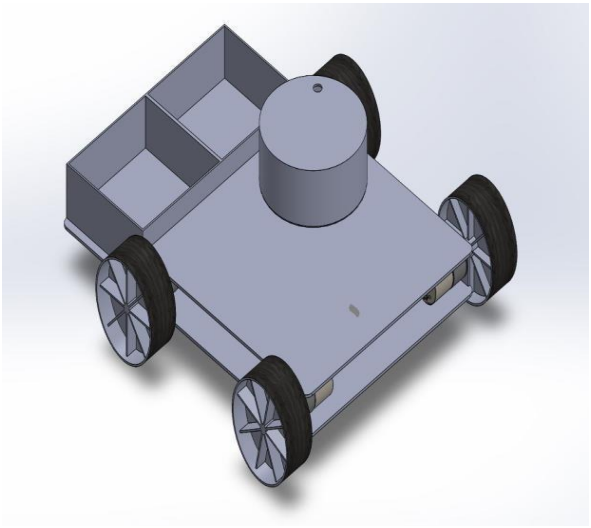

Fig 6.1 Isometric View

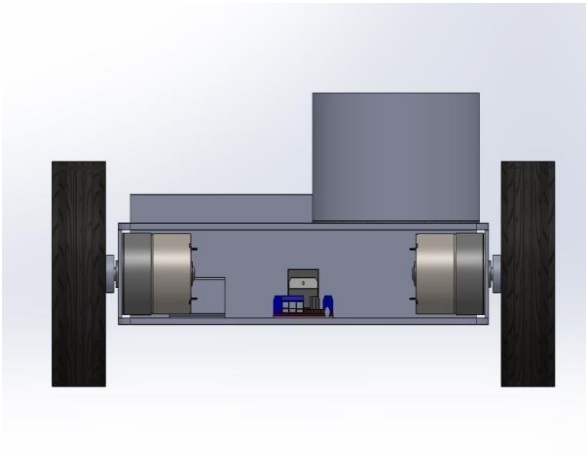

Fig 6.2 Side View

6.2 Assembled view of the robot:

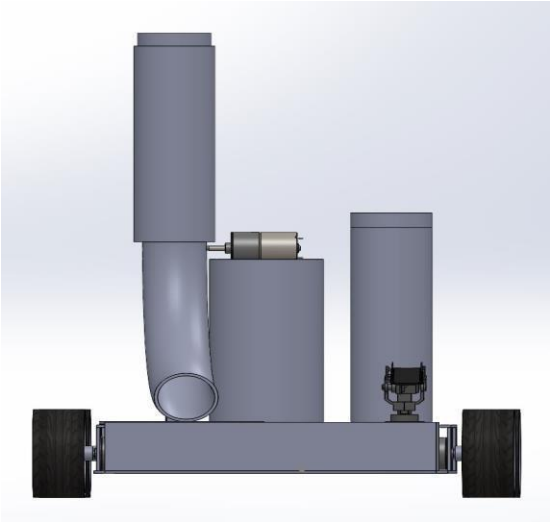

Fig 6.3 front view of the assembled part

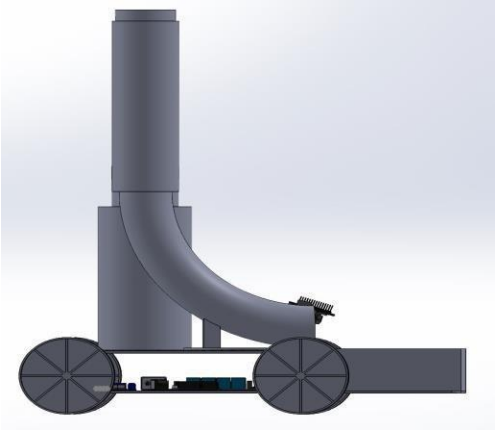

Fig 6.4 side view of the assembled part

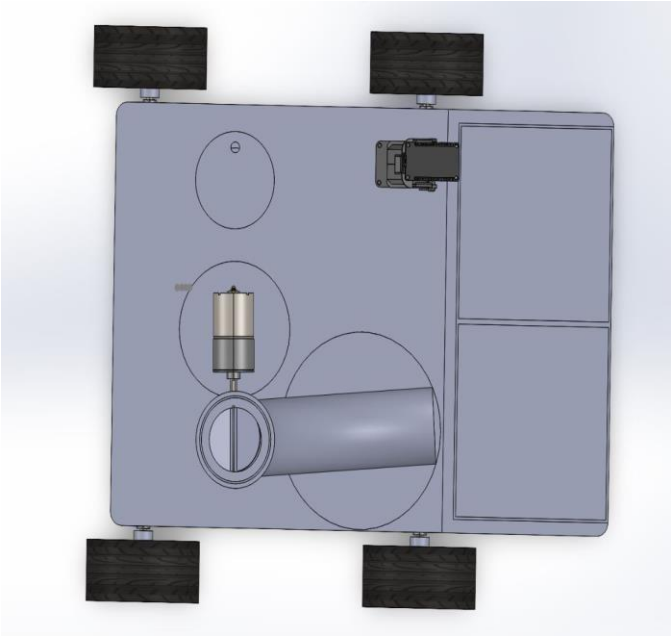

Fig 6.5 Top view of the assembled part

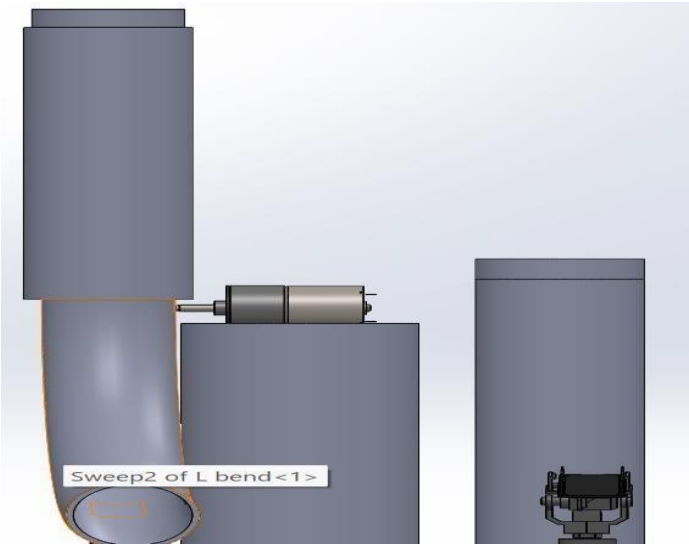

Fig 6.6 Assembled view of food dispensing system

CHAPTER 7
CONCLUSION

In conclusion, the pet care robot that is based on the Internet of Things is an innovative technology that has the potential to change the pet care industry. The pet care robot can provide personalized care and attention to pets while offering pet owners convenience and ease of use by incorporating IoT technology, sensors, and advanced algorithms. The pet care robot is a great option for busy pet owners who want to make sure their animals are taken care of even when they aren't home due to its food and water dispensing system and surveillance and monitoring features. The ESP32 CAM module, SG90 servo motor, 200 RPM DC motor, and Arduino Uno are used to construct a pet care robot that is both functional and dependable. It is likely that in the not-too-distant future, we will see even more advanced pet care robots that can perform a wider range of tasks and provide even more individualized pet care thanks to ongoing advancements in AI and IoT technology. Overall, the IoT-based pet care robot is a promising technology that has the potential to transform how we take care of our furry friends and make pet ownership more accessible and pleasurable for everyone. An IoT-based pet care robot that pet owners can remotely monitor and control is the goal of this project. The pet consideration robot will highlight different sensors and actuators that will empower constant information assortment on pet way of behaving, well-being, and climate, as well as a computerized taking care framework. The cloud-based stage will empower animal people to remotely screen and control the pet consideration robot, as well as get a caution and bits of knowledge on the pet's way of behaving and well-being. By offering pet owners a cost-effective, user-friendly, and adaptable pet care solution, this project has the potential to transform the pet care industry. The population is rapidly aging as a result of the explosive combination of a low birth rate and a long life expectancy. The expenses of medical services in the dark society are expanding decisively, and soon there will be insufficient assets and individuals for care. In this setting, innovative concepts for elderly care solutions that gradually reduce the proportion of human-based care are required. These requirements are the focus of research on robot-based solutions for active aging and elderly care. According to an overall point of view, mechanical technology has the ability to reshape the scene of medical services both in its construction and its activity totally. In point of

fact, digital health technologies like artificial intelligence, 3D printing, and robotics could be used to power automation that would help ensure the sustainability of healthcare systems over the long term. The last option could take over dreary work from medical services laborers, which would permit them to zero in additional on patients and to have lesser responsibility. Robots may be utilized in senior consideration with a few distinct points. On the other hand, the subjects' and their caregivers' privacy may be significantly enhanced by this potent instrument. As a result, robotics is a complex area of ethics. Care robotics run the risk of reducing human contact, increasing the objectification and loss of control of the elderly, and jeopardizing the individual's privacy and freedom (particularly when robots may perform restrictive interventions). Additionally, the use of robots in the care of the elderly may increase the likelihood that the elderly person will be deceived and treated as a child. Software and hardware can be embedded on the Internet of Things platform. From this, it is clear that IoT is a useful method for obtaining data. As previously mentioned, a web service-based SOAP-based mechanism is an excellent choice for managing a variety of home appliances and devices. A variety of sensors are utilized to monitor the pet's various activities, such as an IR sensor to determine whether food is present on the plate. The pet can share its identity thanks to an RFID tag in its collar. The information that has been collected is sent through Arduino's gateway to cloud storage, where it can be retrieved and accessed using a mobile phone number or any other electronic device. Since the entire network is wireless, loss is impossible. IEEE 802.15.4 and IEEE 802.11 are necessary for the realization of two smartphone-based SDR prototypes. For 802.11p, the base

required testing rate is 10MS/s, hence we use PBSK and QPSK each is 4bytes. The addition of RTC to the feeder can further enhance the work.

CHAPTER 8

RESULT AND DISCUSSION

In this chapter we are going to see how the dispenser system works.

A application called sinric pro is providing free IOT services. It is allowing to use upto three devices connected to the application. The application is connected to the Esp32 which is used for the food and water dispenser. Digital switches created with the names of the food and water dispenser and if it is clicked the app sends the signal to the esp32 and esp32 sends the signal to the relay channel to turn on and turn off. The sinric pro app can also be interfaced with the google home application so digital switches will also be displayed and controlled in the google home application also. As the Google home application can also be used to control the dispenser, now the dispenser can be controlled via the Google assistants via voice.

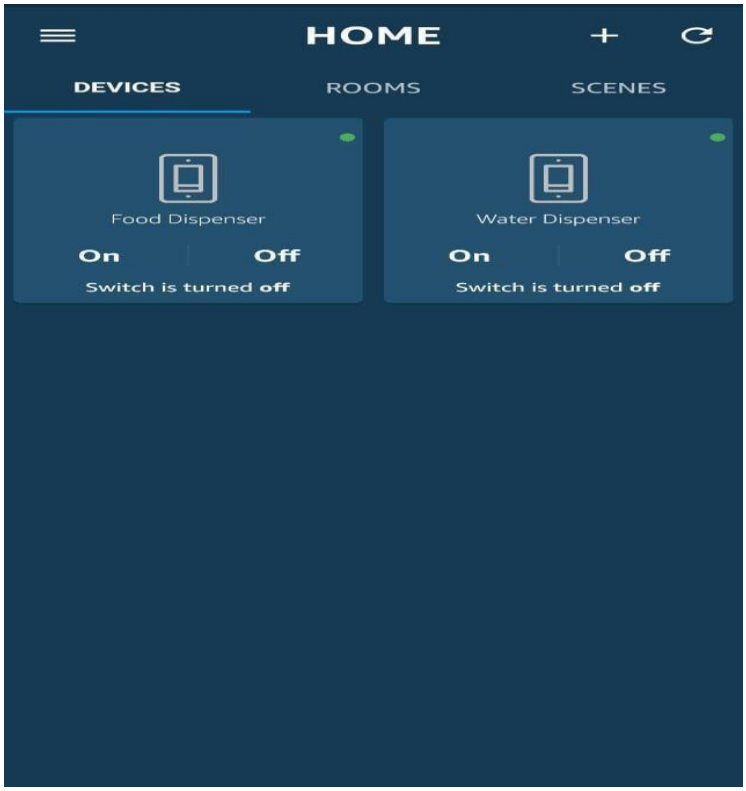

Fig 8. Sinric pro application

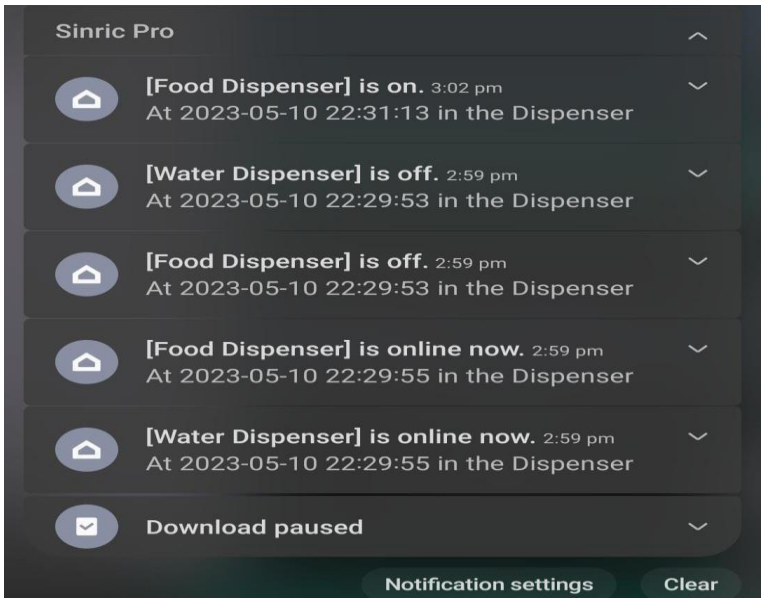

Fig 8.2 Notification from the Sinric pro application.

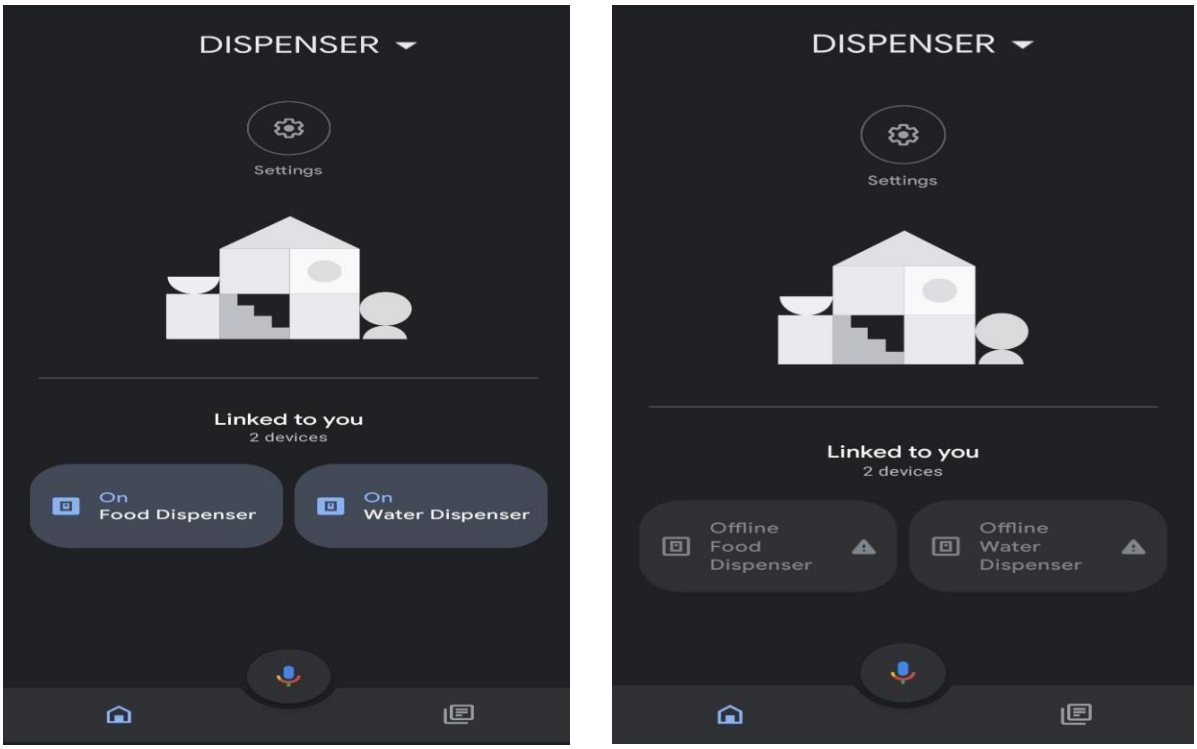

Fig 8.3 Google home Application.

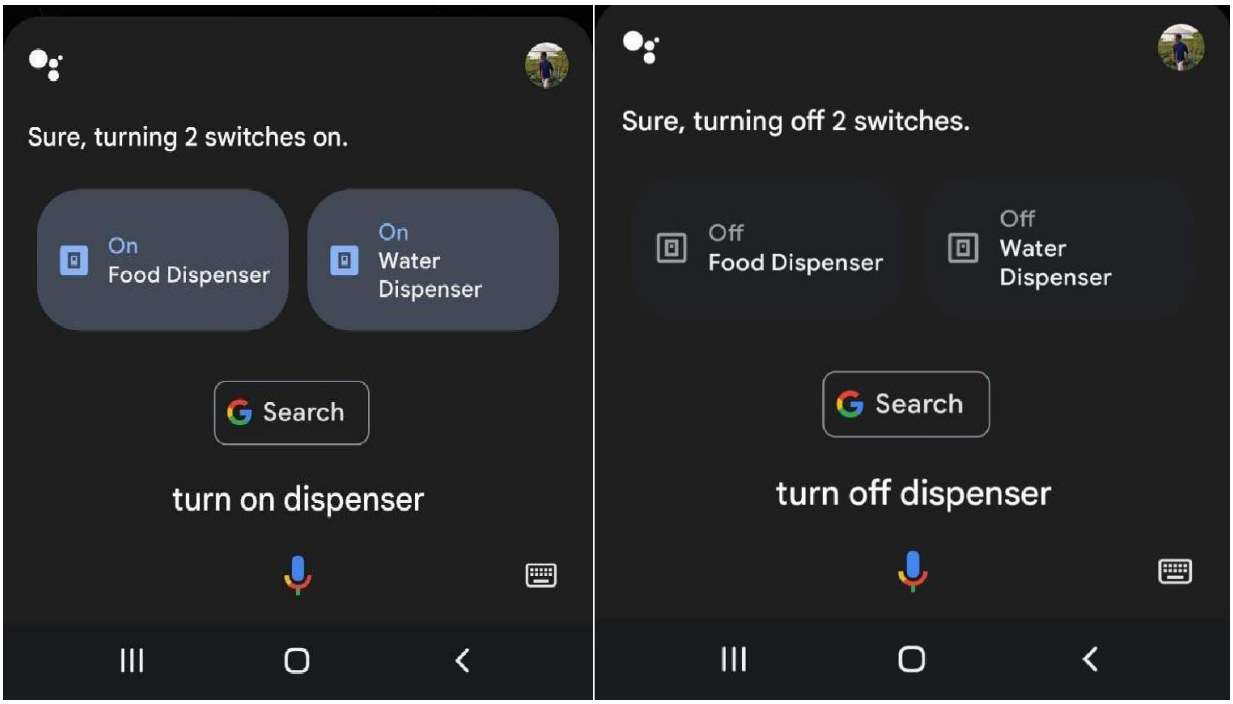

Fig 8.4 Response of Google assistance

8.1 Fabricated image of the robot:

Fig 8.5 Front view of the fabricated robot

Fig 8.6 Isometric View of the fabricated robot.

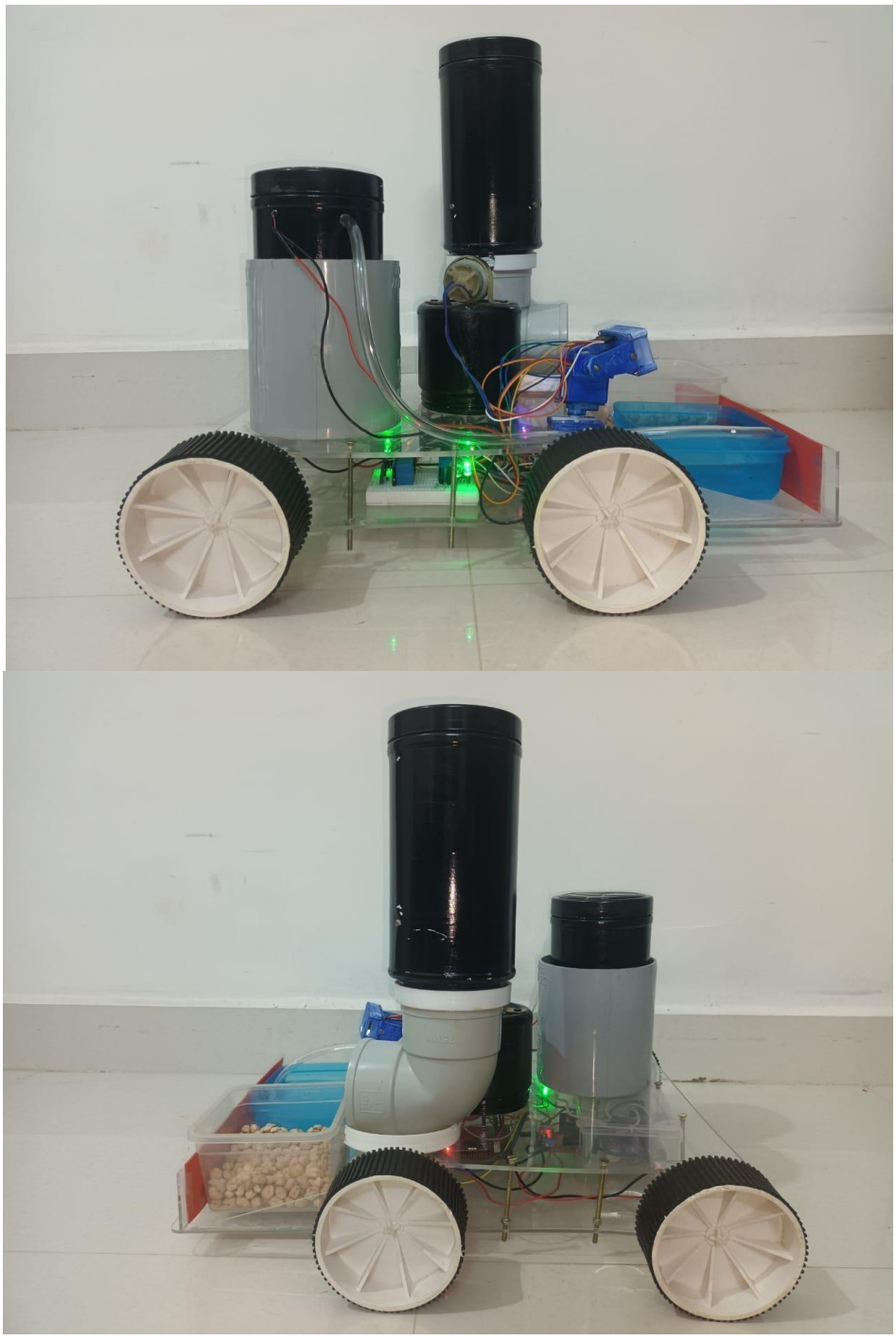

Fig 8.7 Side Views of the fabricated robot.

CHAPTER 9

REFERENCES

[1] TAdetokunbo A. Adenowo, Jonathan C. Anyl, James A. Akobada, “Internet of Things based Pet Feeder Automation Using Raspberry Pi” International Journal of Scientific & Research. Volume 11, Issue 8, August 2020.

[2] M. Ibrahim, h. Zakaria, and E.W. Xian, “Pet food auto feeder by using Arduino,” In IOP Conference Series: Materials science and engineering, Volume 670, No.1, November 2019, p. 012069, IOP Publishing.

[3] Hifengfang, Lidaxu, Yunqiangzhu, Jiaerhengahati, Huanpei, Jianwuyan, Andzhihuiliu (2014), 'An Integrated System for Regional Environmental Monitoring and Management Based On IoT ' IEEE Transactions On Industrial Informatics, vol. 10, no.2, pp. 1596-1605.

[4] David Naranjo-Hern´andez, Laura M. Roa, Fellow, Javier Reina-Tosina, Senior Member IEEE, and Miguel´ Angel Estudillo - Valderrama (2012), 'SoM: A Smart Sensor for Human Activity Monitoring and Assisted Healthy Ageing' IEEE Transactions on Biomedical Engineering, vol. 59, no. 11, pp. 3177-3184.

[5] Seung-Chul Son, Nak-Woo Kim, Byung-Tak Lee Chae Ho Cho, and Jo Woon Chong (2016), ' A Time Synchronization Technique for CoAP - based Home Automation Systems ' IEEE Transactions on Consumer Electronics, vol. 62, no. 1, pp. 10-16.

[6] Jaeseok Yun, Il - Yeop Ahn, Nak -Myung Kim (2015),' A Device Software Platform for Consumer Electronics Based on the Internet of Things ' IEEE Transactions on Consumer Electronics, vol. 61, no. 4, pp. 564-571.

[7] Thinagaran Perumal, A R Ramli, Chui Yew Leong (2014),'Interoperability Framework for Smart Home Systems" IEEE vol.2, no.2, pp. 659-663. [8] B. Ravi babu, P. Pavan Kumar, Dr. P. G. Kuppusamy, “Arduino Mega Based PET Feeding Automation”, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), Volume 14, Issue 4, Ser. I (July-August 2019) PP 13-16, DOI:10.9790/2834-14040113

[8] Shifengfang, Lidaxu, Yunqiangzhu, Jiaerhengahati, Huanpei, Jianwuyan, Andzhihuiliu (2014), 'An Integrated System for Regional Environmental Monitoring And Management Based On IoT ' IEEE Transactions On Industrial Informatics, vol. 10, no.2, pp.1596-1605. [9] .George Mois, Teodora Sanislav, and Silviu C. Olea, (2012), A Cyber-Physical System Environmental Monitoring' vol. 6, no. 14, pp. 2189–2197.

[10]. David Naranjo-Hern´andez, Laura M. Roa, Fellow, Javier Reina-Tosina, Senior Member IEEE, and Miguel´ Angel Estudillo - Valderrama (2012), 'SoM: A Smart Sensor for Human Activity Monitoring and Assisted Healthy Ageing' IEEE Transactions on Biomedical Engineering, vol. 59, no. 11, pp.3177-3184.

[11] P. Martens, M. J. Enders-Slegers, and J. K. Walker, “The emotional lives of companion animals: Attachment and subjective claims by owners of cats and dogs,” Anthrozoös, vol. 29, no. 1, pp. 73–88, 2016.

[12] H. S. Nugrahaeni, “The Relation between Pet Attachment and Pet Owners' Quality of Life,” Universitas Negeri Semarang, Semarang, 2016.

[13] Republic of Indonesia, Kitab Undang-undang Hukum Pidana (Penal Code).

[14] P. H. J. Kahn, B. Friedman, D. R. Pérez-Granados and N. G. Freier, “Robotic pets in the lives of preschool children,” Interaction Studies, vol. 7, no. 3, pp. 405-436, 2006.

[15] C. W. D. Lumoindong and J. W. Simatupang, “CoFiBot, companion and firefighting robot,” in Proc. National Seminar on Energy and Technology, Bekasi, 2018.

[16] H. Herzog, “The impact of pets on human health and psychological well-being: Fact, fiction, or hypothesis?” Current Directions in Psychological Science, vol. 20, no. 4, pp. 236-239, 2011.

[17] S. Jittpalapong, O. Rungphisutthipongse, S. Maruyama, J. J. Schaefer, and R. W. Stich, “Detection of hepatology canis in stray dogs and cats in Bangkok, Thailand,” Annals New York Academy of Sciences, vol. 1081, pp. 479-488, 2006

. [18] C. Maia, B. Almeida, M. Coimbra, M. C. Fernandes, J. M. Cristóvão, C. Ramos, Â. Martins, F. Martinho, P. Silva, N. Neves, M. Nunes, M. L. Vieira, L. Cardoso, and L. Campino, “Bacterial and protozoal agents of canine vector-borne diseases in the bloodof domestic and stray dogs from southern Portugal,” Parasites & Vectors, vol. 8, no. 138, pp. 1-7, 2015.

[19] D. Hardawar, “Sony Aibo hands-on: An adorable robo-pup that needs training,” 23 August 2018. [Online]. Available: https://www.engadget.com/2018/08/23/sony-aibo-hands-on/. [Accessed 10 October 2019].

[20]M. Fulmer, “Minimotos kick-start toy quest’s fortunes,” Los Angeles Times, 29 March 2005. [Online]. Available: https://www.latimes.com/archives/la-xpm-2005-mar-29-fimoto29-story.html. [Accessed 15 October 2019].

APPENDIX:

Coding of the HTML web page:

```
      ESP32CAM- CAR MOVEMENT AND PAN TILT CODE
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "Arduino.h"
#include "esp_camera.h"
#include "esp32-hal-ledc.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include <WiFi.h>

#define CAMERA_MODEL_AI_THINKER

#define PWDN_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

/* Wifi Crdentials */ /* Replace your SSID and Password */
const char* ssid = "Badri";
const char* password = "Badri235";

/* Defining DC motor, Servo and Flash LED pins */
```

```
const int RMotor1 = 14;
const int RMotor2 = 15;
const int LMotor1 = 2;
const int LMotor2 = 3;
const int panServo = 12;
const int tiltServo = 13;
const int FlashPin= 4;



/* Defining initial values */
int speed = 255;
int panVal = 4875;
int tiltVal = 4875;

#define PART_BOUNDARY "123456789000000000000987654321"
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;
httpd_handle_t camera_httpd = NULL;

/* Stream handler */
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];

  static int64_t last_frame = 0;
  if (!last_frame) {
    last_frame = esp_timer_get_time();
  }

  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->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) {
    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;
  }
```

```
    int64_t fr_end = esp_timer_get_time();
    int64_t frame_time = fr_end - last_frame;
    last_frame = fr_end;
    frame_time /= 1000;
    /*Serial.printf("MJPG: %uB %ums (%.1ffps)\n",
                  (uint32_t)(_jpg_buf_len),
                  (uint32_t)frame_time, 1000.0 / (uint32_t)frame_time
                 );*/
  }

  last_frame = 0;
  return res;
}

/* Command handler */
static esp_err_t cmd_handler(httpd_req_t *req)
{
  char* buf;
  size_t buf_len;
  char variable[32] = {0,};
  char value[32] = {0,};

  buf_len = httpd_req_get_url_query_len(req) + 1;
  if (buf_len > 1) {
    buf = (char*)malloc(buf_len);
    if (!buf) {
      httpd_resp_send_500(req);
      return ESP_FAIL;
    }
    if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
      if (httpd_query_key_value(buf, "var", variable, sizeof(variable)) == ESP_OK &&
          httpd_query_key_value(buf, "val", value, sizeof(value)) == ESP_OK) {
      } else {
        free(buf);
        httpd_resp_send_404(req);
        return ESP_FAIL;
      }
    } else {
      free(buf);
      httpd_resp_send_404(req);
      return ESP_FAIL;
    }
```

```
  free(buf);
} else {
  httpd_resp_send_404(req);
  return ESP_FAIL;
}

int val = atoi(value);
sensor_t * s = esp_camera_sensor_get();
int res = 0;
/* Flash LED control */
if (!strcmp(variable, "flash"))
{
  ledcWrite(4, val);
}
else if (!strcmp(variable, "speed"))
{
  /* Setting the motor speed */
  if      (val > 255) val = 255;
  else if (val <   0) val = 0;
  speed = val;
}
/* Robot direction control */
else if (!strcmp(variable, "car")) {
  if (val == 1) {
    Serial.println("Forward");
    ledcWrite(14, speed);
    ledcWrite(15, 0);
    ledcWrite(2, 0);
    ledcWrite(3, speed);
  }
  else if (val == 2) {
    Serial.println("Turn Left");
    ledcWrite(14, speed);
    ledcWrite(15, 0);
    ledcWrite(2, speed);
    ledcWrite(3, 0);
  }
  else if (val == 3) {
    Serial.println("Stop");
    ledcWrite(14, 0);
    ledcWrite(15, 0);
    ledcWrite(2, 0);
```

```
  ledcWrite(3, 0);
}
else if (val == 4) {
  Serial.println("Turn Right");
  ledcWrite(14, 0);
  ledcWrite(15, speed);
  ledcWrite(2,  0);
  ledcWrite(3, speed);

}
else if (val == 5) {
  Serial.println("Backward");
  ledcWrite(14, 0);
  ledcWrite(15, speed);
  ledcWrite(2, speed);
  ledcWrite(3, 0);
}
else if (val == 7) {
  Serial.println("Foward Right");
  ledcWrite(14, 0);
  ledcWrite(15, 0);
  ledcWrite(2, 0);
  ledcWrite(3, speed);
}
else if (val == 6) {
  Serial.println("Foward Left");
  ledcWrite(14, speed);
  ledcWrite(15, 0);
  ledcWrite(2, 0);
  ledcWrite(3,0 );
}
else if (val == 9) {
  Serial.println("Backward Right");
  ledcWrite(14, 0);
  ledcWrite(15,speed);
  ledcWrite(2, 0);
  ledcWrite(3, 0);
}
else if (val == 8) {
  Serial.println("Backward Left");
  ledcWrite(14, 0);
  ledcWrite(15,0);
```

```
    ledcWrite(2, speed);
    ledcWrite(3, 0);
  }
}
/* Pan and Tilt servo control */
else if (!strcmp(variable, "pantilt")) {
  if (val == 1) {
    Serial.println("Tilt Up");
    if   (tiltVal  >  6500)  tiltVal  =  6500;
    else if (tiltVal < 3250) tiltVal = 3250;
    tiltVal = tiltVal - 200;
    ledcWrite(13, tiltVal);
  }
  else if (val == 2) {
    Serial.println("Pan Left");
    if    (panVal  >  6500)  panVal  =  6500;
    else if (panVal < 3250) panVal = 3250;
    panVal = panVal + 200;
    ledcWrite(12, panVal);
  }
  else if (val == 3) {
    Serial.println("Pan Right");
    if    (panVal  >  6500)  panVal  =  6500;
    else if (panVal < 3250) panVal = 3250;
    panVal = panVal - 200;
    ledcWrite(12, panVal);
  }
  else if (val == 4) {
    Serial.println("Tilt Down");
    if   (tiltVal  >  6500)  tiltVal  =  6500;
    else if (tiltVal < 3250) tiltVal = 3250;
    tiltVal = tiltVal + 200;
    ledcWrite(13, tiltVal);
  }

}
else
{
  Serial.println("variable");
  res = -1;
}
```

```
  if (res) {
    return httpd_resp_send_500(req);
  }

  httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
  return httpd_resp_send(req, NULL, 0);
}


static esp_err_t status_handler(httpd_req_t *req) {
  static char json_response[1024];

  sensor_t * s = esp_camera_sensor_get();
  char * p = json_response;
  *p++ = '{';

  p += sprintf(p, "\"framesize\":%u,", s->status.framesize);
  p += sprintf(p, "\"quality\":%u,", s->status.quality);
  *p++ = '}';
  *p++ = 0;
  httpd_resp_set_type(req, "application/json");
  httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
  return httpd_resp_send(req, json_response, strlen(json_response));
}

/* Index HTML page design */
static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,initial-scale=1">
        <title>Iot Based Pet Care Robot</title>
        <style>
            .button {background-color: #0097b5;border: none;border-radius: 4px;color:
white;padding: 10px 25px;text-align: center;font-size: 16px;margin: 4px 2px;cursor:
pointer;}
          .slider {appearance: none;width: 70%;height: 15px;border-radius: 10px;back-
ground: #d3d3d3;outline: none;}
             .slider::-webkit-slider-thumb {appearance: none;appearance: none;width:
30px;height: 30px;border-radius: 50%;background: #0097b5;}
            .label {color: #0097b5;font-size: 18px;}
```

```
        </style>
    </head>
    <body>
    <div align=center><h1 style="color:blue;">IOT BASED PET CARE ROBOT</h1></div>
    <br/>
    <div align=center> <img id= "camstream" src="" style='width:300px;'></div>
    <br/>
    <br/>
    <div align=center>
        <button  class="button"  id="forwardleft"  ontouchstart="fetch(document.loca-
tion.origin+'/control?var=car&val=6');"              ontouchend="fetch(document.loca-
tion.origin+'/control?var=car&val=3');" >Forward Left</button>
          <button   class="button"   id="forward"   ontouchstart="fetch(document.loca-
tion.origin+'/control?var=car&val=1');"              ontouchend="fetch(document.loca-
tion.origin+'/control?var=car&val=3');" >Forward</button>
       <button  class="button"  id="forwardright"  ontouchstart="fetch(document.loca-
tion.origin+'/control?var=car&val=7');"              ontouchend="fetch(document.loca-
tion.origin+'/control?var=car&val=3');" >Forward Right</button>
    </div>
    <br/>
    <div align=center>
         <button   class="button"   id="turnleft"   ontouchstart="fetch(document.loca-
tion.origin+'/control?var=car&val=2');"              ontouchend="fetch(document.loca-
tion.origin+'/control?var=car&val=3');" >Turn Left</button>
     <button class="button" id="stop" onclick="fetch(document.location.origin+'/con-
trol?var=car&val=3');">Stop</button>
        <button   class="button"   id="turnright"   ontouchstart="fetch(document.loca-
tion.origin+'/control?var=car&val=4');"              ontouchend="fetch(document.loca-
tion.origin+'/control?var=car&val=3');" >Turn Right</button>
    </div>
    <br/>
    <div align=center>
       <button  class="button"  id="backwardleft"  ontouchstart="fetch(document.loca-
tion.origin+'/control?var=car&val=8');"              ontouchend="fetch(document.loca-
tion.origin+'/control?var=car&val=3');" >Backward Left</button>
         <button   class="button"   id="backward"   ontouchstart="fetch(document.loca-
tion.origin+'/control?var=car&val=5');"              ontouchend="fetch(document.loca-
tion.origin+'/control?var=car&val=3');">Backward</button>
      <button  class="button"  id="backwardright"  ontouchstart="fetch(document.loca-
tion.origin+'/control?var=car&val=9');"              ontouchend="fetch(document.loca-
tion.origin+'/control?var=car&val=3');" >Backward Right</button>
    </div>
```

```
    <br/>
    <div align=center>
    <label class="label">Flash</label>
     <input type="range" class="slider" id="flash" min="0" max="255" value="0" on-
change="try{fetch(document.location.origin+'/con-
trol?var=flash&val='+this.value);}catch(e){}">
    </div>
    <br/>
    <div align=center>
    <label class="label">Speed</label>
     <input type="range" class="slider" id="speed" min="0" max="255" value="255" on-
change="try{fetch(document.location.origin+'/con-
trol?var=speed&val='+this.value);}catch(e){}">
    </div>
     <br/>
    <div align=center>
   <button class="button" id="tiltup" onclick="fetch(document.location.origin+'/con-
trol?var=pantilt&val=1');" >Tilt Up</button>
    </div>
    <div align=center>
             <button   class="button"   id="panleft"   onclick="fetch(document.loca-
tion.origin+'/control?var=pantilt&val=2');" >Pan Left</button>
            <button   class="button"   id="panright"   onclick="fetch(document.loca-
tion.origin+'/control?var=pantilt&val=3');" >Pan Right</button>
    </div>
     <div align=center>
            <button   class="button"   id="tiltdown"   onclick="fetch(document.loca-
tion.origin+'/control?var=pantilt&val=4');" >Tilt Down</button>
    </div>
    <br/>
    <script>
           window.onload = document.getElementById("camstream").src = window.loca-
tion.href.slice(0, -1) + ":81/stream";
    </script>
    </body>
</html>
)rawliteral";

static esp_err_t index_handler(httpd_req_t *req){
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML));
}
```

```
void startCameraServer()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

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

    httpd_uri_t status_uri = {
        .uri       = "/status",
        .method    = HTTP_GET,
        .handler   = status_handler,
        .user_ctx  = NULL
    };

    httpd_uri_t cmd_uri = {
        .uri       = "/control",
        .method    = HTTP_GET,
        .handler   = cmd_handler,
        .user_ctx  = NULL
    };

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

    Serial.printf("Starting web server on port: '%d'\n", config.server_port);
    if (httpd_start(&camera_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(camera_httpd, &index_uri);
        httpd_register_uri_handler(camera_httpd, &cmd_uri);
        httpd_register_uri_handler(camera_httpd, &status_uri);
    }

    config.server_port += 1;
    config.ctrl_port += 1;
```

```
    Serial.printf("Starting stream server on port: '%d'\n", config.server_port);
    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(stream_httpd, &stream_uri);
    }
}


void initMotors()
{
  /* Configuring PWM channels for DC motors */
  /* ledcSetup(Channel, Frequency, Resolution) */
  ledcSetup(14, 2000, 8); /* 2000 hz PWM, 8-bit resolution and range from 0 to 255 */
  ledcSetup(15, 2000, 8);
  ledcSetup(2, 2000, 8);
  ledcSetup(3, 2000, 8);
  /* Attaching the channel to the GPIO to be controlled */
  /* ledcAttachPin(GPIO, Channel) */
  ledcAttachPin(RMotor1, 14);
  ledcAttachPin(RMotor2, 15);
  ledcAttachPin(LMotor1, 2);
  ledcAttachPin(LMotor2, 3);
}

void initServo() {
  /* Configuring PWM channels for servo motors */
  ledcSetup(12, 50, 16); /*50 hz PWM, 16-bit resolution and range from 3250 to 6500
*/
  ledcSetup(13, 50, 16);
  ledcAttachPin(panServo, 12);
  ledcAttachPin(tiltServo, 13);
  /* Initializing servo motors */
  ledcWrite(12, panVal);
  ledcWrite(13, tiltVal);
}


void initFlash() {
  /* Configuring PWM channels for Falsh LED */
  ledcSetup(4, 5000, 8); /* 5000 hz PWM, 8-bit resolution and range from 0 to 255 */
  ledcAttachPin(FlashPin, 4);
}
```

```
void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  initMotors();
  initServo();
  initFlash();

  Serial.println();

  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;
  if (psramFound()) {
     config.frame_size = FRAMESIZE_QVGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_QVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

  esp_err_t err = esp_camera_init(&config);
```

```
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_QVGA);
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);

  /* Connecting to WiFi */
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
  for (int i = 0; i < 5; i++) {
    ledcWrite(4, 10);
    delay(50);
    ledcWrite(4, 0);
    delay(50);
  }

}

void loop() {

}
```

ESP32 DISPENSER CODE

```
#include <Arduino.h>                                    //ArduinoJson Library:
https://github.com/bblanchon/ArduinoJson
#include <WiFi.h>
```

```
#include "SinricPro.h"                            //SinricPro Library: https://sinricpro.github.io/esp8266-esp32-sdk/
#include "SinricProSwitch.h"
#include <map>


#ifdef ENABLE_DEBUG
       #define DEBUG_ESP_PORT Serial
         #define NODEBUG_WEBSOCKETS                  //arduinoWebSockets Library: https://github.com/Links2004/arduinoWebSockets
       #define NDEBUG
#endif

#define WIFI_SSID         "Badri"      //Enter WiFi Name
#define WIFI_PASS         "Badri235"      //Enter WiFi Password
#define APP_KEY         "d0d5cf64-06f6-493a-b570-0b4c96f5fac8"              //Enter APP-KEY
#define APP_SECRET           "b229069b-066-46b2-8ebe-644877f53200-c37ddb22-5ec9-47c2-8980-28227457a0e3"     //Enter APP-SECRET

//Enter the device IDs here
#define device_ID_1   "644e32bc32d40edc3ff636c"  //SWITCH 1 ID
#define device_ID_2   "644e330312d40edc3ff63cf"  //SWITCH 2 ID


// define the GPIO connected with Relays and switches
#define RelayPin1 23 //D23
#define RelayPin2 22  //D22


#define SwitchPin1 13  //D13
#define SwitchPin2 12  //D12

#define wifiLed    2   //D2


#define BAUD_RATE   9600

#define DEBOUNCE_TIME 250

typedef struct {      // struct for the std::map below
  int relayPIN;
```

```
  int flipSwitchPIN;
} deviceConfig_t;


 std::map<String, deviceConfig_t> devices = {
     //{deviceId, {relayPIN,  flipSwitchPIN}}
    {device_ID_1, {  RelayPin1, SwitchPin1 }},
    {device_ID_2, {  RelayPin2, SwitchPin2 }}

};

typedef struct {      // struct for the std::map below
  String deviceId;
  bool lastFlipSwitchState;
  unsigned long lastFlipSwitchChange;
} flipSwitchConfig_t;

std::map<int, flipSwitchConfig_t> flipSwitches;      // this map is used to map
flipSwitch PINs to deviceId and handling debounce and last flipSwitch state checks
                                                     // it will be setup in
"setupFlipSwitches" function, using informations from devices map

void setupRelays() {
  for (auto &device : devices) {           // for each device (relay, flipSwitch
combination)
    int relayPIN = device.second.relayPIN; // get the relay pin
    pinMode(relayPIN, OUTPUT);             // set relay pin to OUTPUT
    digitalWrite(relayPIN, HIGH);
  }
}

void setupFlipSwitches() {
  for (auto &device : devices)  {                      // for each device (relay
/ flipSwitch combination)
    flipSwitchConfig_t flipSwitchConfig;               // create a new flipSwitch
configuration

    flipSwitchConfig.deviceId = device.first;          // set the deviceId
    flipSwitchConfig.lastFlipSwitchChange = 0;         // set debounce time
    flipSwitchConfig.lastFlipSwitchState = false;      // set lastFlipSwitchState
to false (LOW)--
```

```
    int flipSwitchPIN = device.second.flipSwitchPIN;  // get the flipSwitchPIN

     flipSwitches[flipSwitchPIN] = flipSwitchConfig;     // save the flipSwitch
config to flipSwitches map
    pinMode(flipSwitchPIN, INPUT_PULLUP);                  // set the flipSwitch
pin to INPUT
  }
}

bool onPowerState(String deviceId, bool &state)
{
  Serial.printf("%s: %s\r\n", deviceId.c_str(), state ? "on" : "off");
  int relayPIN = devices[deviceId].relayPIN; // get the relay pin for corresponding
device
  digitalWrite(relayPIN, state);             // set the new relay state
  return true;
}

void handleFlipSwitches() {
  unsigned long actualMillis = millis();                                    //
get actual millis
  for (auto &flipSwitch : flipSwitches) {                                   //
for each flipSwitch in flipSwitches map
      unsigned long lastFlipSwitchChange = flipSwitch.second.lastFlipSwitch-
Change; // get the timestamp when flipSwitch was pressed last time (used to
debounce / limit events)

   if (actualMillis - lastFlipSwitchChange > DEBOUNCE_TIME) {               //
if time is > debounce time...

     int flipSwitchPIN = flipSwitch.first;                                  //
get the flipSwitch pin from configuration
     bool lastFlipSwitchState = flipSwitch.second.lastFlipSwitchState;      //
get the lastFlipSwitchState
     bool flipSwitchState = digitalRead(flipSwitchPIN);                     //
read the current flipSwitch state
     if (flipSwitchState != lastFlipSwitchState) {                          //
if the flipSwitchState has changed...
#ifdef TACTILE_BUTTON
       if (flipSwitchState) {                                               //
if the tactile button is pressed
#endif
```

```
        flipSwitch.second.lastFlipSwitchChange = actualMillis;                //
update lastFlipSwitchChange time
        String deviceId = flipSwitch.second.deviceId;                         //
get the deviceId from config
        int relayPIN = devices[deviceId].relayPIN;                            //
get the relayPIN from config
        bool newRelayState = !digitalRead(relayPIN);                          //
set the new relay State
        digitalWrite(relayPIN, newRelayState);                                //
set the trelay to the new state

        SinricProSwitch &mySwitch = SinricPro[deviceId];                      //
get Switch device from SinricPro
        mySwitch.sendPowerStateEvent(!newRelayState);                         //
send the event
#ifdef TACTILE_BUTTON
       }
#endif
      flipSwitch.second.lastFlipSwitchState = flipSwitchState;                //
update lastFlipSwitchState
     }
   }
  }
}

void setupWiFi()
{
  Serial.printf("\r\n[Wifi]: Connecting");
  WiFi.begin(WIFI_SSID, WIFI_PASS);

  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.printf(".");
    delay(250);
  }
  digitalWrite(wifiLed, HIGH);
     Serial.printf("connected!\r\n[WiFi]:   IP-Address   is   %s\r\n",   WiFi.lo-
calIP().toString().c_str());
}

void setupSinricPro()
{
```

```
  for (auto &device : devices)
  {
    const char *deviceId = device.first.c_str();
    SinricProSwitch &mySwitch = SinricPro[deviceId];
    mySwitch.onPowerState(onPowerState);
  }

  SinricPro.begin(APP_KEY, APP_SECRET);
  SinricPro.restoreDeviceStates(true);
}

void setup()
{
  Serial.begin(BAUD_RATE);

  pinMode(wifiLed, OUTPUT);
  digitalWrite(wifiLed, LOW);

  setupRelays();
  setupFlipSwitches();
  setupWiFi();
  setupSinricPro();
}

void loop()
{
  SinricPro.handle();
  handleFlipSwitches();
}
```

turnitin

Similarity Report ID: oid:9380:35092024

16% Overall Similarity

Top sources found in the following databases:

- 12% Internet database
- Crossref database
- 13% Submitted Works database
- 3% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**Engineers Australia on 2020-11-17** Submitted works	2%
2	**irjmets.com** Internet	1%
3	**Higher Education Commission Pakistan on 2022-05-17** Submitted works	<1%
4	**Asia Pacific University College of Technology and Innovation (UCTI) on...** Submitted works	<1%
5	**Engineers Australia on 2020-07-17** Submitted works	<1%
6	**Middle East College of Information Technology on 2022-06-25** Submitted works	<1%
7	**Universiti Teknologi Malaysia on 2023-02-18** Submitted works	<1%