VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

FACULTY: Prof. Dheeba J

Submitted for the course:

Internet of Things

Topic:

Automatic Pet Feeding System

Submitted by:

Nishith Surana-18BCE0220
Rohit Mehta-18BCE0199
Om Shiv Keshri-18BCI0165

Index

1.Abstract

The work is about pet feeding machine automatically for a daily minimum period of time of eight hours when all people of nuclear family members are busy at work for the survival in the metro cities and other cities. This set up is controlled by the Mobile app automatically. The pets of home can get food after an interval of half hours and the same can be monitored using mobile app and the owner of the pet always ensure about the feeding of the pet especially dogs and cats when they are busy at work. The food for pet has been kept inside a box while leaving home. The food would be served to the pets automatically up to a certain quantity only when the pet comes near to the box when they feel hungry. It is an excellent idea to protect the pets from starving whenever no one is at home and all are busy with their scheduled routine works and job at office or in business. The cost of a care taking of a pet has been reduced down due to the launching of such machine in the market. Such idea is being globally getting popular due to automatic system and mobile app involvement.

2.Introduction

We chose this project because pet keeping is a time consuming responsibility and we want to provide convenience to owners by helping them feed their pets easily and smartly

2.1.Motivation:

Keeping pets takes many commitments. This includes keeping them company, showing your concerns and of course, feeding them on time and in the correct way. However, not everyone is a pet expert, taking care of your pet's diet can be hard and time consuming. One of the top health concerns of pets are overeating and obesity. Especially at younger age, they are usually satisfied with however much is given to them. Many adult pets are fed unscientifically that later may cause short lifespan. Another problem of feeding pets is that owners might not always be home regularly. Being occupied by personal plans knowing that they still have a starving little fellow at home to be taken care of is always a concern that bothers owners. The third concern that we want to deal with is the fact that there hasn't been any product on the market right now that is able to dispense different foods for different kinds of pets. However, pets themselves might not necessarily recognize the potential health problems of eating the wrong food. Therefore, we want to take care of owners' concern of feeding by building a cloud based automatic pet feeder that can dispense the correct amount of food on time, based on the movement , distance from the dispenser at optimal temperature .

2.2.Objective:

Our project is designed to help all the pet owners to feed their pets remotely and smartly.

Benefits to customer:

- Owners will be able to feed the correct amount of food to their pets by analysing it on the cloud platform , ThingSpeak.
- Owners will be able to monitor their pet's health condition via their daily consumption of food recorded by the Cloud. Sensors will be embedded on the food plate, which measures the difference of weights by the end of each day.
- Owner's pet can be fed automatically when its proactive and looking for food around the dispenser .

3.Project Description and Goals:

Nowadays most of us are fascinated to have pets at their home. But these pets have to be taken care properly. Their feeding on time is an important task as they become part of our family. But in our busy schedule we fail to pay attention on our pet thus it doesn't get proper food on time. This paper addresses the above issue by introducing an Automatic Pet Feeding System to ensure feeding pet on frequent interval of time. Automatic Pet Feeding System consists food storage, servo motor, dispenser, feeding bowl, etc. It also features Arduino to automatically control the operations. It is also possible to make more hi-tech by adding cameras and audio box to check activities of pet and talk to it. Automatic pet feeding system features a machine which can feed pets (e.g. dogs) automatically after frequent intervals in absence of his master. By using machine master don't have to stay with his pet every time to feed it and he gets liberty to do his other works outside without caring about his pet. Dish for feeding pet could be filled in number of ways one is we can set the time and date using Arduino UNO which is displayed on LCD screen fitted on body of pet feeder. This is our automatic pet feeder powered by arduino, using a auger, and programmable with two feeding times with user set quantity of food, with a battery backed up internal clock.
Our goal is to optimize the dispense of the food and it will help to keep the track of the pet's health more efficiently based on the amount of food they have consumed by the Pet Feeder.

4.Literature Review (Existing System):

Current automated pet feeders on the market meet various parts of our design objectives but are quite costly. The Petwant SmartFeeder by GemTune permits scheduled feeding and system configuration via a smartphone app. The Automatic Pet Feeder by Wireless Whiskers utilises RFID to release food for the pet when close by and it is at the appropriate feed time. The Pet-net Smartfeeder is an automated pet feeder that focuses on delivering smarter meal portions for the pet, and it comes with a smartphone application as the main user interface. The smartphone application can modify a pet's feeding schedules and monitor a pet's overall dietary habits, but it can monitor only one pet and does not support multiple pets. The Feed and Go Smart pet feeder is a Wi-Fi enabled automated pet feeder that allows a user to remotely feed a pet, and it also comes with a smartphone application

that can modify a pet's feeding schedule and food amount. It is also supported as a web, Android, and iOS application. Moving away from existing general feeders, more macro components must also be considered, specifically the mechanism being used to dispense the food from the container. This mechanism must have fine control of dispensing food and also be mechanically sound enough to avoid jamming and binding along the path of the food. Several dispensing mechanisms were found in practical scenarios. The first mechanism is the paddle based dispenser seen in cereal dispensers. These are simple but often bind and do not have very fine control. The second mechanism is the sliding door dispenser. This is a simple open and close sliding door that would rarely experience any binding but would be very difficult to control portions. The final mechanism is the feed screw, or auger. This design uses a helical screw to push food along a pipe. Small amounts of food fall between each screw blade and are pushed by the rotating motion. This allows for precise feeding as well as a design that mitigates binding on the edges.

5.Technical Specification (Components List):

Name	Quantity	Component
U2	1	Arduino Uno R3 (Integrated Wifi Module)
D2	1	Blue LED
PIEZO1	1	Piezo
R1	1	220 Ω Resistor
DIST1	1	Ultrasonic Distance Sensor
PIR1	1	2.5 , -229.23029457816847 , -152.7216324982433 , -164.11458866489386 PIR Sensor
SERVO1	1	Positional Micro Servo
U1	1	Temperature Sensor [TMP36]
P2	1	0 , 5 Power Supply

6.Working Model of Sensors and Actuators:

6.1.Arduino Uno R3 (Integrated WiFi):

The Arduino Uno WiFi is an Arduino Uno with an integrated WiFi module. The board is based on the ATmega328P with an ESP8266WiFi Module integrated. The ESP8266WiFi Module is a self contained SoC with integrated TCP/IP protocol stack that can give access to your WiFi network (or the device can act as an access point). One useful feature of Uno WiFi is support for OTA (over-the-air) programming, either for transfer of Arduino sketches or WiFi firmware.

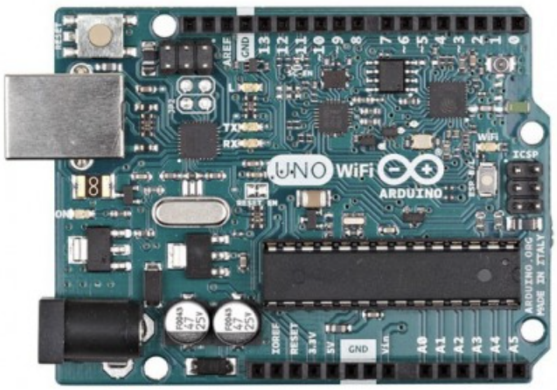


Figure 1: Arduino Uno R3 (Integrated WiFi)

6.2.Temperature Sensor (TM36):

TMP36 is a low voltage, precision centigrade temperature sensor • Temperature range -40°C to 125°C • It is a chip that provides a voltage output that is linearly proportional to the temperature in °C and used for classifying the optimal temp and time for dispense .A temperature sensor creates a changing voltage signal depending on the temperature it senses. It has three pins: one that connects to ground, another that connects to 5 volts, and a third that outputs a variable voltage to Arduino, similar to the analog signal from a potentiometer.

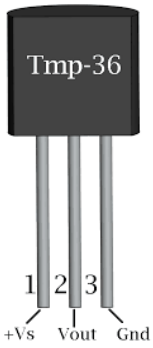


Figure 2: Temperature Sensor (TM36)

6.3.Distance Sensor (HC-SR04):

The HC-SR04 Ultrasonic distance sensor consists of two ultrasonic transducers. The one acts as a transmitter which converts electrical signal into 40 KHz ultrasonic sound pulses. The receiver listens for the transmitted pulses. If it receives them it produces an output pulse whose width can be used to determine the distance the pulse travelled. Since it operates on 5 volts, it can be hooked directly to an Arduino or any other 5V logic microcontrollers.

VCC is the power supply for HC-SR04 Ultrasonic distance sensor which we connect the 5V pin on the Arduino. Trig (Trigger) pin is used to trigger the ultrasonic sound pulses. Echo pin produces a pulse when the reflected signal is received. The length of the pulse is proportional to the time it took for the transmitted signal to be detected. GND should be connected to the ground of Arduino.

Figure 3: Distance Sensor (HC-SR04)

6.4.PIR Sensor:

PIR sensor can detect animal/human movement in a requirement range. PIR is made of a pyroelectric sensor, which is able to detect different levels of infrared radiation. The detector itself does not emit any energy but passively receives it. Pin1 corresponds to the drain terminal of the device, which connected to the positive supply 5V DC. Pin2 corresponds to the source terminal of the device, which connects to the ground terminal via a 100K or 47K resistor. The Pin2 is the output pin of the sensor. The pin 2 of the sensor carries the detected IR signal to an amplifier from the Pin3 of the sensor connected to the ground.

Figure 4: PIR Sensor

6.5.Positional Servo Motor:

Tiny little servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds you're used to 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. Of course, its not nearly as strong as a standard servo. Works great with the Motor Shield for Arduino or by just wiring up with the Servo library. Comes with a few horns and hardware.

Figure 5: Positional Servo Motor

6.6.Photo Resistor:

A Photoresistor (acronymed LDR for Light Decreasing Resistance, or light-dependent resistor, or photo-conductive cell) is a passive component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface.

Figure 6: Photo Resistor

Note: Rest of the component are resistors and LEDs

7.PROPOSED ARCHITECTURE

7.1.Design Approach (Circuit Diagram)

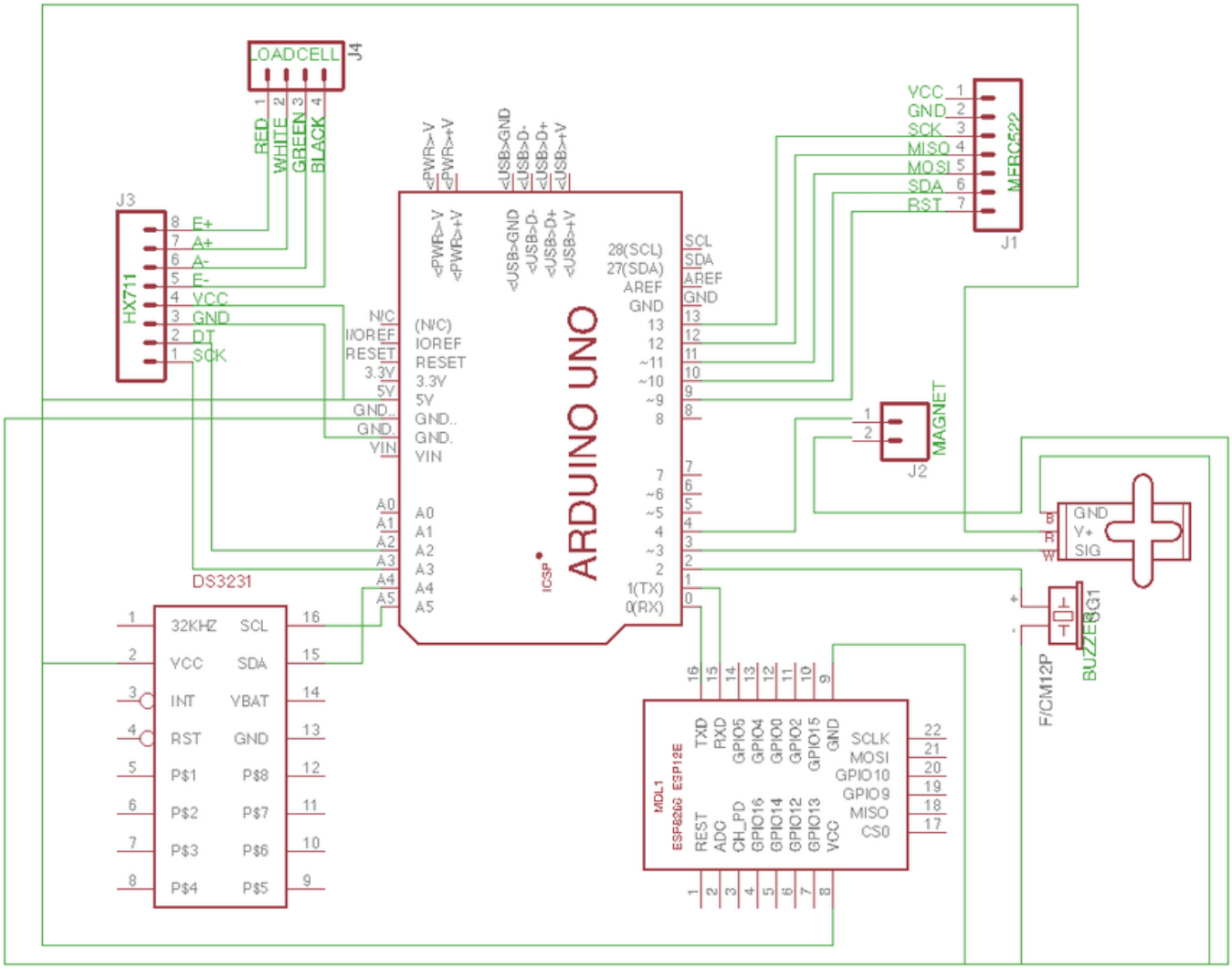


Figure 7: Design Approach (Circuit Diagram)

7.2.Design Approach (Block Diagram)

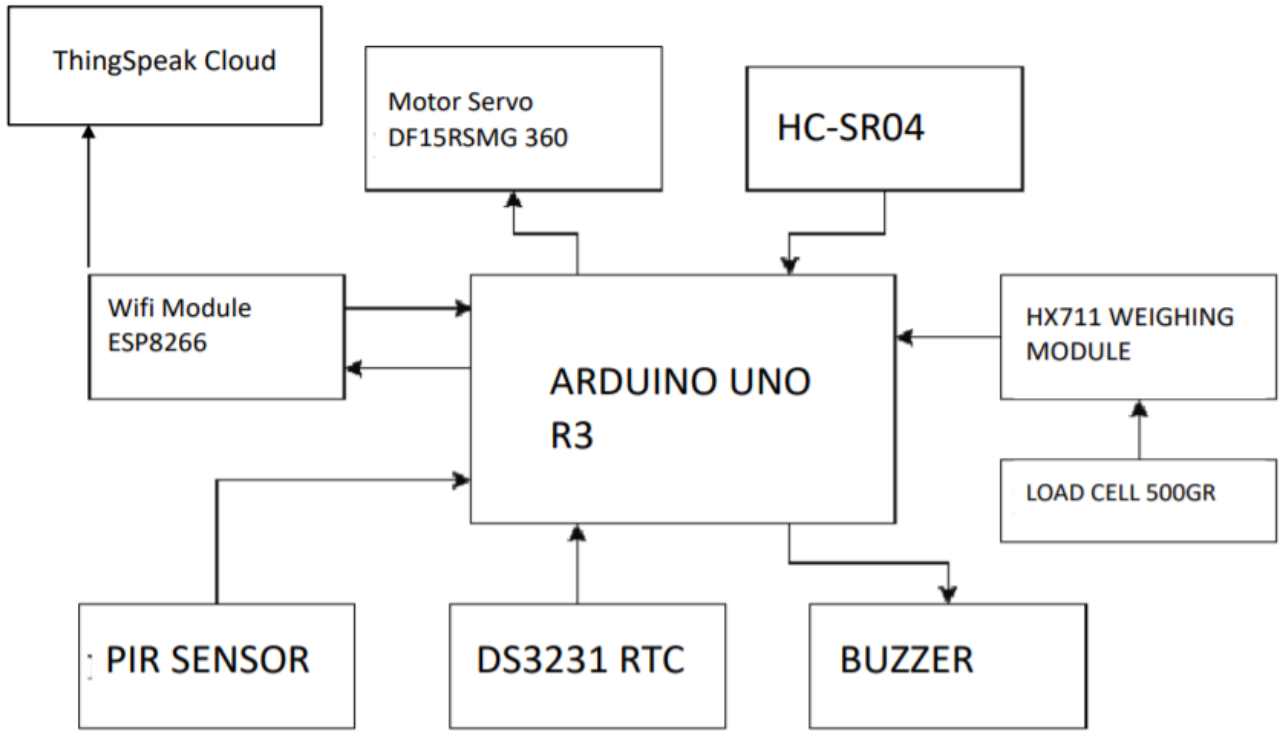


Figure 8: Design Approach (Block Diagram)

8.Software Simulation Setup

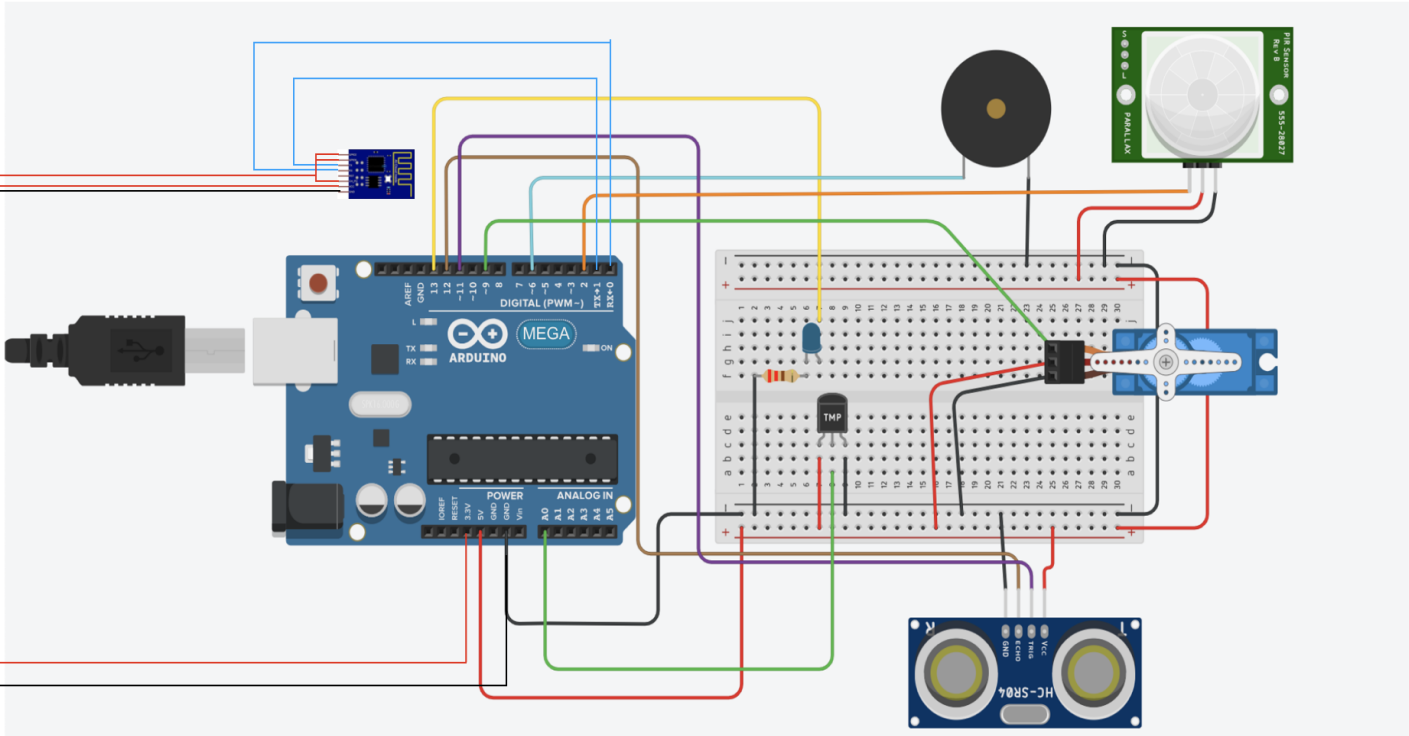


Figure 9: Software Simulation Setup

9.ARDUINO CODING:

```
#include <ESP8266_Lib.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#include <WiFiClient.h>
#include <WiFiServer.h>
#include <Servo.h>
#include "ThingSpeak.h"
#include <SoftwareSerial.h>
#define ESP8266_BAUD 115200

Servo myservo;
int LED = 13;
int pir = 2;
int buzzer = 6;
int tempSensor = A0;
const int triPin = 11;
const int echoPin = 12;

//variables
int val = 0;
int duration;
double dis_cm = 0.0;
double dis_inches = 0.0;
double tempValue = 0.0;
double realTemp = 0.0;

//SoftwareSerial EspSerial(2, 3);

char ssid[] = "air12871";   // your network SSID (name)
char pass[] = "pranay0707";   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)
WiFiClient  client;
unsigned long myChannelNumber = 1578216;
const char * myWriteAPIKey = "M7WZ4KP0L6PT6FK5";

int number1 = 0;
int number2 = random(0,100);
int number3 = random(0,100);
int number4 = random(0,100);
String myStatus = "";
```

```
void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(tempSensor,INPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(pir,INPUT);
  pinMode(triPin, OUTPUT);
  pinMode(echoPin, INPUT);
  myservo.attach(9);

  Serial.begin(115200);
  Wifi.mode(WIFI_STA);
  ThingSpeak.begin(client);

}


void loop()
{

  //detect motion
  val = digitalRead(pir);

  //measure and calibrate temperature value
  tempValue = analogRead(tempSensor);
  realTemp = (((tempValue/1024.0) * 5.0)-.5)*100;

// listen for user input and send it to the ESP8266
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  ThingSpeak.setField(1, number1);
  ThingSpeak.setField(2, number2);
  ThingSpeak.setField(3, number3);
```

```
  //shoot the pin at the object
  digitalWrite(triPin, LOW);
  delayMicroseconds(5);
  digitalWrite(triPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triPin, LOW);
  delayMicroseconds(5);

   //capture echo
  duration = pulseIn(echoPin, HIGH);


  //measure distance
  dis_cm = (duration/2) / 29.1;
  dis_inches = (duration/2) /74.0;

}
  //main program
  if(val == HIGH && dis_cm > 50 && realTemp >=38.3 && realTemp <= 40.2 )
  {
      digitalWrite(LED, HIGH);
      tone(buzzer, 200, 250);
      delay(300);
      tone(buzzer, 300, 300);
      delay(350);
      tone(buzzer, 450, 500);
      myservo.write(180);
      delay(500);
      myservo.write(0);

     }

    else
 {
      digitalWrite(LED, LOW);
      digitalWrite(buzzer,LOW);
      myservo.write(0);

  }
  ThingSpeak.setStatus(myStatus);

  // write to the ThingSpeak channel
```

```
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }
  number1 = (int)realTemp;
  number2 = (int)dis_cm;
  number3++;
  {

  //Display reading in Serial Monitor

  Serial.print("Distance reading: ");
  Serial.print(dis_cm);
  Serial.print(" cm, ");

  Serial.print(dis_inches);
  Serial.println(" inches ");

  Serial.print("Cat Temperature: ");
  Serial.print(realTemp);
  Serial.println(" degree celcius ");

  delay(1000);

   }

}

}
```

10. Result and Discussion

10.1.Software Simulation Results

Serial Monitor

Figure 10: Software Simulation Results

Motor Rotation, Buzzer and LED Out

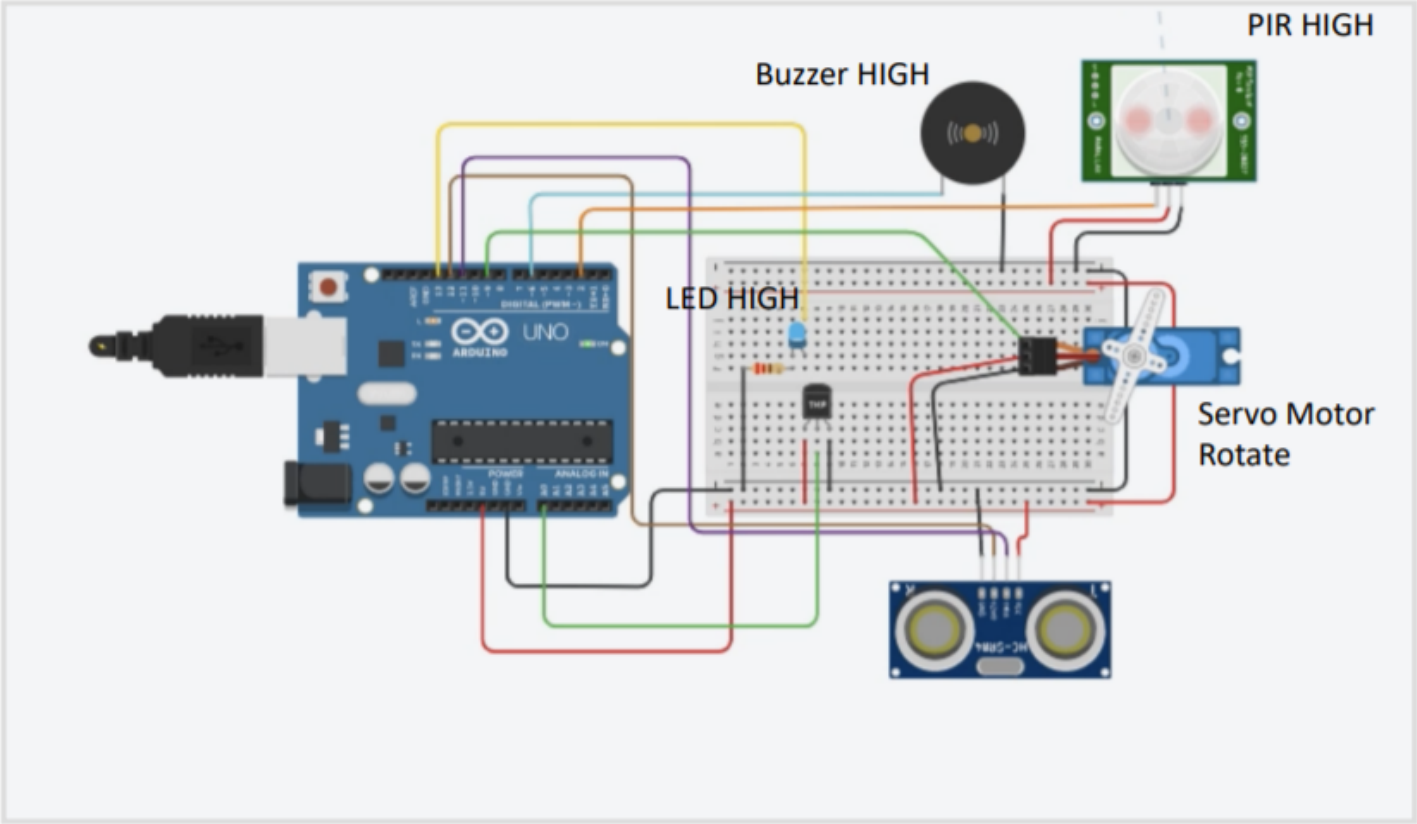
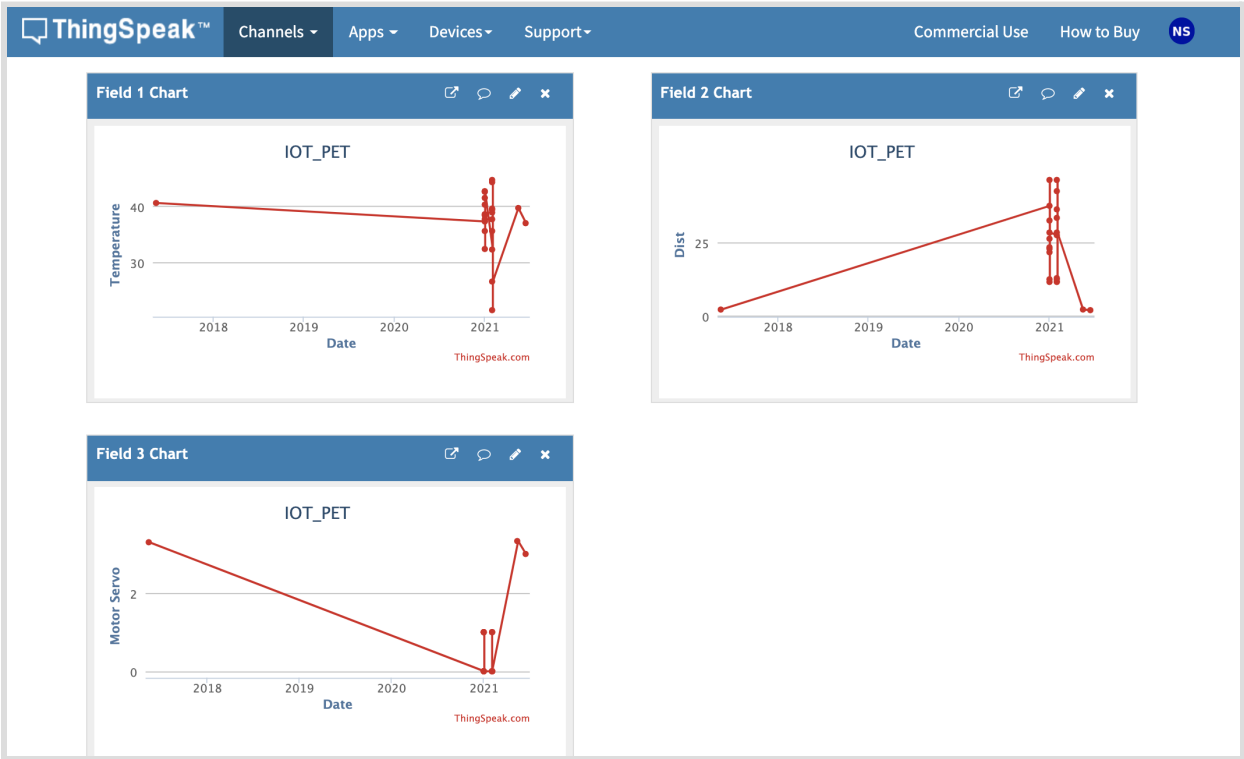


Figure 11: Motor Rotation, Buzzer and LED Out

10.2.Cloud Data Visualization (ThingSpeak)



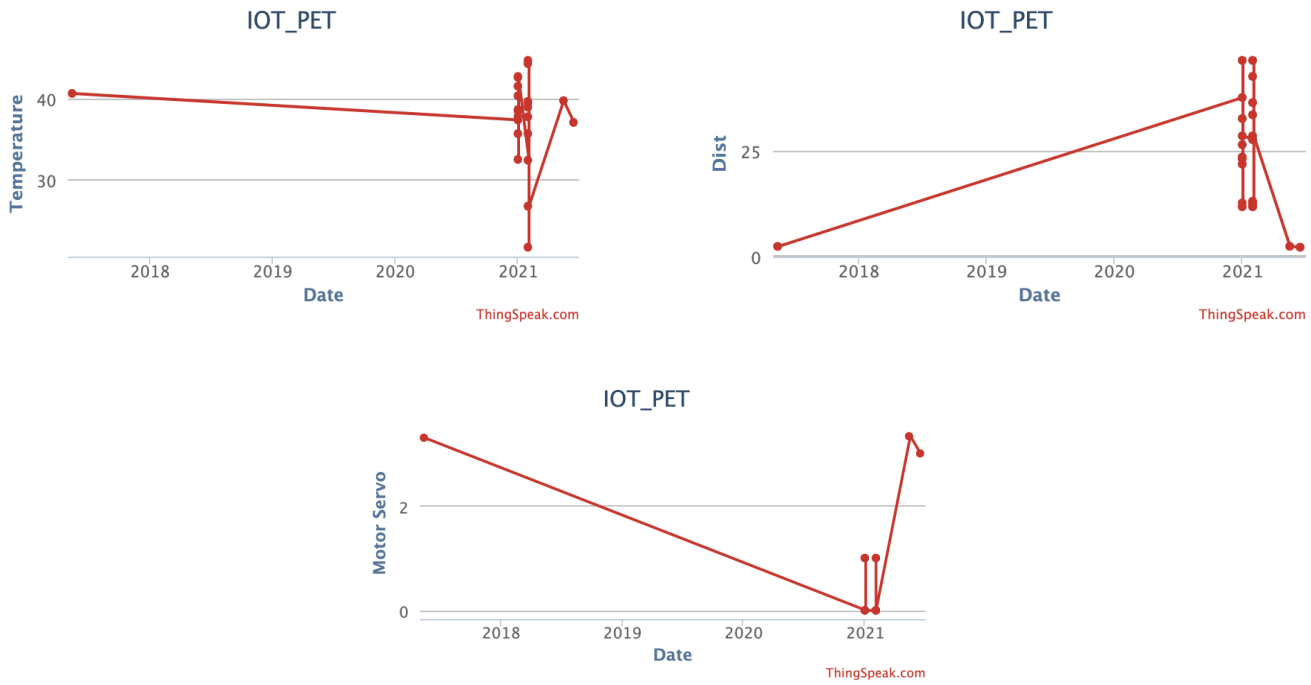


Figure 12: Output of Cloud Data Visualization (ThingSpeak)

From the above data visualisation of the pet movements and temperature for dispensing the food we can analysis and predict the pet's feeding time thereby properly scheduling it . It can also help in thus maintaining dietary plan on the pets.

11.Conclusion

This project brought together several components and idea to achieve a common goal that is design an automated pet feeder using Arduino UNO. the key components of the project include a servo motor sensor which will be programmed to serve the food as soon as the pet comes. It relieves the owner from having to feed his pet multiple times a day. The proposed project senses the presence of the pet using the PIR Sensor , Distance sensor and uses wifi to upload the serial monitor data to the cloud platform , ThingSpeak .

Through this project, it helps to develop creativity in crating projects and modify existing projects to be more energy efficient with new fabrication methods. This smart pet feeder innovation makes it easy for consumers to feed their pets and will not leave their pets hungry again.. This indicates that the project has the potential to be extended to external agencies to expand its use. It is recommended that the promotion to be carried out for commercialisation purpose.

12.Future Work

Keeping in mind the current scenario of industries, our project will not be limited to just a feeding system but it could be extended to various other domains.

Some of them are:
Production and manufacturing Industries: Where there is a specific requirement for the precise control of material.

In agriculture and horticulture where this project could be deployed to control and automate the fertilizer dispensing system.
In cattle farms this project could be deployed to Automate the process of food dispensing for cattle.
Hence the potential of our project is immense and if deployed correctly, this could bring a revolution in various industries.

13.REFERENCES

[1]. Tiwari, M., Hawal, S., Mhatre, N., Bhosale, A. and Bhaumik, M., 2018. Automatic Pet Feeder Using Arduino. [online] Ijirset.com. Available at: <http://www.ijirset.com/upload/2018/march/149_Automatic%20Pet%20Feeder%20Using%20arduino%20IJ70303199.pdf>.
[2]. Subaashri, S., Sowndarya, M., Sowmiyalaxmi, D., Sivassan, S. and Rajasekaran, C., 2017. Automatic Pet Monitoring and Feeding System Using IoT. [online] Sphinxsai.com. Available at: <http://www.sphinxsai.com/2017/ch_vol10_no14/2/(253-258)V10N14CT.pdf>.
[3]. Abdulla, R., Selvaperumal, S., Al-Adilee, M. and Eldebani, A., 2020. IOT based Pet Feeder. The Mattingley Publishing Co., Inc, [online] Available at: <https://www.researchgate.net/publication/340261266_IOT_based_Pet_Feeder>.
[4]. Adriansyah, A., Wibowo, M. and Ihsanto, E., 2016. Design of Pet Feeder using Web Server as Internet of Things Application. 2nd International Conference on Electrical Engineering and Informatics (Icon-EEI 2016), [online] Available at: <https://www.researchgate.net/publication/330702107_Design_of_Pet_Feeder_using_Web_Server_as_Internet_of_Things_Application>.
[5]. Karyono, K. and Nugroho, H., 2016. Smart dog feeder design using wireless communication, MQTT and Android client. IEEE 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), [online] Available at: <https://www.researchgate.net/publication/314105271_Smart_dog_feeder_design_using_wireless_communication_MQTT_and_Android_client>.
[6]. https://create.arduino.cc/projecthub/circuito-io-team/iot-pet-feeder-10a4f3