Chapter 1

INTRODUCTION

Automatic pet feeder is one of the new technologies for feeding pet. It will help pet owner to take care of their pet while they are not at home. Even when owners are not at home, they still can feed their pet.

It is built to help pet owner for taking care of their pet in their absence. IoT pet feeder is a device that will be controlled by a mobile application through internet. It will automatically dispense predetermined amount of food and water to the bowls. As pet lovers, user should understand that, pets also need a proper diet management. Whether user is away from home unexpectedly or simply would like one less chore to worry about, user can feel secure that the beloved pet will be cared for and fed on time every time.

This device will solve a major problem which pet owners face i.e., making sure that each pet has access to a healthy amount of food throughout the day, regardless of the owner's schedule.

The automatic pet feeder will give pet owners a solution to both problems, thereby improving the lives of both pets and owners by allowing the owner to reliably provide food to a pet at the time the owner wishes and keep the pet from reaching the food stored for later feedings. Many animal feed systems can be designed to function as an automatic device that allow the user to feed whenever he wishes from anywhere through internet. The purpose of having sensors in a system like this is to automate the feed process completely with less human interference

1.1 PETS AND EMOTIONS

A pet, or companion animal, is an animal kept primarily for a person's company or entertainment rather than as a working animal, livestock, or a laboratory animal. Popular pets are often considered to have attractive/cute appearances, intelligence, and relatable personalities, but some pets may be taken in on an altruistic basis (such as a stray animal) and accepted by the owner regardless of these characteristics. Two of the most popular pets are dogs and cats..

Pets provide their owners, or guardians, both physical and emotional benefits. Walking a dog can provide both the human and the dog with exercise, fresh air, and social interaction. Pets can give companionship to people who are living alone or elderly adults who do not have adequate social interaction with other people. There is a medically approved class of therapy animals that are brought to visit confined humans, such as children in hospitals or elders in nursing homes. Pet

therapy utilizes trained animals and handlers to achieve specific physical, social, cognitive, or emotional goals with patients

People most commonly get pets for companionship, to protect a home or property, or because of the perceived beauty or attractiveness of the animals. This means that a pet will have all of the basic emotions: joy, fear, anger, disgust, and, yes, love, but the dog does not experience the more complex emotions like guilt, pride, and shame.

Emotions

As humans, we feel many emotions: joy, sadness, anger, surprise, empathy, sympathy, trust, love, and envy. We sometimes have a difficult time recognizing our own emotions, much less interpreting those of others. But it seems that our dogs have an easy time reading our feelings. Do you agree? Do you think your pet knows when you are sad? Does he know when you are afraid? Is it bad to display your emotions in front of your furry friend?

Most dog owners contend their pooch truly does understand them. And that understanding extends beyond verbal communication. Sure, if you tell a well-trained dog to "sit" he will drop down on his haunches. If you say, "fetch" she will scoot after the thrown ball. But do dogs understand much more than the spoken command? Yes! Dogs discern subtle visual cues and they "get the message" without a single word being uttered. Our dogs understand our emotions and communicate their understanding without us saying anything! When they gather this visual information and add it to our tone of voice when we do speak, they can communicate with us!

1.2 PET SHELTER AND ENVIRONMENT

An animal shelter or pound is a place where stray, lost, abandoned or surrendered animals – mostly dogs and cats – are housed. The word "pound" has its origins in the animal pounds of agricultural communities, where stray livestock would be penned or impounded until they were claimed by their owners.

An animal in a shelter has four outcomes: return to owner, adoption, transfer to another shelter or rescue facility, or euthanasia.[6] Return to owner is when a stray animal, that was found and housed at the shelter, is picked up by its owner. Most animal shelters practice adoption, where an animal in their care is given or sold to an individual who will keep it and care for it. Some shelters work with rescue organizations, giving an animal to the rescue rather than adopting it to an individual. Some jurisdictions mandate that shelters cooperate with rescues; some shelters utilize

rescues to offload animals with health or behaviour problems that they are not equipped to deal with. Many shelters practice some level of euthanasia.[5][6]

A house will provide shelter from sun, wind, rain and cold and will be a comforting refuge to your pet while you are out. There are basically two types of kennels; the traditional wooden ones and the modern plastic ones.

A suitable environment for any animal encompasses a wide range of needs. It should provide space in which the animals can express their physical and social behaviour. It should also be secure and sufficiently hygienic to prevent disease transmission between animals, and between humans and animals.

1.3 WEARABLE DEVICES

Wearable technology is any kind of electronic device designed to be worn on the user's body. Such devices can take many different forms, including jewellery, accessories, medical devices, and clothing or elements of clothing. The term *wearable computing* implies processing or communications capabilities, but the sophistication among wearables can vary.

Modern wearable technology falls under a broad spectrum of usability, including smartwatches, fitness trackers such as the Fitbit Charge, VR headsets, smart jewellery, web-enabled glasses and Bluetooth headsets. Wearables work differently, based on the category they belong to, such as health, fitness, or entertainment. Predominantly, wearable technology functions by incorporating microprocessors, batteries and connectivity to the internet so the collected data can be synced with other electronics, such as mobile devices or laptops.

Examples of wearable technology

- Smart jewellery. This can include smart rings, wristbands, watches and pins. ...
- Body-mounted sensors. ...
- Fitness trackers. ...
- Smart clothing. ...
- Augmented reality (AR) headsets. ...
- VR headsets. ...
- AI hearing aids and so on

Wearable technology has many uses, including health and fitness tracking, chronic disease management, interactive gaming, performance monitoring and navigation tracking. The following

are the most popular current and next-generation applications of wearable technology: Epidermal skin technology.

The number of connected wearable devices worldwide has more than doubled in the space of three years, increasing from 325 million in 2016 to 722 million in 2019. The number of devices is forecast to reach more than one billion by 2022.

1.4 INTERNET OF THINGS

The Internet of things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network, and be individually addressable.

The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, as well as learning. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. The IoT's major significant trend in recent years is the explosive growth of devices connected and controlled via the Internet. The wide range of applications for IoT technology mean that the specifics can be very different from one device to the next but there are basic characteristics shared by most. The IoT creates opportunities for more direct integration of the physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions

IoT networks can be classified into four broad classes as follow.

- Cellular networks like LTE-M, NB-IoT, etc.
- LAN/PAN like Bluetooth, WI-FI, etc.
- LPWAN like Lora WAN, Sigfox, etc.
- Mesh protocols like RFID, ZigBee, Z-wave, etc.

1.5 MOBILE APPLICATIONS

A mobile application, most referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Apps are generally small, individual software units with limited function. This use of app software was originally

popularized by Apple Inc. and its App Store, which offers thousands of applications for the iPhone, iPad and iPod Touch.

A mobile application, most referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Apps are generally small, individual software units with limited function. This use of app software was originally popularized by Apple Inc. and its App Store, which offers thousands of applications for the iPhone, iPad and iPod Touch.

There are several types of apps currently available.

Gaming apps**:** The equivalent of computer video games, they are among the most popular types of apps. They account for one-third of all app downloads and three-fourths of all consumer spending.

Productivity apps**:** These focus on improving business efficiency by easing various tasks such as sending emails, tracking work progress, booking hotels, and much more.

Lifestyle and entertainment apps**:** Increasingly popular, these encompass many aspects of personal lifestyle and socialization such as dating, communicating on social media, as well as sharing (and watching) videos. Some of the most widely known apps such as Netflix, Facebook or TikTok fall into this category.

Chapter 2

LITERATURE SURVEY

[1]Juneyoung Ahn, Jini Kwon, Hyejeong Nam, Hyun-Kook Jang and Jee-In Kim, "Pet buddy: A wearable device for canine behavior recognition using a single IMU" 2016 International Conference on Big Data and Smart Computing (BigComp), Hong Kong, China, 2016, pp. 419-422, doi: 10.1109/BIGCOMP.2016.7425961.

A few households raising pets is increasing worldwide. On the other hand, there are issues on health of the pets such as the concern of controlling meals, scheduling exercises, as well as increasing stress levels of the pets that are left behind due to the owner's busy schedule. It is required to find resolutions for the issues. This research is the first step for the enhancement the health by monitoring behaviours of the pets. A wearable device using a single Inertial Measurement Unit (IMU) is developed. The device can be attached to canine and data from the device will be used to monitor and analyse behaviours of the canine.

[2]Y. -J. Lin, C. -W. Chuang, C. -Y. Yen, S. -H. Huang and S. -Y. Lee, "Smart Pet Clothing for Monitoring of Health and Mood" 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 2018, pp. 1-4, doi: 10.1109/ISCAS.2018.8351547.

A smart pet clothing with full hardware and software support for internet of things is proposed. The hardware comprises three parts: a special pet sensor, an analog front-end circuit for detecting electrocardiogram (ECG) and breath signals, and a micro printed circuit board with signal communication. The software also consists of three parts: an algorithm for biosignal processing, an application (app) as graphical user interface (GUI), and a web server for healthcare.

[3]C. Sazara and X. Gao, "Predicting Animal Shelter Pet Adoption Times and Feature Importance Analysis using CatBoost" 2022 IEEE 11th International Conference on Intelligent Systems (IS), Warsaw, Poland, 2022, pp. 1-4, doi: 10.1109/IS57118.2022.10019608.

Animal shelters in the US provide care for stray or unwanted animals. Animal shelters have become crucial for the welfare of the animals brought to these facilities. This study uses the intake and outcome data provided by the Austin Animal Centre from 2013 to July 2022. Main focus of the paper is to study different factors to determine possible stay times of animals in the shelter

until adoption and build a predictive model for the future adoption time. For this purpose, first, the intake and outcome data are put into single table format. Then, adoption times are divided into two major groups: Adopted in less than 15 days and more than 15 days. To measure the impact of different attributes, a tree-based boosting method is used. This method is trained, validated, and tested with the corresponding splits of the dataset. At the end, we provide an in-depth analysis of the attributes leading to different stay times. We also provide a GitHub link for the dataset and the trained models in this paper.

[4]T. Sangvanloy and K. Sookhanaphibarn, "Automatic Pet Food Dispenser by using Internet of Things (IoT)," 2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech), Kyoto, Japan, 2020, pp. 132-135, doi: 10.1109/LifeTech48969.2020.1570620257.

The objective of this project is to develop an automatic pet feeding with Internet of Things (IoT). We developed a feeder to help allocate dry food diet to small pets such as dogs and cats. It will be very useful whenever a pet owner is outside the residence and/or unable to feed his/her pets normally. When the free feeding is happened, it will cause the obesity of pets. This machine will be also used for monitor pet's eating habits to train the pets for scheduled meals.

[5]E. Jain, S. Badwaik, S. Khirwadkar, S. Thakare, M. Uike and P. H. Chandankhede, "Design of Smart Pet Food Dispenser using Embedded System," 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2023, pp. 1-5, doi: 10.1109/ESCI56872.2023.10100166.

The project's goal is to create an Internet of Things-based autonomous pet feeding system (IoT). Here, we're creating a feeder for pets and owners that may be used whenever a pet owner is away from home and unable to feed their animals normally. Pets will become obese if we give them free food. This device will also be used to train pets to eat on schedules by keeping track of their feeding patterns. The objective of the following project is to develop automatic pet feeding with the Internet of Things (IoT). Here we are developing a feeder for the pets and the owners that whenever a pet owner is outside the residence and unable to feed his or her pets normally. When we freely feed pets, it will cause obesity in pets. This machine will also be used to monitor the pet's eating habits to train the pets for scheduled meals.

[6] G. M. Debele and X. Qian, "Automatic Room Temperature Control System Using Arduino UNO R3 and DHT11 Sensor," 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 2020, pp. 428-432, doi: 10.1109/ICCWAMTIP51612.2020.9317307.

This paper presents designing and implementing an Automatic room temperature control system using the Arduino and DHT11 sensor. The fan speed control system has also been proposed. Here, the user sets the minimum and the maximum reference temperature range from the keypad. The DHT11 sensor senses the surrounding room temperature and gives the result in degrees Celsius. Both the reference and the measured values are displayed on the Liquid Crystal Display (LCD). The Arduino microcontroller, being the processing unit of the system, gets the sensor's measured value and compares it with the set threshold. The results are: when the measured room temperature is less than the minimum of the threshold value; then, the microcontroller turns on the heater. If the measured room temperature is greater than the maximum threshold value, then the fan triggered on. The speed of the fan will be controlled by the pulse width modulation (PWM) technique based upon the temperature difference between the sensor reading and the maximum threshold. The larger the temperature difference, the larger the fan's duty cycles, and the faster its speed. Finally, if the room's measured temperature is between the setpoint range, all the loads are turned off. That means the room temperature is maintained normal.

[7] H. Y. Lin et al., "DC servo speed control of an inkjet print head transport system using a phase-locked loop," Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE, Mie, Japan, 1996, pp. 458-463 vol.2, doi: 10.1109/AMC.1996.509292.

A low-cost servo control system has been developed to regulate the speed of a belt driven moving carriage. It is used by a print head transport system which is designed for an inkjet printer developed by OES/ITRI in Taiwan. The servo controller is realized by a phase-locked loop with a DC servo motor. The phase-locked loop includes a lead-lag compensator which is designed based on the experimental system identification of the print head transport system. The experimental transfer function of the system is obtained under a set of sweeping sinusoidal excitation. On the basis of that model, a lead-lag compensator is then designed by using the root locus method. The compensator is proven to be effective in regulating the speed variation of the carriage within 10% while the carriage moves at 33 inches/sec. Experiments are performed to verify the effectiveness

of the lead-lag compensator. This approach has made it possible for inkjet printers to be realized to demonstrate extreme improvement in printing throughput using low-cost ICs.

[8] Y. Chen and M. Elshakankiri, "Implementation of an IoT based Pet Care System," 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), Paris, France, 2020, pp. 256-262, doi: 10.1109/FMEC49853.2020.9144910.

As pet ownership is soaring each year, the demands for a higher quality of pet care products are increasing as well. This has driven the development of the Internet of Things (IoT) technology in this field. Using the technology of IoT, pet owners can remotely track their pet's activity and location, monitor their pet's health condition or even interact with their pets. All these smart pet care products are playing an indispensable role in the pet owner's daily life. In the present project, we apply the IoT technology to implement an integrated system including pet food feeder, water dispenser, and litter box, which are the three most fundamental elements that pet owners will be concerned about when they are busy or away from their pets. The three subsystems are connected to the local network with Arduino Uno boards and Wi-Fi modules. Furthermore, the data collected from each sensor are processed and displayed on a smartphone application. Thus, pet owners through only one single interface, they can obtain all the information regarding pet's food consumption, water consumption, as well as defecation timing, duration, and frequency. Additionally, a controlling function is also enabled in the application for the pet owners to dispense food anytime and anywhere. An overall statistical chart with the mentioned values is presented in the application, updating from time to time. With this pet care system in a smartphone application, we provide pet owners an efficient, convenient and low-cost tool for pet care.[8]

Chapter 3

PROBLEM IDENTIFICATION AND OBJECTIVES

3.1 PROBLEMS

- Now a days we can see various problems faced by a pet when the owner leaves the pet alone in the home for long time.
- Because of this pets can't able to get the food on time and they become weak and sick.
- We can't able to monitor the health of the pets.

The cost is more because the owners will appoint the pet sitters to take care of the pets when they are far away from the home.

3.2 The objectives /Aim of this project are:

- To create an automatic feeding machine to feed food and water for pets.
- Own Module Designed to take care of pet in absence of owner for long duration.
- Use of IoT based wearable devices for the pet to monitor the health parameter and it includes GPS system.
- Smart Sensors are utilized to monitor and maintain hygienic environment. Mobile app is developed to automate the pet shelter.

Chapter 4

METHODOLOGY

4.1 PROPOSED BLOCK DIAGRAM

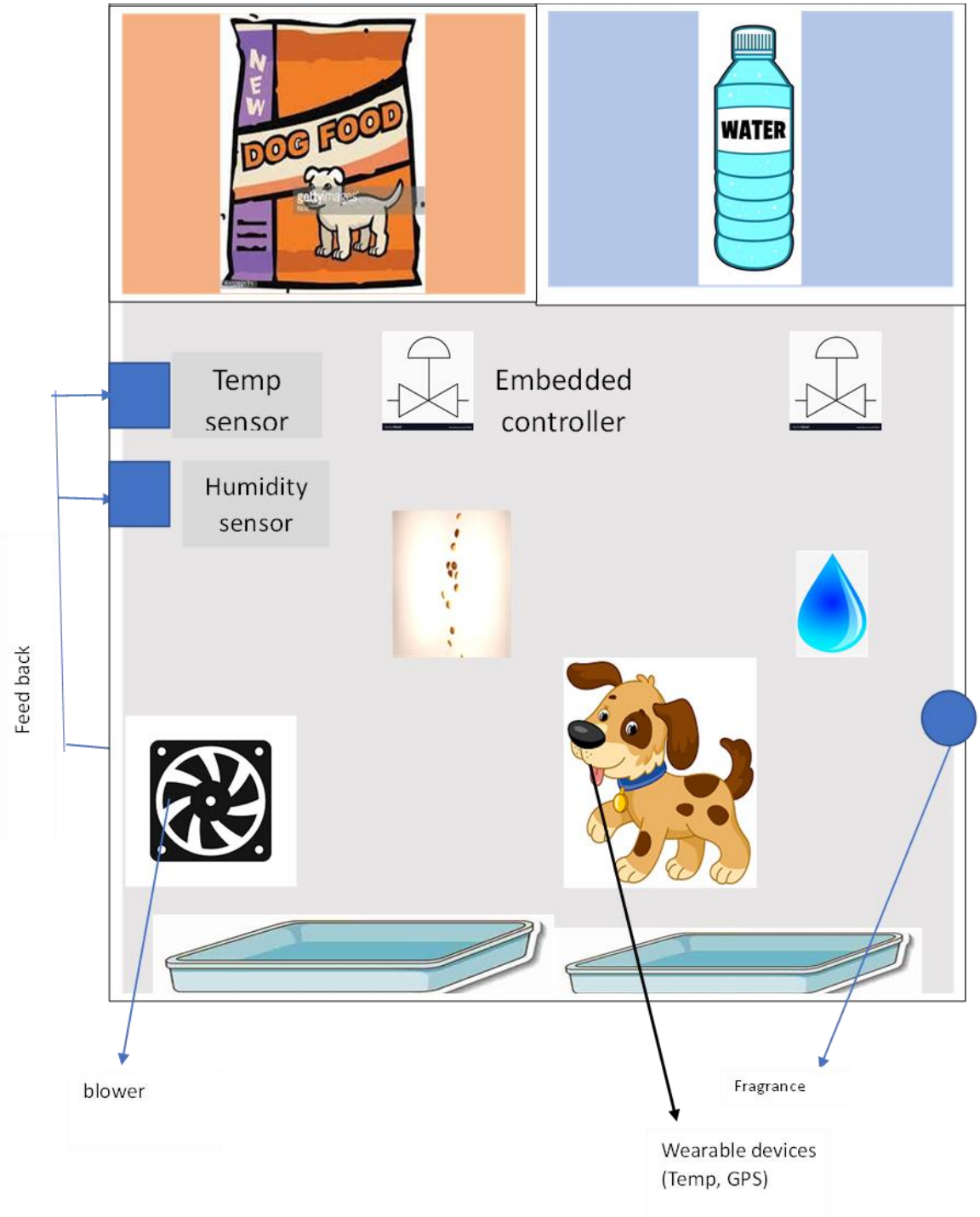

Fig 4.1 Proposed Block Diagram

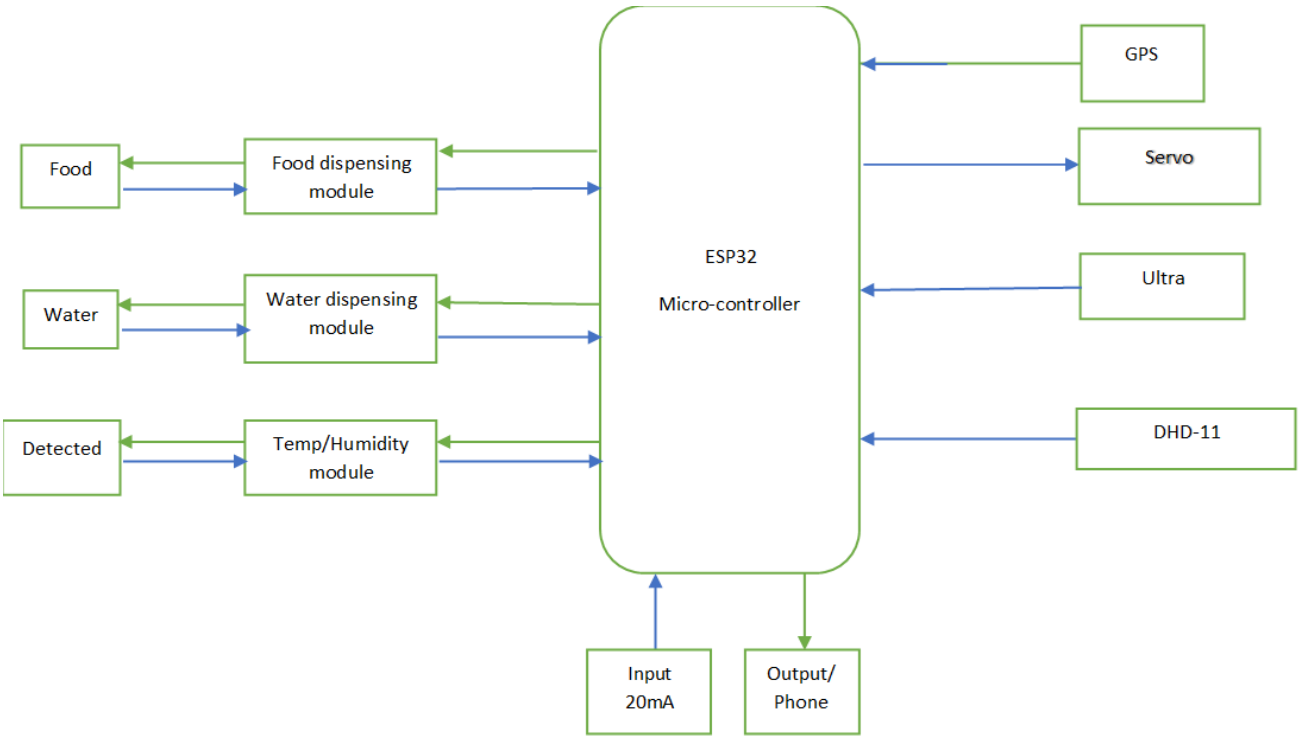

Fig 4.2 Block Diagram

Wearable device: collar belt → GPS, heartbeat and Spo2

Sensors: temperature sensors, humidity sensors, which is used to sense the external parameters.

IoT: internet of things connecting devices and mobiles through Internet

Food and Water Dispenser: food dispensers and water dispensers, which dispensers required amount of food and water

Controller: Embedded controller, control valve used to control dispensers.

Chapter 5

IMPLEMENTATION

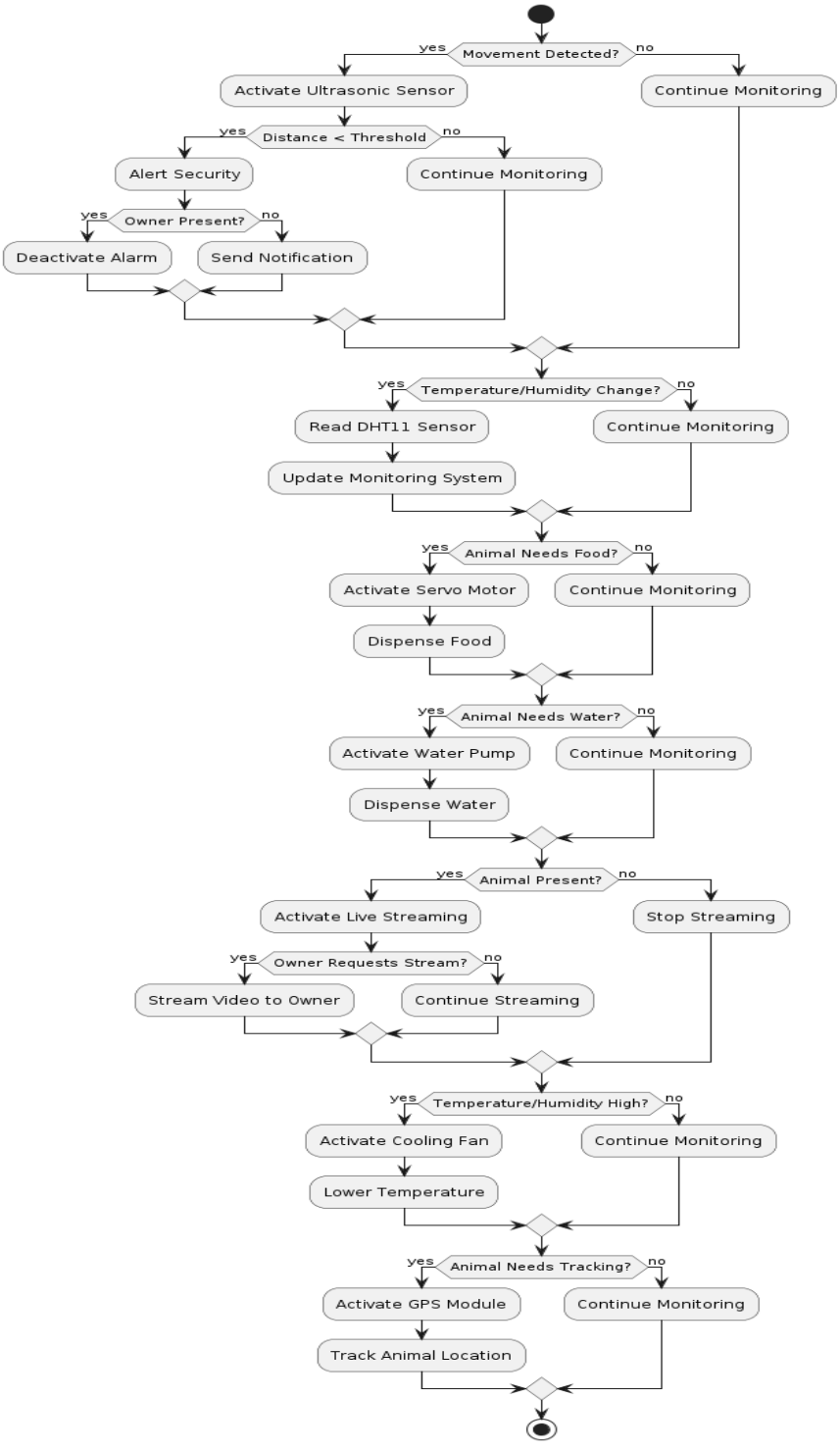

Fig 5.1 Flowchart of pet shelter automation

This flowchart represents the following functionalities:

Monitoring for movement using an ultrasonic sensor.

Alerting security in case of unauthorized movement.

Notifying the owner if movement is detected and they are not present.

Monitoring temperature and humidity changes using a DHT11 sensor.

Dispensing food automatically when needed using a servo motor.

Dispensing water automatically when needed using a water pump.

Activating live streaming using an ESP32 cam if an animal is present.

Streaming video to the owner upon request.

Activating a cooling fan if the temperature or humidity is high.

Tracking the location of an animal using GPS.

This flowchart outlines the logical flow of operations in a pet shelter with various automated functionalities to ensure the well-being and security of the animals housed within it.

Detailed Explanation:

1. Hardware Setup:

- Servo Motor: Implement a servo motor-based food dispenser mechanism to dispense pet food at scheduled intervals.
- Water Pump: Utilize a water pump system for automated water dispensing to ensure continuous access to fresh water for pets.
- ESP32-CAM: Install ESP32-CAM modules for live streaming video feed from the pet shelter, allowing remote monitoring of pets.
- DHT11 Sensor: Integrate DHT11 temperature and humidity sensors to monitor environmental conditions within the pet shelter and ensure optimal comfort for pets.
- DC Fan: Incorporate DC fans for cooling purposes to regulate temperature and maintain a comfortable environment, especially during hot weather.
- Ultrasonic Sensor: Install ultrasonic sensors for movement detection to monitor pet activity and trigger alerts in case of unusual behavior or distress.
- GPS Module: Integrate GPS modules for tracking and locating pets within the shelter premises, enabling efficient management and retrieval.

2. Food Dispensing System:

- Design a food dispenser mechanism controlled by the servo motor to release predefined portions of pet food at scheduled intervals.
- Implement scheduling logic to automate food dispensing based on predefined feeding times and portion sizes, ensuring regular and controlled feeding for pets.
- Integrate feedback mechanisms to monitor food levels in the dispenser and trigger alerts when refills are required, ensuring continuous availability of food.

3. Water Dispensing System:

- Develop a water dispensing system using the water pump to provide pets with a constant supply of fresh water.
- Implement sensors or timers to regulate water flow and prevent overflow or wastage, ensuring efficient water dispensing while minimizing resource consumption.

4. Live Streaming and Monitoring:

- Configure ESP32-CAM modules to capture live video streams from different areas of the pet shelter, providing real-time surveillance and monitoring capabilities.
- Develop a web-based interface or mobile application to access the live video feed remotely, allowing pet owners or shelter staff to monitor pets' activities and well-being from anywhere.

5. Temperature and Humidity Monitoring:

- Connect DHT11 sensors to the ESP32-CAM modules to measure temperature and humidity levels within the pet shelter.
- Implement data logging and visualization features to monitor environmental conditions over time and identify trends or anomalies that may affect pet health and comfort.

6. Cooling System:

- Install DC fans strategically within the pet shelter to regulate temperature and improve air circulation, especially during hot weather conditions.
- Implement temperature control algorithms to adjust fan speed based on ambient temperature measurements, maintaining optimal conditions for pets' well-being.

7. Movement Detection and Alerts:

- Deploy ultrasonic sensors at key locations within the pet shelter to detect pet movements and activity levels.
- Set up alerting mechanisms to notify shelter staff or pet owners in real-time of any unusual behavior or signs of distress observed through movement detection sensors.

8. GPS Tracking:

- Attach GPS modules to pet collars or harnesses to track their location and movements within the shelter premises.

- Develop a centralized tracking system to monitor pet whereabouts, ensuring efficient management, and facilitating quick retrieval in case of escape or emergencies.

5.1 Hardware Implementation

Component specification and working:

5.1.1 ESP- WROOM-32

Fig 5.2 ESP-Wroom-32

ESP32-WROOM-32 is a powerful, generic Wi-Fi + Bluetooth® + Bluetooth LE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding. At the core of this module is the ESP32-D0WDQ6 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz. The chip also has a low-power coprocessor that can be used instead of the CPU to save power while performing tasks that do not require much computing power, such as monitoring of peripherals. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S, and I2C.

The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is all-around: using Wi-Fi allows a large physical range and direct connection to the Internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep

current of the ESP32 chip is less than 5 µA, making it suitable for battery powered and wearable electronics applications. The module supports a data rate of up to 150 Mbps, and 20 dBm output power at the antenna to ensure the widest physical range The operating system chosen for ESP32 is free RTOS with L w IP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that users can upgrade their products even after their release, at minimum cost and effort.

5.1.2 Dht11 sensor

Fig 5.3 DHT11

DHT11 output calibrated digital signal. It applies exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with8-bit single-chip computer. Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of program in OTP memory, when the sensor is detecting, it will cite co-efficient from memory. Small size & low consumption & long transmission distance(100m) enable DHT11 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

5.1.3 DC cooling Fan

Fig 5.4 DC Cooling Fan

A cooling fan is widely used to extend life of your system by cooling off heat of the system that many electrical components are mounted in a very high density and dissipating heat. Since we SANYO DENKI developed “San Ace” which is the first AC fan in Japan in 1965, we have increased fan motor lineup until now meeting customer’s needs rapidly based on our tremendous career. We SANYO DENKI will continue to develop new fans with high airflow, low noise, low vibration, and energy saving design.

5.1.4 Servo motor

Fig 5.5. Servo motor

SG90 9 g Micro Servo Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware. Specifications

• Weight: 9 g

• Dimension: 22.2 x 11.8 x 31 mm approx.

• Stall torque: 1.8 kg f cm

• Operating speed: 0.1 s/60 degree

5.1.5 ULTRA SONIC SENSOR

Fig 5.6 Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit.

The basic principle of work:

(1) Using IO trigger for at least 10us high level signal,

(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.

(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Table 4.1

Working of Ultrasonic Sensor

The ultrasonic sensor (or transducer) works on the same principles as a radar system. An ultrasonic sensor can convert electrical energy into acoustic waves and vice versa. The acoustic wave signal is an ultrasonic wave traveling at a frequency above 18kHz. The famous HC SR04 ultrasonic sensor generates ultrasonic waves at 40kHz frequency.

Typically, a microcontroller is used for communication with an ultrasonic sensor. To begin measuring the distance, the microcontroller sends a trigger signal to the ultrasonic sensor. The duty cycle of this trigger signal is 10µS for the HC-SR04 ultrasonic sensor. When triggered, the ultrasonic sensor generates eight acoustic (ultrasonic) wave bursts and initiates a time counter. As soon as the reflected (echo) signal is received, the timer stops. The output of the ultrasonic sensor is a high pulse with the same duration as the time difference between transmitted ultrasonic bursts and the received echo signal.

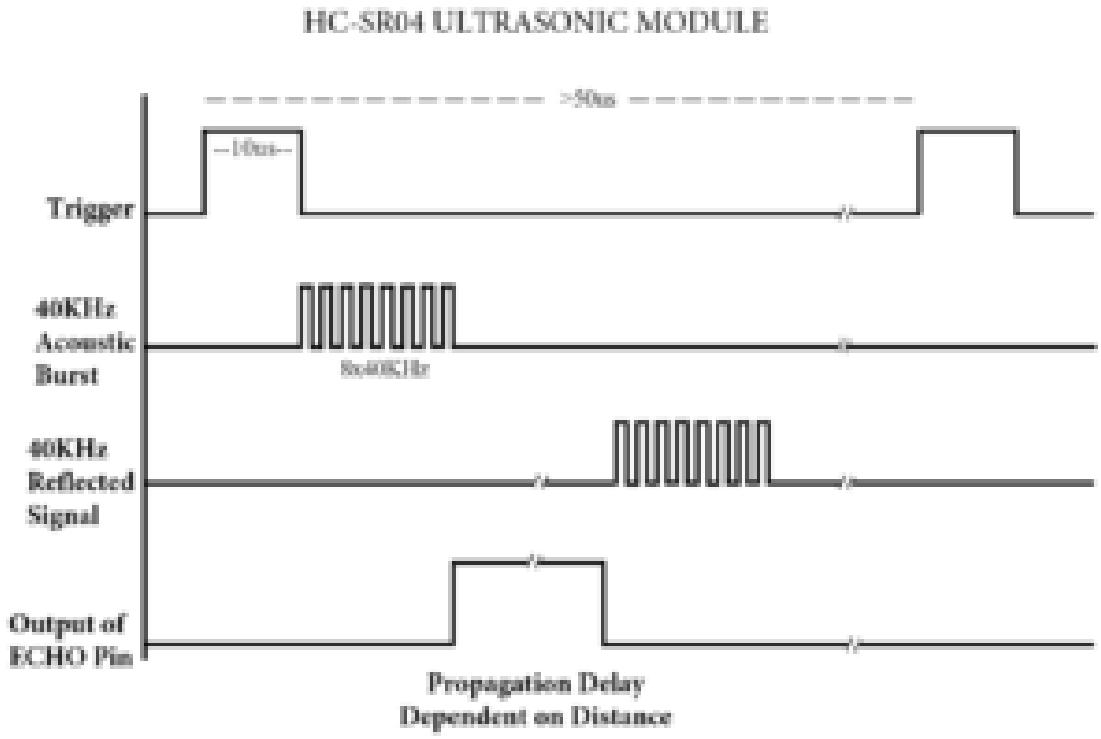

Fig 5.7 Representation of trigger signal, acoustic bursts, reflected signal and output of echo pin.

The microcontroller interprets the time signal into distance using the following functions:

$$Distance\ (cm) = \frac{echo\ pulse\ width\ (uS)}{58}$$

$$Distance\ (inch) = \frac{echo\ pulse\ width\ (uS)}{148}$$

Theoretically, the distance can be calculated using the TRD (time/rate/distance) measurement formula. Since the calculated distance is the distance traveled from the ultrasonic transducer to the object—and back to the transducer—it is a two-way trip. By dividing this distance by 2, you can determine the actual distance from the transducer to the object. Ultrasonic waves travel at the speed of sound (343 m/s at 20°C). The distance between the object and the sensor is half of the distance traveled by the sound wave.[iv] The following equation calculates the distance to an object placed in front of an ultrasonic sensor:

$$distance = \frac{time\ taken\ x\ speed\ of\ sound}{2}$$

Limitation of Ultrasonic Sensor

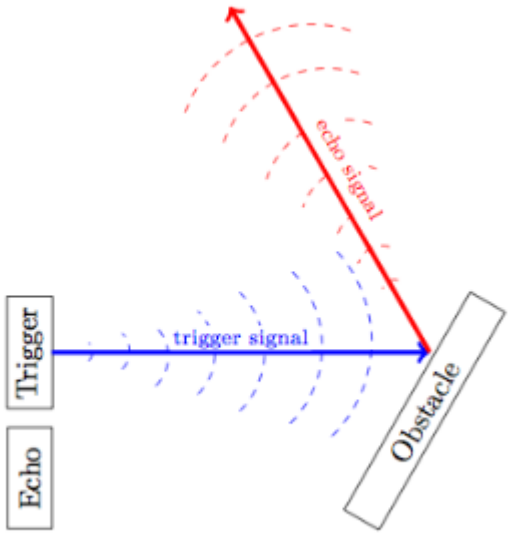

Fig 5.8 Representation of ultrasonic signal deflected due to target object's position, resulting in error

The intense sensitivity of ultrasonic sensors makes them efficient, but that sensitivity can also cause problems. Ultrasonic sensors can detect false signals coming from the airwaves disturbed by an air conditioning system and a pulse coming from a ceiling fan, for instance.

Ultrasonic sensors can detect objects placed within their range, but they cannot distinguish between different shapes and sizes. However, one can overcome this limitation can by using two sensors instead of just one sensor. One can install both sensors a distance away from each other, or they can be adjacent. By observing the overlapped shaded region, one can get a better idea of the shape and size of the target object.

Ultrasonic transducers Ultrasonic transducers are used to emit the ultrasonic waves with high frequency. These transducers are operated by using electrical current. In this project 12V battery is used switch on the source.

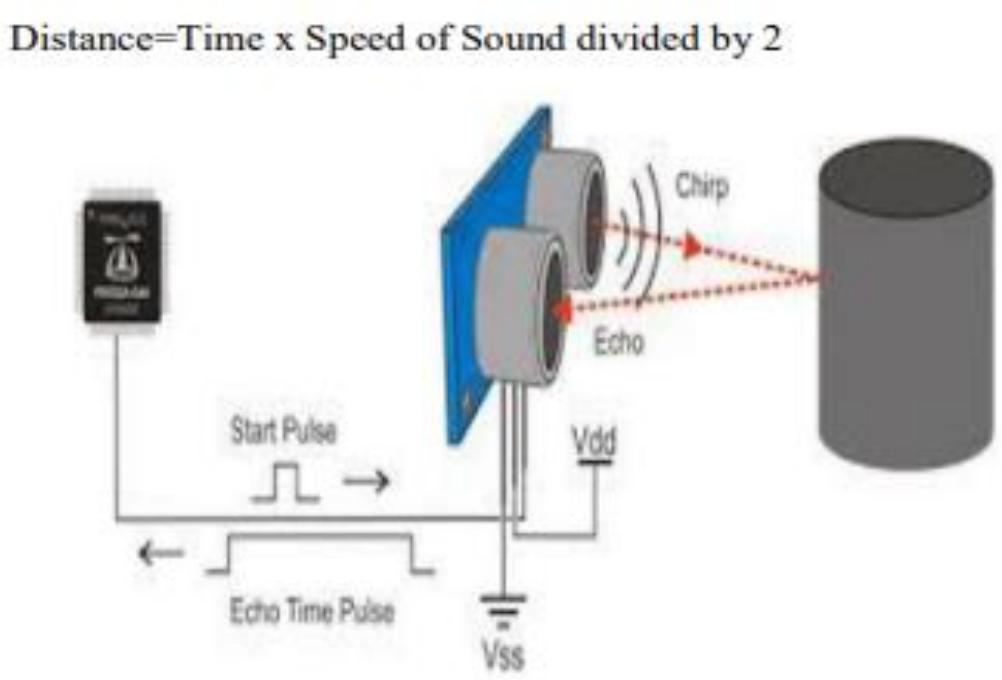

Fig 5.9 Working of ultrasonic sensor

Time = the time between when an ultrasonic wave is transmitted and when it is received you divide this number by 2 because the sound wave has to travel to the object and back.

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent range accuracy and stable readings in an easy-to-use package. It operation is not affected by sunlight or black material like Sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). Similar in performance to the SRF005 but with the low-price of a Sharp infrared sensor.

Power Supply : 5V DC

Quiescent Current : <2mA

Effectual Angle: <15°

Ranging Distance : 2cm – 500 cm/1" - 16ft

Resolution : 0.3 cm

Application:

• Applications ranging occasions;

• measuring the distance between objects:

• Programmable car obstacle avoidance:

• robot obstacle avoidance:

• teaching apparatus;

• security, industrial control

5.1.6 MAX30100:

The device has two LEDs, one emitting a red light, another emitting infrared light. For pulse rate, only the infrared light is needed. Both the red light and infrared light is used to measure oxygen levels in the blood. More on that later.

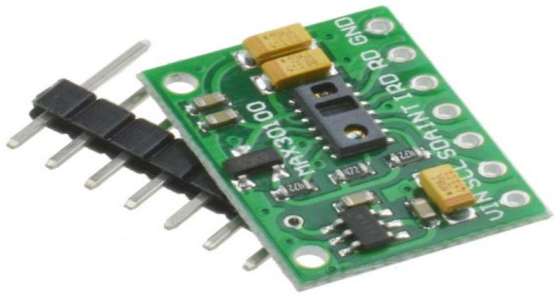

Fig 5.10 MAX30100

When the heart pumps blood, there is an increase in oxygenated blood as a result of having more blood. As the heart relaxes, the volume of oxygenated blood also decreases. By knowing the time between the increase and decrease of oxygenated blood, the pulse rate is determined.

It turns out, oxygenated blood absorbs more infrared light and passes more red light while deoxygenated blood absorbs red light and passes more infrared light. This is the main function of the MAX30100: it reads the absorption levels for both light sources and stored them in a buffer that can be read via I2C.

Purple MAX30100 Module	Arduino UNO/Nano
VIN	5V
GND	GND
SCL	A5
SDA	A4
INT	D2

The chip has an integrated optical sensor that derives its reading from emitting two wavelength of light from the two LED's then measures the absorbance of pulsing blood through a photodetector. The signal is processed by a low noise analog signal processing unit and communicated to the Microcontroller through the i2C Interface.

The MAX30100 operates from 1.8v and 3.3v voltage input and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times. The device is suitable for wearable devices like smart watch, medical monitoring equipment's, fitness assistant and smart suits. Required Components Arduino Microcontroller, ESP8266 (Arduino IDE Integrated), Teensy MCU (TeensyDuino Integrated),Buzzer / Alarm (Optional)LCD / OLED i2C Display (Optional)Solder Less BreadboardJumper Wire.

5.1.7. Water Pump

Submersible Pumps are efficient for pumping out septic tanks. Fluid is transferred into hoses to storage tanks and taken to a treatment facility. Submersible pumps are often used to pump excess water from work sites or flooded basements on construction sites. They can also be used to pump slurries Submersible pumps are centrifugal pumps whose hydraulic components (pump casing, impeller, diffuser element) are flooded by the fluid handled. Usually, this type of pump is not fitted with a suction line. A submersible pump whose motor is arranged above the floor is referred to as a vertical shaft submersible pump

Using attention:

1. The pump does not have a power supply, it is a DC pump. It cannot be used directly with AC 220V voltage.
2. Land use need to allow water to flow into the pump (nonself priming pump), the installation of water level higher than the pump.
3. To keep the water clean, so as not to block the pump rotor. the pump should be cleaned regularly to keep the pump clean.

Features:

100% brand new, high quality

Type: Horizontal Mute Sounds Mini Submersible Pump

Material: engineering plastic

Drive mode: DC brushless design, magnetic drive

500 hours of continuous working life

5.1.8.Nodemcu Wifi Module:

The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, Espressif Systems (A Shanghai-based Semiconductor Company) has released an adorable, bite-sized WiFi enabled microcontroller – ESP8266, at an unbelievable price! For less than $3, it can monitor and control things from anywhere in the world – perfect for just about any IoT project.

ESP-12E Module

The development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS.

ESP-12E Chip

- Tensilica Xtensa® 32-bit LX106
- 80 to 160 MHz Clock Freq.
- 128kB internal RAM
- 4MB external flash
- 802.11b/g/n Wi-Fi transceiver

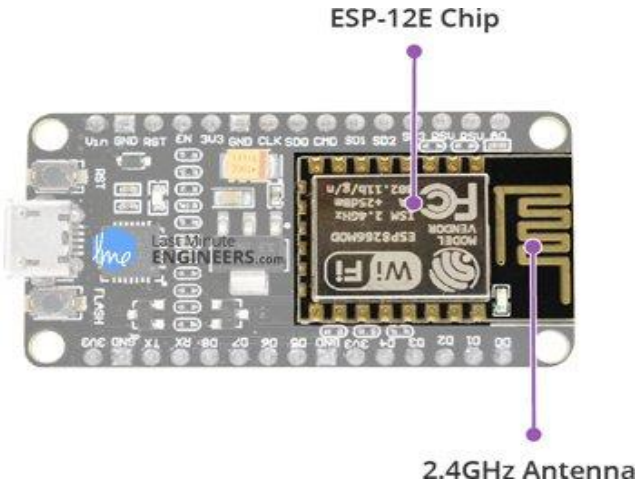

Fig 5.11 Node MCU WIFI module

There's also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile.

Power Requirement

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

Power Requirement

- Operating Voltage: 2.5V to 3.6V
- On-board 3.3V 600mA regulator
- 80mA Operating Current
- 20 µA during Sleep Mode

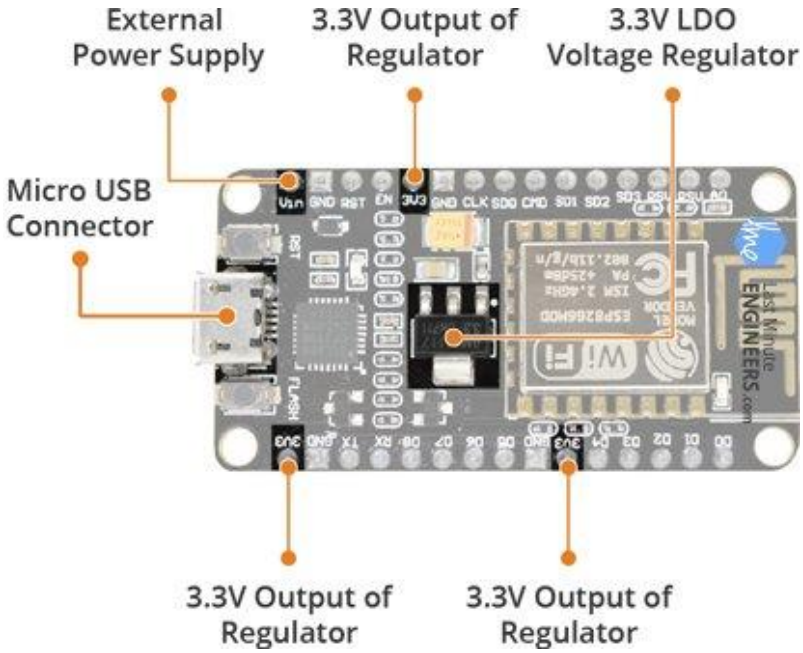

Fig 5.12 ESP8266

Power to the ESP8266 NodeMCU is supplied via the on-board MicroB USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin can be used to directly supply the ESP8266 and its peripherals.

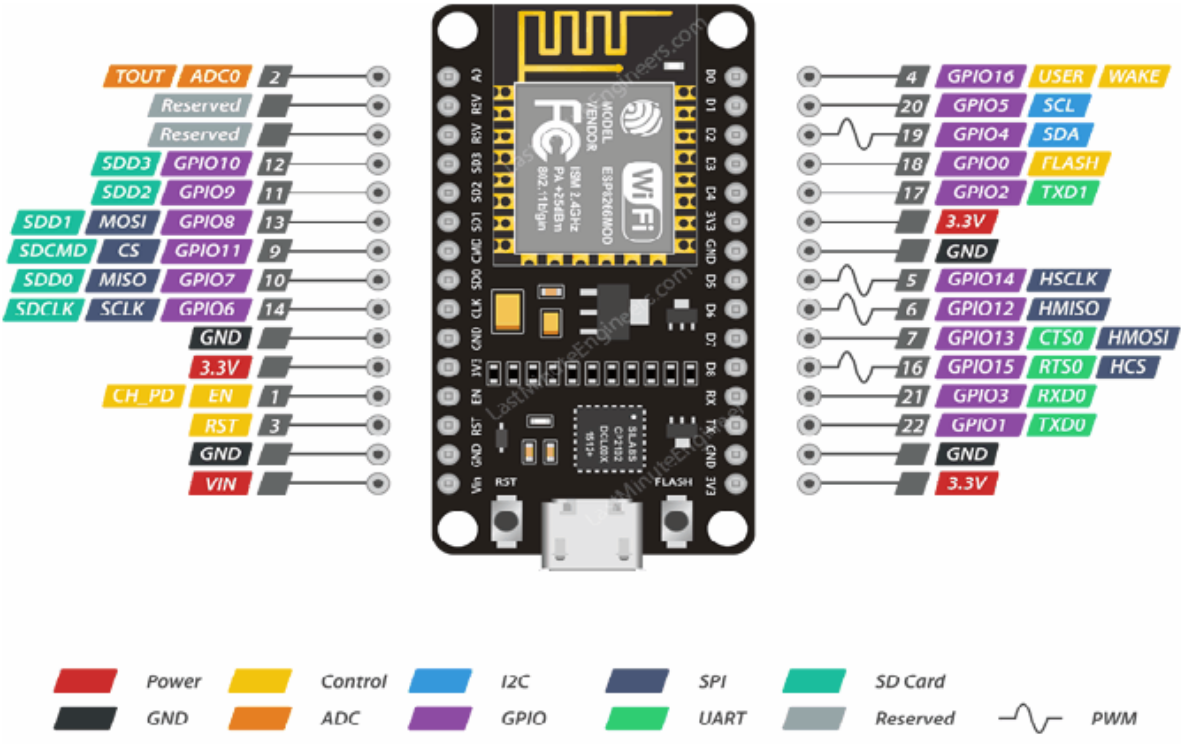

Fig 5.13 ESP8266 pin diagram

5.1.9. GPS

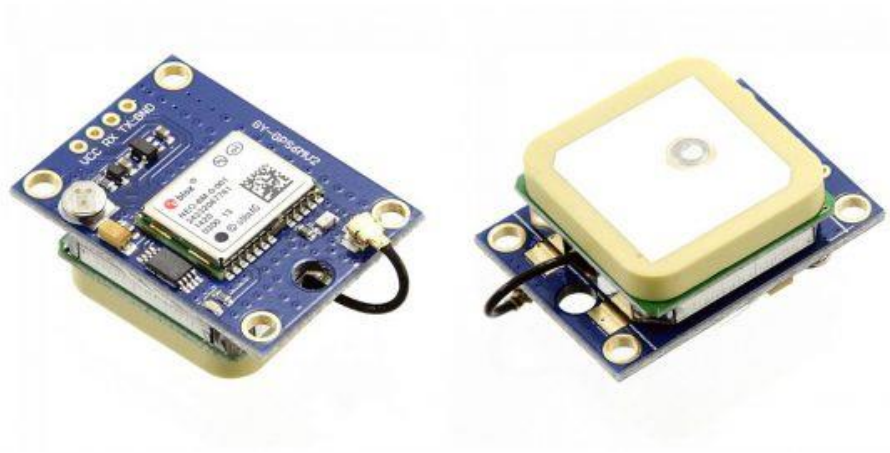

Fig 5.14 GPS Module

As indicated, the GPS modules are based on the u-blox NEO-6M GPS engine. The type number of the NEO-6M is NEO-6M-0-001, and its ROM/FLASH version is ROM 7.0.3 (PCN reference UBX-TN-11047-1). The NEO-6M module includes one configurable UART interface for serial communication, but the default UART (TTL) baud rate here is 9,600. Because the GPS signal is

right-hand circular-polarized (RHCP), the style of the GPS antenna will be different from the common whip antennas used for linear polarized signals. The most popular antenna type is the patch antenna. Patch antennas are flat, generally have a ceramic and metal body, and are mounted on a metal base plate. They are often cast in a housing. For more information about u-blox reference designs, see their **website**. Remember, the position of the antenna mounting is very crucial for optimal performance of the GPS receiver. When using the patch antenna, it should be oriented parallel to the geographic horizon. The antenna must have full view of the sky, ensuring a direct line of sight with as many visible satellites as possible.

Initial test setup

For a quick test using your Windows computer, you just need to establish a serial communication with the GPS module using one USB-UART adapter like the PL2303 USB-to-Serial Converter module. The hardware setup is pretty simple:

NEO-6M GPS Module	USB-to-Serial Converter
TX	RX
RX	TX
GND	GND
VCC	5 V

Table 5.3 Initial test setup

Note that when the GPS module works, the green indicator on the GPS module will blink (the red one is for power-on indication), and the figures regarding the time, latitude, longitude, etc., will be displayed in the u-center software window. Finally, you should compare the figures shown by the software with the data collected by another trusty GPS device to ensure that your NEO-6M GPS module is perfect (and precise) in all respects.

5.1.10 Telegram Bot:

TELEGRAM BOTS

Bots: An introduction for developers

Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and inline requests. You control your bots using HTTPS requests to our Bot API.

What can I do with bots?

A chat with @TechCrunchBot also showing search results from the @gif inline-bot

- To name just a few things, you could use bots to:
- Get customized notifications and news. A bot can act as a smart newspaper, sending you relevant content as soon as it's published.
- Integrate with other services. A bot can enrich Telegram chats with content from external services.
- Gmail Bot, GIF bot, IMDB bot, Wiki bot, Music bot, Youtube bot, GitHub bot
- Accept payments from Telegram users. A bot can offer paid services or work as a virtual storefront.
- Demo Shop Bot
- Create custom tools. A bot may provide you with alerts, weather forecasts, translations, formatting or other services.
- Markdown bot, Sticker bot, Vote bot, Like bot
- Build single- and multiplayer games. A bot can offer rich HTML5 experiences, from simple arcades and puzzles to 3D-shooters and real-time strategy games.
- GameBot, Gamee
- Build social services. A bot could connect people looking for conversation partners based on common interests or proximity.
- Do virtually anything else. Except for dishes — bots are terrible at doing the dishes.

How do bots work?

At the core, Telegram Bots are special accounts that do not require an additional phone number to set up. Users can interact with bots in two ways:

Send messages and commands to bots by opening a chat with them or by adding them to groups. This is useful for chat bots or news bots like the official TechCrunch bot.

Send requests directly from the input field by typing the bot's @username and a query. This allows sending content from inline bots directly into any chat, group or channel.

Messages, commands and requests sent by users are passed to the software running on your servers. Our intermediary server handles all encryption and communication with the Telegram API for you. You communicate with this server via a simple HTTPS-interface that offers a simplified version of the Telegram API. We call that interface our Bot API.

How do I create a bot?

Fig 5.15 The Botfather -Telegram Bot

The Botfather.

There's a… bot for that. Just talk to BotFather (described below) and follow a few simple steps. Once you've created a bot and received your authorization token, head down to the Bot API manual to see what you can teach your bot to do.

5.2 Software Implementation:

5.2.1 Arduino IDE

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

The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

• Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

• Simple, clear programming environment - The Arduino Software (IDE) is easy-touse for beginners, yet flexible enough for advanced users to take advantage of as well.

• Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

• Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

5.2.2 Embedded C:

- When designing software for a smaller embedded system with the 8051, it is very common place to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in nature. If a hardware engineer is tasked with designing both the hardware and the software, he or she will frequently be tempted to write the software in assembly language.
- The trouble with projects done with assembly code can is that they can be difficult to read and maintain, especially if they are not well commented. Additionally, the amount of code reusable from a typical assembly language project is usually very low. Use of a higher-level language like C can directly address these issues. A program written in C is easier to read than an assembly program.
- Since a C program possesses greater structure, it is easier to understand and maintain. Because of its modularity, a C program can better lend itself to reuse of code from project to project. The division of code into functions will force better structure of the software and lead to functions that can be taken from one project and used in another, thus reducing overall development time. A high order language such as C allows a developer to write code, which resembles a human's thought process more closely than does the equivalent assembly code. [25]The developer can focus more time on designing the algorithms of the system rather than having to concentrate on their individual implementation. This will greatly reduce development time and lower debugging time since the code is more understandable.
- By using a language like C, the programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly. Many target processors have C compilers available, which support ANSI C.
- All of this is not to say that assembly language does not have its place. In fact, many embedded systems (particularly real time systems) have a combination of C and assembly code. For time critical operations, assembly code is frequently the only way to go. One of

the great things about the C language is that it allows you to perform low-level manipulations of the hardware if need be, yet provides you the functionality and abstraction of a higher order language.

Arduino - Installation

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

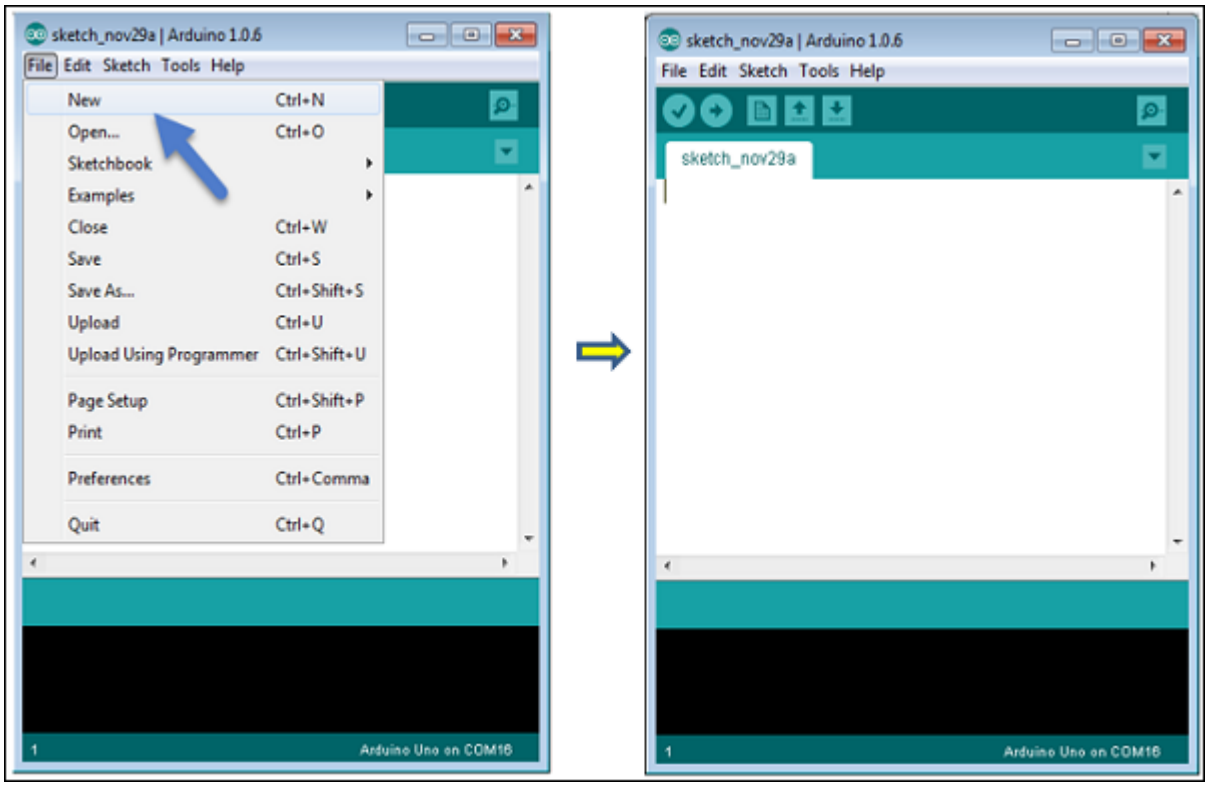

Fig 5.16 creating new file

To open an existing project example, select File → Example → Basics → Blink.

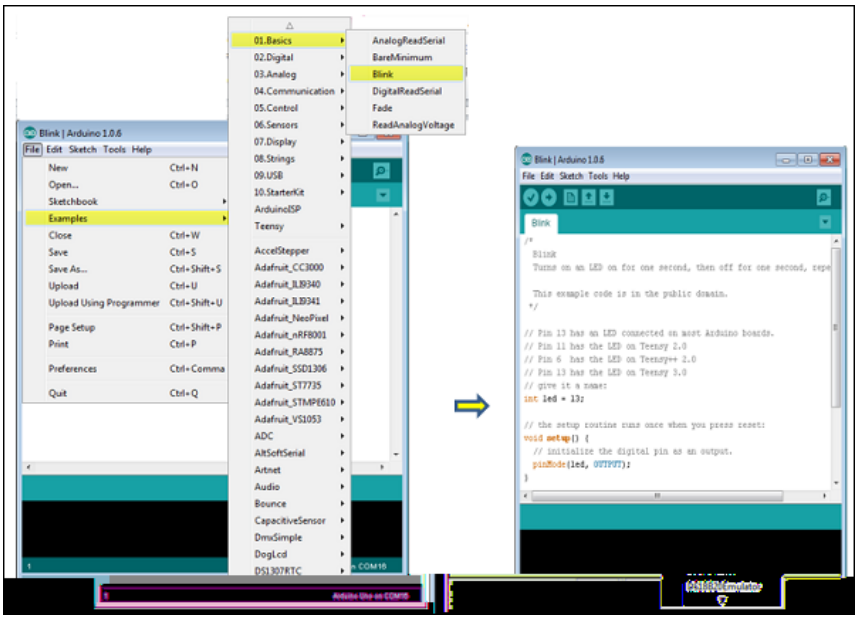

Fig 5.17 selecting board

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

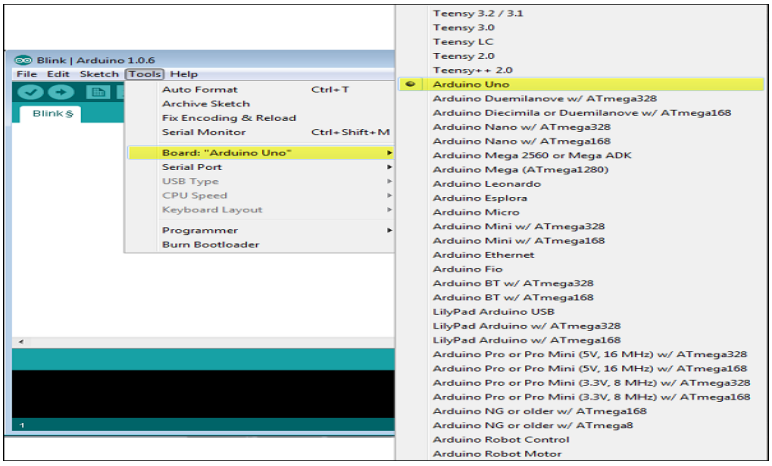

Fig 5.18 selecting Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

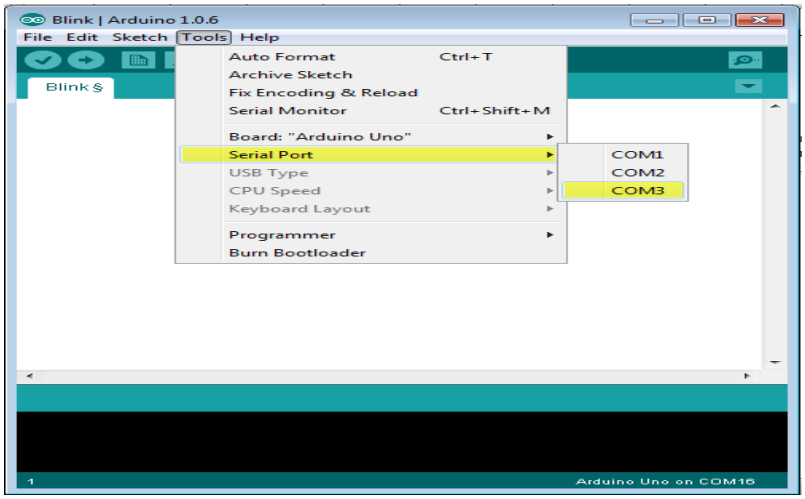

Fig 5.19 selecting serial port

Embedded C:

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems.

Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

What is an Embedded System?

An Embedded System can be best described as a system which has both the hardware and software and is designed to do a specific task. A good example for an Embedded System, which many households have, is a Washing Machine.

Embedded Systems can not only be stand-alone devices like Washing Machines but also be a part of a much larger system. An example for this is a Car. A modern day Car has several individual embedded systems that perform their specific tasks with the aim of making a smooth and safe journey.

Some of the embedded systems in a Car are Anti-lock Braking System (ABS), Temperature Monitoring System, Automatic Climate Control, Tyre Pressure Monitoring System, Engine Oil Level Monitor, etc.

Programming Embedded Systems

As mentioned earlier, Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array).

All these devices have one thing in common: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to define how the device actually works.

Embedded Software or Program allow Hardware to monitor external events (Inputs) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc.

From the above statement, it is clear that the Software part of an Embedded System is equally important to the Hardware part. There is no point in having advanced Hardware Components with poorly written programs (Software).

There are many programming languages that are used for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA (high-level programming languages), Visual Basic, JAVA Script (Application level Programming Languages), etc.

In the process of making a better embedded system, the programming of the system plays a vital role and hence, the selection of the Programming Language is very important.

Factors for Selecting the Programming Language

The following are few factors that are to be considered while selecting the Programming Language for the development of Embedded Systems.

- **Size:** The memory that the program occupies is very important as Embedded Processors like Microcontrollers have a very limited amount of ROM.
- **Speed:** The programs must be very fast i.e. they must run as fast as possible. The hardware should not be slowed down due to a slow running software.

- **Portability:** The same program can be compiled for different processors.
- **Ease of Implementation**
- **Ease of Maintenance**
- **Readability**

Earlier Embedded Systems were developed mainly using Assembly Language. Even though Assembly Language is closest to the actual machine code instructions, the lack of portability and high amount of resources spent on developing the code, made the Assembly Language difficult to work with.

There are other high-level programming languages that offered the above mentioned features but none were close to C Programming Language.

Introduction to Embedded C Programming Language

Before going in to the details of Embedded C Programming Language and basics of Embedded C Program, we will first talk about the C Programming Language.

The C Programming Language, developed by Dennis Ritchie in the late 60's and early 70's, is the most popular and widely used programming language. The C Programming Language provided low level memory access using an uncomplicated compiler (a software that converts programs to machine code) and achieved efficient mapping to machine instructions.

The C Programming Language became so popular that it is used in a wide range of applications ranging from Embedded Systems to Super Computers.

Embedded C Programming Language, which is widely used in the development of Embedded Systems, is an extension of C Program Language. The Embedded C Programming Language uses the same syntax and semantics of the C Programming Language like main function, declaration of datatypes, defining variables, loops, functions, statements, etc.

The extension in Embedded C from standard C Programming Language include I/O Hardware Addressing, fixed point arithmetic operations, accessing address spaces, etc.

Difference between C and Embedded C

There is actually not much difference between C and Embedded C apart from few extensions and the operating environment. Both C and Embedded C are ISO Standards that have almost same syntax, datatypes, functions, etc.

Embedded C is basically an extension to the Standard C Programming Language with additional features like Addressing I/O, multiple memory addressing and fixed-point arithmetic, etc.

C Programming Language is generally used for developing desktop applications whereas Embedded C is used in the development of Microcontroller based applications.

Basics of Embedded C Program

Now that we have seen a little bit about Embedded Systems and Programming Languages, we will dive in to the basics of Embedded C Program. We will start with two of the basic features of the Embedded C Program: Keywords and Datatypes.

Keywords in Embedded C

A Keyword is a special word with a special meaning to the compiler (a C Compiler for example, is a software that is used to convert program written in C to Machine Code). For example, if we take the Keil's Cx51 Compiler (a popular C Compiler for 8051 based Microcontrollers) the following are some of the keywords:

- bit
- sbit
- sfr
- small
- large

These are few of the many keywords associated with the Cx51 C Compiler along with the standard C Keywords.

Data Types in Embedded C

Data Types in C Programming Language (or any programming language for that matter) help us declaring variables in the program. There are many data types in C Programming Language like signed int, unsigned int, signed char, unsigned char, float, double, etc. In addition to these there few more data types in Embedded C.

The following are the extra data types in Embedded C associated with the Keil's Cx51 Compiler.

- bit
- sbit
- sfr
- sfr16

The following table shows some of the data types in Cx51 Compiler along with their ranges.

Data Type *Bits (Bytes)* *Range*

bit	1	0 or 1 (bit addressable part of RAM)
signed int	16 (2)	-32768 to +32767
unsigned int	16 (2)	0 to 65535
signed char	8 (1)	-128 to +127
unsigned	8 (1)	0 to 255
float	32 (4)	±1.175494E-38 to ±3.402823E+38
double	32 (4)	±1.175494E-38 to ±3.402823E+38
sbit	1	0 or 1 (bit addressable part of RAM)
sfr	8 (1)	RAM Addresses (80h to FFh)
sfr16	16 (2)	0 to 65535

Basic Structure of an Embedded C Program (Template for Embedded C Program)

The next thing to understand in the Basics of Embedded C Program is the basic structure or Template of Embedded C Program. This will help us in understanding how an Embedded C Program is written.

The following part shows the basic structure of an Embedded C Program.

- Multiline Comments . . . . . Denoted using /*……*/
- Single Line Comments . . . . . Denoted using //
- Preprocessor Directives . . . . . #include<…> or #define
- Global Variables . . . . . Accessible anywhere in the program
- Function Declarations . . . . . Declaring Function
- Main Function . . . . . Main Function, execution begins here

  {

  Local Variables . . . . . Variables confined to main function

  Function Calls . . . . . Calling other Functions

  Infinite Loop . . . . . Like while(1) or for(;;)

  Statements . . . . . .

  ….

  ….

  }

- Function Definitions . . . . . Defining the Functions
  {
  Local Variables . . . . . Local Variables confined to this Function
  Statements . . . . . .
  ….
  ….
  }

Different Components of an Embedded C Program

Comments: Comments are readable text that are written to help us (the reader) understand the code easily. They are ignored by the compiler and do not take up any memory in the final code (after compilation).

There are two ways you can write comments: one is the single line comments denoted by // and the other is multiline comments denoted by /*….*/.

Preprocessor Directive: A Preprocessor Directive in Embedded C is an indication to the compiler that it must look in to this file for symbols that are not defined in the program.

In C Programming Language (also in Embedded C), Preprocessor Directives are usually represented using #include… or #define….

In Embedded C Programming, we usually use the preprocessor directive to indicate a header file specific to the microcontroller, which contains all the SFRs and the bits in those SFRs.

In case of 8051, Keil Compiler has the file “reg51.h”, which must be written at the beginning of every Embedded C Program.

Global Variables: Global Variables, as the name suggests, are Global to the program i.e. they can be accessed anywhere in the program.

Local Variables: Local Variables, in contrast to Global Variables, are confined to their respective function.

Main Function: Every C or Embedded C Program has one main function, from where the execution of the program begins.

- When designing software for a smaller embedded system with the 8051, it is very common place to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in

nature. If a hardware engineer is tasked with designing both the hardware and the software, he or she will frequently be tempted to write the software in assembly language.

- The trouble with projects done with assembly code can is that they can be difficult to read and maintain, especially if they are not well commented. Additionally, the amount of code reusable from a typical assembly language project is usually very low. Use of a higher-level language like C can directly address these issues. A program written in C is easier to read than an assembly program.
- Since a C program possesses greater structure, it is easier to understand and maintain. Because of its modularity, a C program can better lend itself to reuse of code from project to project. The division of code into functions will force better structure of the software and lead to functions that can be taken from one project and used in another, thus reducing overall development time. A high order language such as C allows a developer to write code, which resembles a human's thought process more closely than does the equivalent assembly code. [25]The developer can focus more time on designing the algorithms of the system rather than having to concentrate on their individual implementation. This will greatly reduce development time and lower debugging time since the code is more understandable.
- By using a language like C, the programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly. Many target processors have C compilers available, which support ANSI C.
- All of this is not to say that assembly language does not have its place. In fact, many embedded systems (particularly real time systems) have a combination of C and assembly code. For time critical operations, assembly code is frequently the only way to go. One of the great things about the C language is that it allows you to perform low-level manipulations of the hardware if need be, yet provides you the functionality and abstraction of a higher order language.

Chapter 6

RESULT AND DISCUSSION

6.1 SHELTER STRUCTURE AND DESIGN

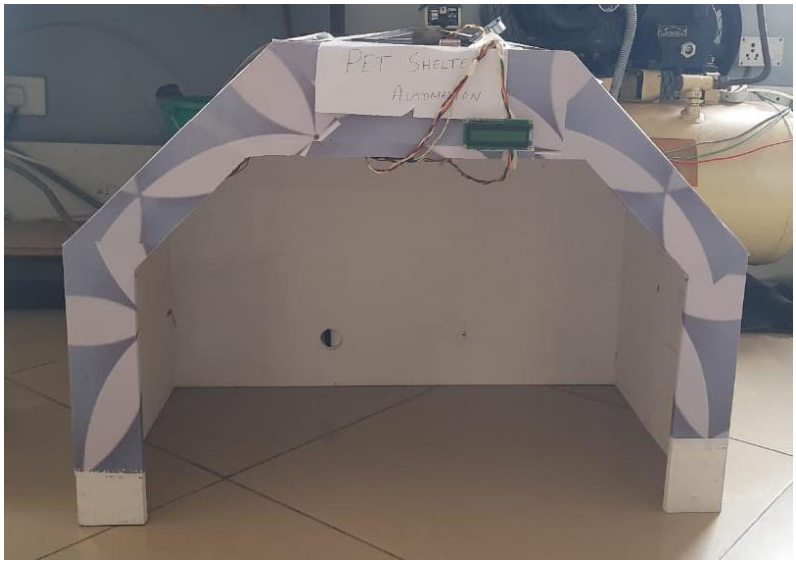

Fig 6.1 Pet Shelter

Shelter is designed in the form of trapezoidal home for dogs and cats. It has built-in temperature sensor which detects the temperature and humidity of the shelter. It has a cooling fan installed which turns ON and OFF as required. It has FOOD and WATER dispensers which dispenses food as required. Pets can sleep in there and spend their time.

6.2 Temperature and Humidity sensing module and controlling results

Temperature and Humidity is sensed and DC Fan is turned on when Temperature is greater than 27 degrees Celsius. It shows nan if dht11 does not sense the temperature.

6.3 Food Dispensing Module

Fig 6.2 Food Dispensing module

Food containing box is attached with a pipe which helps the food to fall into the bowl.

As the servo motor turns ON and starts to run, the food gets dispensed.

6.4 Telegram output

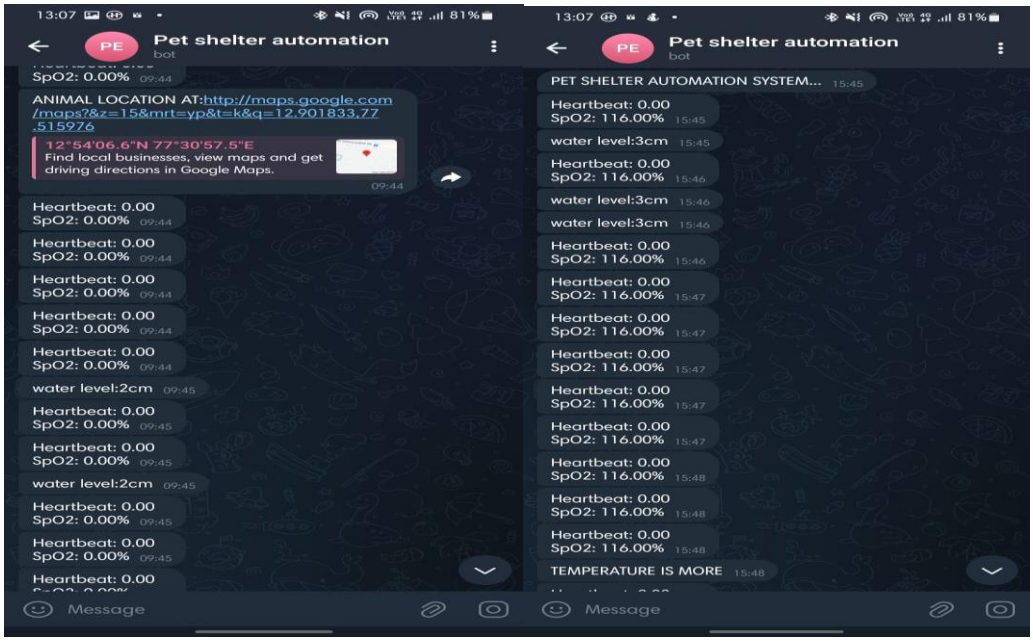

Fig 6.3 Output of telegram chat

6.5 LED Display output

Fig 6.4 LED Display

It provides the display of the output of :

- Temperature and humidity
- Heartbeat and Spo2
- Food is dispensing
- Water level

Chapter 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The interaction between humans and physical devices is attracting attention. Many studies have attempted to provide a natural and intuitive approach to request services.

The current trend of combining pet control and IoT technology offers exciting future developments. The proposed system is also referred on smart-home technology, including the smart pet door and pet feeder.

The results not only present the key improvement of the pet monitor system involved in the IoT technology, but also meet the demand of pet owners. The basic vision behind the IoT, it may have a new way of operational method, it may have a new method of connecting devices, and they might be even completely in clean-slate approach.

As the full operational definition is finalized, but there are numerous research issues that can be worked on. As a next step, we will fully integrate the other pet care devices into our system, including advanced sensors, pet cam, etc.

7.2 FUTURE SCOPE

Our Automatic pet feeder has a lot more things to improve. It can also be used for multiple pets.Multiple devices can be controlled using same application.We have developed only for Android and further it can be developed for iOS aswell.

For further and deeper research, they can put more functionalities in our Automatic Pet feeder like putting an additional camera so that the owner can see or monitor if his/her pet is eating its meal or not.With the help of pet's collar belt we can install GPS system and tracking system.

It can be used not only for dogs but also for other animals like cat aswell.With that, the diverse needs of the owners can be met, and the health, monitor, and location tracking for pets are all covered. Besides, standing as the cloud term, how to connect the numerous networking devices around the globe is the next issue. In the future, we will centralize on the study of the IoT gateway and long-distance detection of the pets.

REFERENCES

[1] Juneyoung Ahn, Jini Kwon, Hyejeong Nam, Hyun-Kook Jang and Jee-In Kim, "Pet buddy: A wearable device for canine behavior recognition using a single IMU," 2016 International Conference on Big Data and Smart Computing (BigComp), Hong Kong, China, 2016, pp. 419-422, doi: 10.1109/BIGCOMP.2016.7425961.

[2] Y. -J. Lin, C. -W. Chuang, C. -Y. Yen, S. -H. Huang and S. -Y. Lee, "Smart Pet Clothing for Monitoring of Health and Mood," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 2018, pp. 1-4, doi: 10.1109/ISCAS.2018.8351547.

[3] C. Sazara and X. Gao, "Predicting Animal Shelter Pet Adoption Times and Feature Importance Analysis using CatBoost," 2022 IEEE 11th International Conference on Intelligent Systems (IS), Warsaw, Poland, 2022, pp. 1-4, doi: 10.1109/IS57118.2022.10019608.

[4] T. Sangvanloy and K. Sookhanaphibarn, "Automatic Pet Food Dispenser by using Internet of Things (IoT)," *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*, Kyoto, Japan, 2020, pp. 132-135, doi: 10.1109/LifeTech48969.2020.1570620257.

[5] E. Jain, S. Badwaik, S. Khirwadkar, S. Thakare, M. Uike and P. H. Chandankhede, "Design of Smart Pet Food Dispenser using Embedded System," 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2023, pp. 1-5, doi: 10.1109/ESCI56872.2023.10100166.

[6] G. M. Debele and X. Qian, "Automatic Room Temperature Control System Using Arduino UNO R3 and DHT11 Sensor," 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 2020, pp. 428-432, doi: 10.1109/ICCWAMTIP51612.2020.9317307.

[7] H. Y. Lin et al., "DC servo speed control of an inkjet print head transport system using a phase-locked loop," Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE, Mie, Japan, 1996, pp. 458-463 vol.2, doi: 10.1109/AMC.1996.509292.

[8] Y. Chen and M. Elshakankiri, "Implementation of an IoT based Pet Care System," 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), Paris, France, 2020, pp. 256-262, doi: 10.1109/FMEC49853.2020.9144910.