Table of Content

List of Figures

List of Tables

Abstract

The proposed system is about pet feeding when all people of nuclear family members are busy at work for the survival in the metro cities and other cities. The proposed system will be particularly useful for those people who have to worry about leaving their pets alone at home while going out. The system ensures the pets always remain fed and healthy even when the owner is not at home. The implementation is in 2 modes: AUTOMATIC and MANUAL. Automatic mode consists of PIR sensor, SERVO motors and active RFID. As the pet travels towards the food bowl, PIR sensor detects its motion, RFID detects which pet it is, then the SERVO motor gets activated immediately and it serves the food for that pet and a SMS will be sent to the phone through the GSM module. In manually operated mode, an app called Blynk has been installed in cell phones. Basically, Blynk is a digital dashboard where a graphic interface can be built by simply dragging and dropping widgets. Here there is a use of a button. As the button is clicked, data is sent through the cloud, goes on to the ESP which is connected to the model and then it serves the food. RTC widget is a clock used to retrieve real time from server. The time input is used to take the time from the user. Here, user enters the time, it goes to the server, then through ESP it goes to the model, it compares the real time data obtained from the server to the time entered by the user. It is a 24-hour format. If the time matches then it serves the food. Food level is known through ultrasonic sensor and the level is shown on the app. Here, ultrasonic sensor is used for measuring food.

Chapter 1

Introduction

In early 1982 the concept of the network of smart devices was discussed, with a modified Coke machine. This coke machine is modified at “Carnegie Mellon University” and becoming the first Internet-connected appliance. This machine was able to report its inventory and whether newly loaded drinks were cold. In 1994 Reza Raji explained the idea of IoT as “small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories”. After that many companies proposed various solutions like Microsoft’s at Work or Novell’s Nest. Bill Joy proposed Device to Device communication as a part of his “Six Webs” frameworks at the World Economic Forum at Davos in 1999. The thought of Internet of Things first became popular in 1999. British entrepreneur Kevin Ashton first used the term Internet of Things in 1999 while working at Auto-ID labs. Besides that near field communication, barcode scanners, QR code scanners and digital watermarking are the various devices which are working on IoT in the present scenario.

Today, Internet application development demand is very high. So IoT is a major technology by which can produce various useful internet applications. Basically, IoT is a network in which all physical objects are connected to the internet through network devices or routers and exchange data. IoT allows objects to be controlled remotely across existing network infrastructure. IoT is a very good and intelligent technique which reduces human effort as well as easy access to physical devices. This technique also has autonomous control feature by which any device can control without any human interaction.

IOT LIFECYCLE-

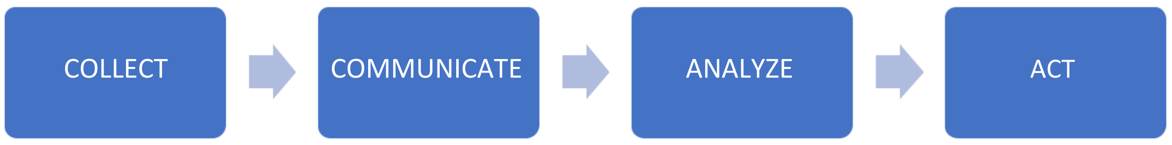


Fig.1.1

COLLECT – Devices and Sensors are collecting data everywhere.

- At home
- In car
- At office
- In manufacturing plan

COMMUNICATE – Sending data and events through networks to some destination.

- A cloud platform
- Private data centre
- Home network

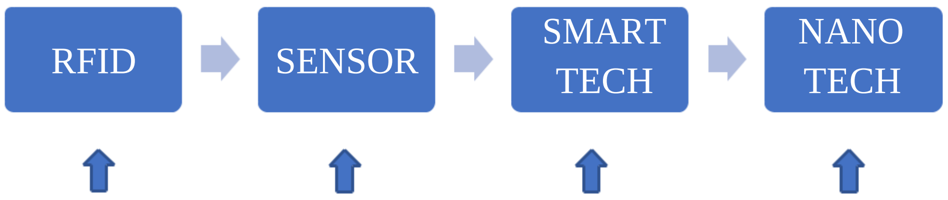
ANALYSIS – Creating Information from the data.

- Visualizing the data
- Building reports
- Filtering data

ACTION – Taking action based on the information and data.

- Communicate with another machine
- Send a notification (SMS, email, text)
- Talk to another system

WORKING OF IOT-



To identify and track the data of things.	To collect and process the data to detect the changes in the physical status of things	To enhance the power of network by developing processing capabilities to different parts of the network.	To make the smaller and smaller things have the ability to connect and interact.

Fig.1.2

APPLICATIONS OF IOT-

1. SMART HOME - Whenever the idea of IoT system pops up, the most important and efficient application that stands out every time is Smart Home ranking. Database of smart homes for IoT Analytics includes 256 companies and start-up's. More companies are now actively being involved in smart homes than similar other applications in the field of IoT. The estimated amount of funding for Smart Home start-up's exceeds $2.5bn and is ever growing. The list of start-up's includes prominent start-up company names such as Alert Me or Nest as well as a number of multinational corporations like Philips, Haier, or Belkin etc.

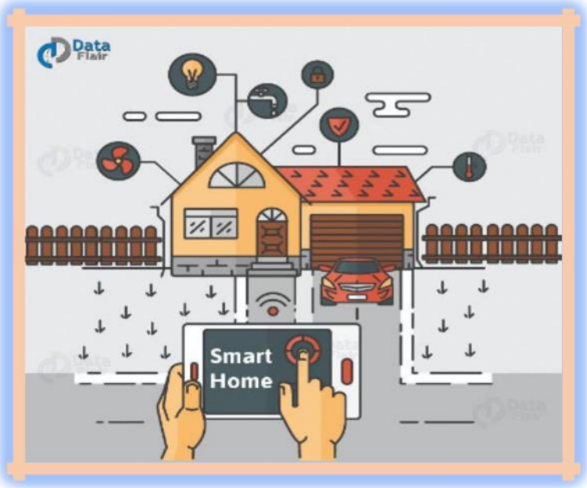


Fig.1.3

2. WEARABLE - Just like smart homes, wearables remain a hot topic too among potential IOT applications. Every year, consumers all across the globe await the release of Apple smartwatch. Apart from this, there are plenty of other wearable devices that make our life easy such as the Sony Smart B Trainer, or Look-see bracelet, the Myo gesture control.

Fig.1.4

3. SMART CITY - The smart city like the name suggests is a very big innovation and spans a wide variety of use cases, from water distribution to traffic management to waste management, environmental monitoring, and urban security. The reason why it is so popular is that it tries to remove the discomfort and problems of

Fig.1.5

people who live in cities. IoT solutions offered in the Smart City area solve various city-related problems comprising of traffic, reduce air and noise pollution and help make cities safer.

4. SMART GRIDS - Smart grids is another area of application that stands out. A smart grid basically promisestoextract information on the behaviours of consumers and electricity suppliers in an automated fashion in order to improve the efficiency, economics, and reliability of electricity distribution.

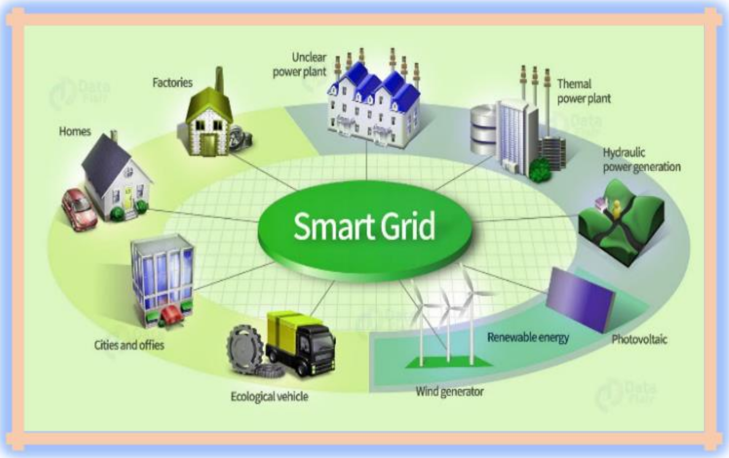


Fig.1.6

5. CONNECTED CAR - Connected car technology is a vast and an extensive network of multiple sensors, antennas, embedded software, and technologies that assist in communication to navigate in our complex world. It has the responsibility of making decisions with consistency, accuracy, and speed. It also has to be reliable. These requirements will become even more critical when humans give up entirely the control of the steering wheel and brakes to the autonomous or automated vehicles that are being successfully tested on our highways right now.

Fig.1.7

Chapter 2

Problem Definition

Construction of an automatic pet feeder which would be operated without the need of a human being or it can be used manually. A container will store food and it will be physically rotated by a servo motor. Arduino Mega can be used and it will be connected to a mobile based App known as “Blynk”. This application will provide both functionality and information such as detecting and notifying automatic feeding of pet, setting timer for frequent feeding, providing a button to the user to manually drop food from the container, checking food quantity on the container and notifying when food is low.

Chapter 3

Literature Survey

In this paper, smart dog feeder has been designed using wireless communication, MQTT and smart client is presented. This device can provide regular feeding without disrupting owner's work. Owners can monitor feeding process with their Android smartphone virtually. This system gives authentication with RFID, set feeding time and portion per serving through Android smartphone, send feeding report and dog arrival when the feeding time has arrived. Every setting about feeding time, portion, stock and waiting time will be set on Android with minimum requirement of Jelly Bean version, SDK 18 and has been installed with Appliance Hub application. This system has stock information, feed schedule, waiting time and owner's name from server uses MQTT protocol. All information will be sent in JSON format and will be processed by this system, Android and server. Smart Dog Feeder will save the schedule and set RTC alarm which will interrupt when the feed time has arrived. Authentication process is done by checking RFID tag which is attached on dog collar. Food will be served based on user's setting and be measured by load cell. Experiment is done by seeing punctuality, portion congruence, delivery of settings and notification within devices. The result of experiment is Smart Dog Feeder can receive messages from server and do feeding at the right time. [1]

Over the years, more and more households are adopting pets. Pet feeder came into existence as more and more pet owners found it difficult to cater time to feed their pets. Pet feeders are automated machines that dispense food at present schedule. The proposed project is about programmable pet feeder implementation. It is timed based and it dispense a certain amount of food at specific time of the day. It is a programmable system which is mainly controlled by a microcontroller. It consists of a LCD screen for input display, buzzer to alert pets for meals, stepper motor to control the speed and a turn table which is divided into different sections for placement of different food. User will be able to select the food to be dispensed out at their desired timing. [2]

This system has three modules each of which has the IR unique feature. They are pet monitoring door, pet food feeder and pet collar system. In pet door, IR sensor consist an IR LED and photodiode, in which IR LED emits IR radiation and photodiode detects the radiation. Photodiode conducts current in reverse direction, whenever light falls on it, and voltage across it changes, this voltage change is sensed by voltage comparator and generates output accordingly. In this IR based security alarm circuit, IR LED is placed in front of photodiode, so that IR light can directly fall on photodiode. Whenever someone moves through this beam, IR rays stops falling on photodiode and Buzzer start beeping. In pet food feeder, the feeders must be disassembled for cleaning and then reassembled before further use. The pet feeder can keep the pet food and water clean until the pet is ready to eat. It also has a bowl cover that opens and closes automatically. The bowl cover is actuated by an infrared proximity sensor and battery-operated electric motor. The sensor detects the presence of the pet and then opens the cover, enabling only the pet to have access to the food. When the pet is out of sensor range, the bowl cover closes automatically. This keeps dust, flies, and bugs from reaching the food and keeps the food fresh. [3]

In this this system remote controlled and GSM based automated pet feeder, is been implemented. In this design user can adjust the feed time, time gap between consecutive feeds and the quantity of feed served. This design also contains the call for pet at feed time, refill alert, dual power supply with battery charger, Massage alert system for owner in case of pet don’t get it’s feed, safety lock for container, sensor based system to serve previously served feed in case of left feed and the priority feeder with dual option of serve as by owner can opt for multi time and pet can opt for 1 time between feed time gap. [4]

In the this system phone controlled automatic pet feeder is implemented. The phone controlled automatic pet feeder is meant to provide users to a way to feed their pets precisely and automatically. It consists of two parts: the hardware and a compatible software running on Android. The software allows users to type in their pet’s information including name, weight and feeding amount. The information will then be transmitted to the hardware where the pets can eat their food. A maximum of two pets at one time is supported. The device can distinguish different pets within a range of 15cm and dispense a specific amount of food based on the user’s input. [5]

Chapter 4

Explanation of Current Project

- **Components before modification**

➢ Arduino Mega 2560

- Microcontroller – ATmega2560 (8bit)
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limit) : 6-20V
- Digital I/O Pins: 54 (of which 15 provide PWM output)
- Analog Input Pins: 16
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA+
- Flash Memory: 256 KB of which 8 KB used by bootloader

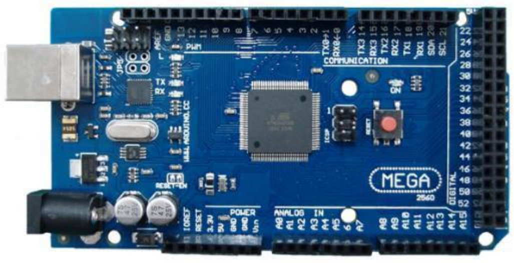


Fig.4.1: Arduino Mega 2560

➢ ESP 8266 WIFI Module

- 802.11 b/g/n protocol.
- Wi-Fi Direct (P2P), soft-AP.
- Integrated TCP/IP protocol stack.
- Input Voltage: 3.3V
- Integrated TR switch, balun, LNA, power amplifier and matching network.
- Integrated PLL, regulators, and power management units.
- +19.5dBm output power in 802.11b mode.
- Integrated temperature sensor.

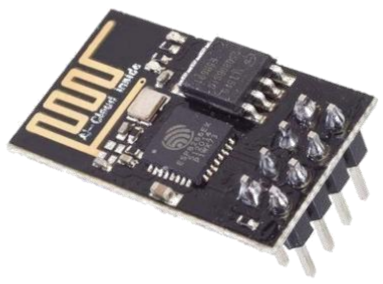


Fig.4.2: ESP8266

- 3.3 V. LD33 Voltage Regulator
    - Input Voltage : 5V-25V
    - Output Voltage : 3.3V

Fig.4.3: LD33 Voltage Regulator

- PIR Sensor
    - Wide range on input voltage varying from 4.V to 12V (+5V recommended)
    - Output voltage is High/Low (3.3V TTL)
    - Can distinguish between object movement and human movement
    - Has to operating modes - Repeatable(H) and Non- Repeatable(H)
    - Cover distance of about 120° and 7 meters
    - Low power consumption of 65mA
    - Operating temperature from -20° to +80° Celsius

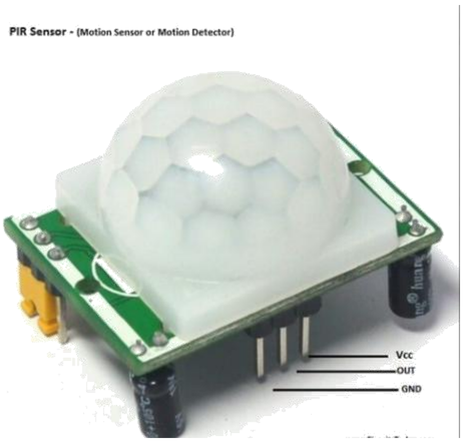


Fig.4.4: PIR Sensor

- 180° MG995 Servo Motor
  - Torque: 4.8V: 130.54 oz-in (9.40 kg-cm)
    6.0V: 152.76 oz-in (11.00 kg-cm)
  - Speed: 4.8V: 0.20 sec/60°
    6.0V: 0.16 sec/60°
  - Weight: 1.94 oz (55.0 g)
  - Dimensions: Length:1.60 in (40.7 mm)
    Width:0.78 in (19.7 mm)
    Height:1.69 in (42.9 mm)
  - Gear type: Metal
  - Pulse cycle: 1ms
  - Rotation range: 0-180 degrees
  - Connector type: JR

Fig.4.5: 180° MG995 Servo Motor

- Ultrasonic Sensor
  - Operating voltage: +5V
  - Theoretical Measuring Distance: 2cm to 450cm

- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

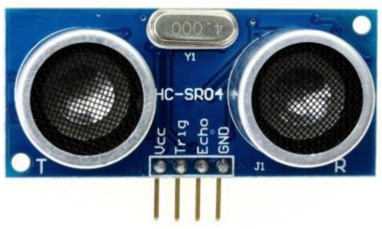


Fig.4.6: Ultrasonic Sensor

➢ 5 V. Adapter

Fig.4.7: Power Adapter

- Gives I/P power to the system.

➢ Breadboard

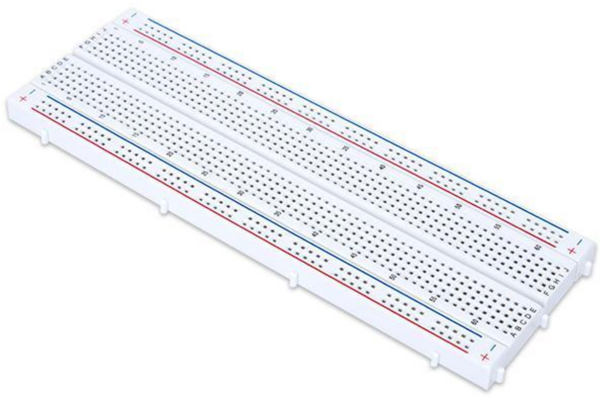


Fig.4.8: Breadboard

- Used for merging different circuits/modules.

- Jumper Cables

Fig.4.9 Jumper cables

  - Used to connect modules with Arduino and breadboard.
  - Types: Male-Male, Male-Female, Female-Female.

Software Components-

- Blynk Mobile App
  - Cloud based mobile application used to connect Arduino to Blynk server and operate the system from anywhere.
- Mobile Phone
  - It is an essential device to make the Blynk application work.
- Arduino IDE
  - It is the Horcrux of the system. It manages every components and executes the desired code for desired output.

Cost Estimation-

SR NO	**COMPONENT**	**COST**
1	Arduino MEGA	₹858
2	ESP WIFI Module	₹150
3	Voltage Regulator	₹25
4	PIR Sensor	₹90
5	Servo Motor	₹400
6	Ultrasonic Sensor	₹150
7	2 x 5V Adapter	₹200
Total		₹1873

Table 4.1

• Project Implementation

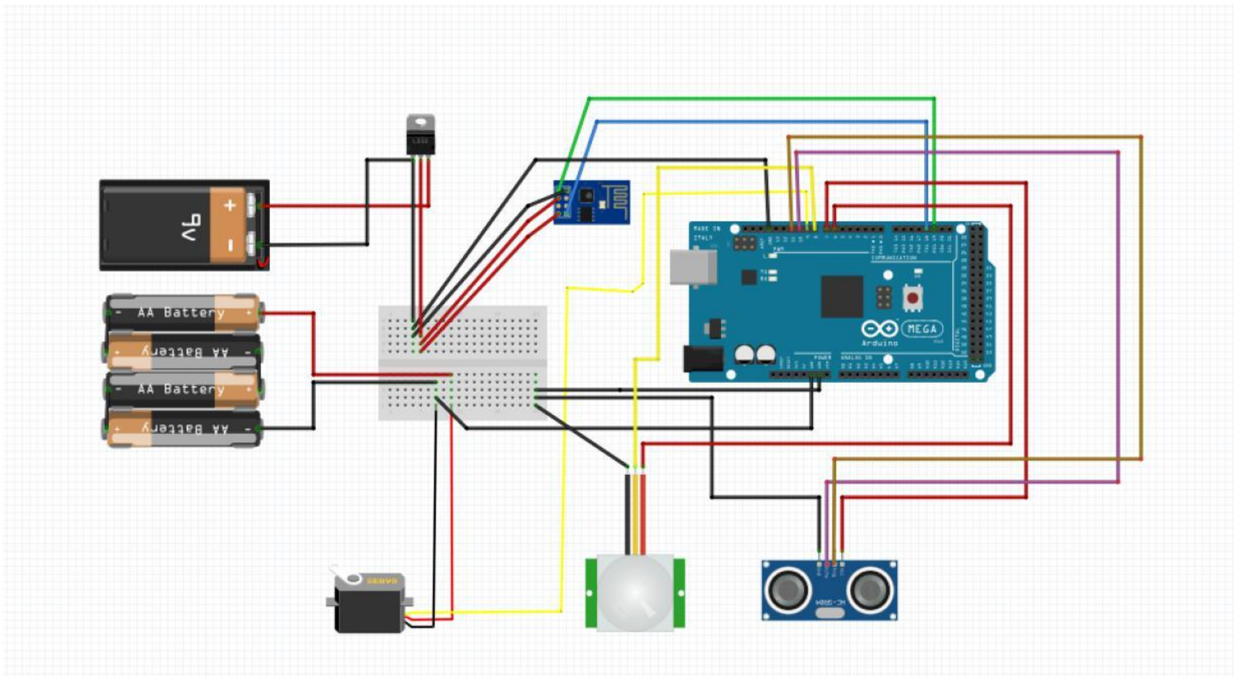


Fig 4.10 Circuit Diagram of existing project

Connections-

The VCC and GND of the 9V battery is connected to the VCC in and GND in of the 3.3V voltage regulator. Then VCC out of the voltage regulator is connected to breadboard. The GND of the voltage regulator is connected to the GND of the Arduino and the GND of the WIFI module. The VCC out which carries the 3.3v from the voltage regulator is connected to the VCC and CH_PD pins of the WIFI module. The TX and RX pins of the WIFI module is connected to the Arduino pins RX1 and TX1 respectively. The VCC and GND of the battery connected in series are connected to the breadboard and then connected to VCC and GND of the servo. The GND of the battery in also connected to the GND pin on the Arduino. The pulse pin of the servo is connected to pin 9 on the Arduino. The VCC of the PIR sensor in connected to pin 6 on the Arduino and the GND pin is connect to the GND of the Arduino through the breadboard and the data pin is connected to pin 8 on the Arduino. The VCC of the ultrasonic sensor is connected to pin 7 on Arduino and the GND pin is connected to the GND of the Arduino through the breadboard. The echo and trig pin in connected to pin 10 and pin 11 of Arduino respectively.

• Testing and Debugging

The first phase of testing started with:

Testing of all the sensors one by one. While testing the ultrasonic sensor we found out that reading did not change even after doing the connections properly, we tried changing the code multiple times then also it was not working. Hence, we concluded that it was faulty. Then after we exchanged the sensor with a new one it started to give an accurate reading.

While testing the PIR sensor, at the start it will sense even though there was no motion. After further research, we found out that the sensor should we allowed to heat up a little after start up and then its readings become accurate.

While testing the Servo Motor we found out that when the servo is run by the power supplied by the Arduino it is not sufficient, the servo does not work. And due to that, there will be insufficient power for the other sensors. So to solve this problem we added an external power supply for the servo. Even after the external power supply, it was not responding. Then after some research, we found out that when one adds an external power supply for a sensor, the negative of that power supply should be connected to the negative of the Arduino. After all this, the servo was working properly.

While testing the ESP 8266 we followed the proper procedure for the connection and set the BAUD rate to 115200 on the serial monitor or else it will not respond properly because the default BAUD rate of ESP is 115200. In the beginning, it was accepting AT commands properly but after some testing, it stopped responding. After some research, we came to know that 3.3V provided by the Arduino is not stable and is not recommended for ESP 8266 to connect to it. We checked the voltage reading of the 3.3V from the Arduino and it was more than 4V. This was the reason ESP was heating up a little bit and stop responding after some time. Then we had to replace the ESP with a new one. This we bought an LD33 3.3V voltage regulator and also added an external power supply for ESP which will pass through the voltage regulator and give clean 3.3V without any fluctuation and also connected the GND of the external power supply to the GND of the Arduino. Now we tried the connections with the new ESP and it was not responding. After some research, it told us to check the voltage and if it is ok then try updating the firmware. Since the voltage was perfect we tried updating the firmware, but in the updating process, it was still not detecting. After further checking, we understood that it was faulty. Then we had to replace the ESP again. This time it was working properly and the voltage was even stable at 3.3V.

The second phase of testing started with:

Checking if the ESP connects to a WIFI and connects to the server provided by Blynk. In this we were trying to connect the ESP to Blynk and pass data to Arduino through Blynk. At the beginning we tried to run the basic code of Blynk just to check if it would connect and every time it showed an error message "ESP not responding". We tried checking the ESP separately and tried running the AT command again it was responding properly. Then we tried to run the basic code again but were still showing the same error message. We did some research but majority of the Blynk application was done on Arduino Uno, so we tried some other basic code provided by the Blynk example generator. The results were still the same and it was still showing the same error message. Then after multiple attempts since it was not responding we shifted from Blynk to RemoteXY. RemoteXY was simple and straightforward, we just had to follow the instructions provided and it was easy to connect the smartphone to the ESP. Everything was working properly but it was connected locally through the ESP and cannot be remotely accessed so we tried to send data through the cloud provided by RemoteXY. First we had to generate a cloud ID of the RemoteXY Cloud to connect to it. Then we changed the code to connect to the cloud and then we found out that there is a huge time delay between the data send and received. Hence we could not use RemoteXY for our project and we shifted back to Blynk. We tried to search about Blynk and Arduino connections on Blynk community Q&A, no luck. While searching we found a site where they used Arduino and Blynk in there project. After that we understood that we have to connect the RX and TX pins from the ESP to the TX and RX pins of the Arduino and also mentions which RX and TX pair we are using on the Arduino in the code, e.g. : "#define EspSerial Serial1" which will connect to the TX1 and RX1 on the Arduino. After doing the appropriate connections, Blynk was working properly and there was no error message. Now we were able to control the system remotely.

The third phase of testing started with:

Connecting all the sensors and components together to the Arduino. While connecting all the sensors we noticed that there was insufficient number of VCC and GND pins on the Arduino. So for the GND pin we connected it to the breadboard, so that PIR and Ultrasonic sensor could be connected to ground, and for VCC for both the sensors we made two digital pins as output and HIGH in step up function in the code which made those two pins into VCC pins which give 5V. Then we connected the VCC of both the sensors to these two digital pins and it is working properly. As for the ESP and servo their VCC and GND were connected to the external power supply the GND of both the power supply is connected to two separate GND on the Arduino. Hence all the sensors were getting sufficient amount of power.

• Explanation of Existing Project

This system is an automatic pet feeder and it is implemented in two modes AUTOMATIC and MAUNAL modes. In the AUTOMATIC mode when the pet comes near the system the PIR sensor senses its motion then activates the Servo motor which drops the food in the bowl below. When the food is dropped through the servo motor a notification is sent to the owner of the pet through the Blynk app saying that your pet has eaten its food through the ESP8266 module. The ultrasonic sensor is used to check the levels of food in the container and is displayed in the Blynk app, if the food level is low then it will send a notification to the owner and a red led will light up in the Blynk app. In the MAUNAL mode the user can control the system, during this mode the PIR module will not be activated. In this there is a real time clock which show the time. The user can drop food for the pet manually by clicking a button, he can also set a specific time at which the food should be dropped and at that given time the food will dropped without the user clicking and buttons. When the timer is set an orange led will light up showing that the timer is activated and after the set time the led will turn off.

- **Result**

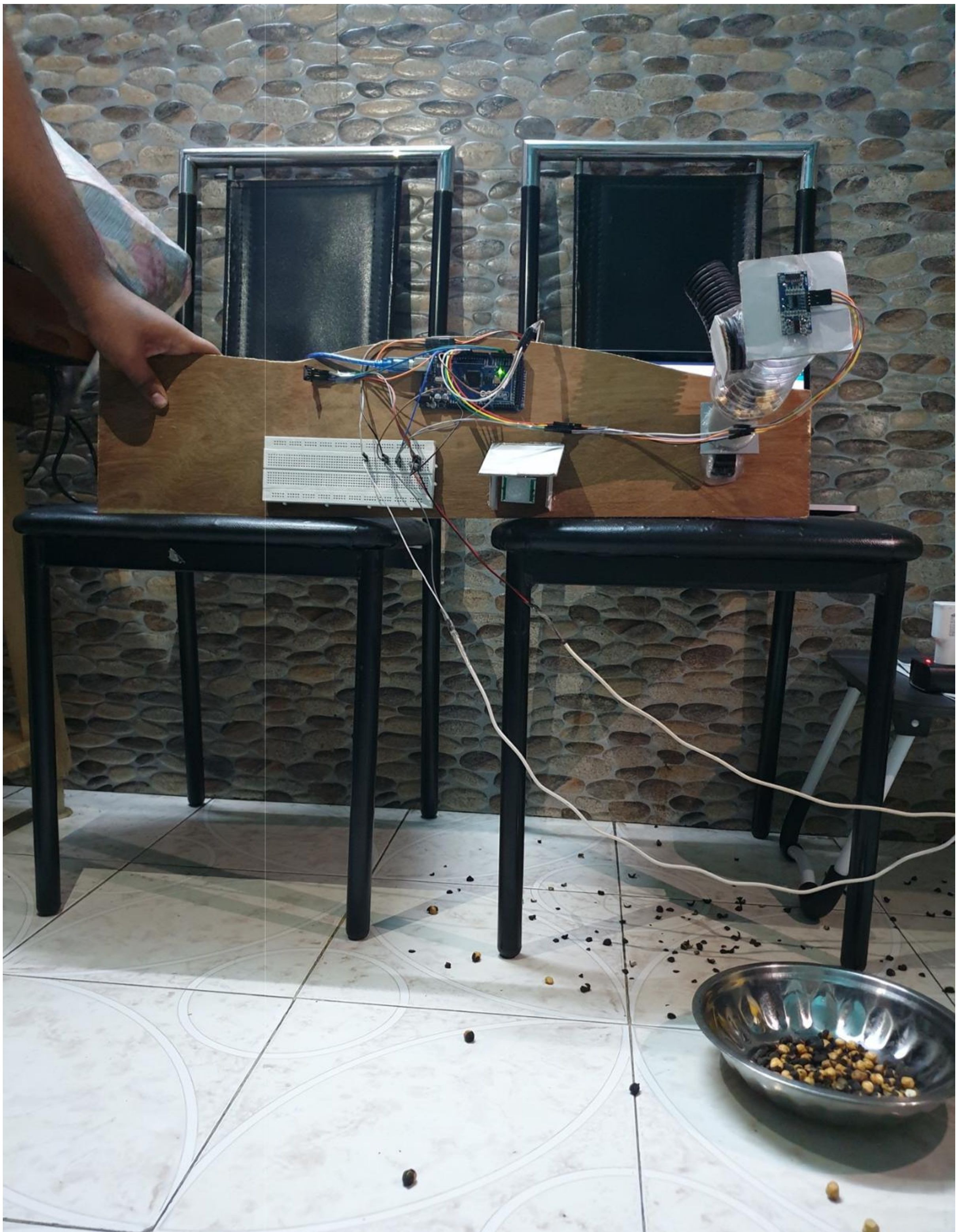


Fig 4.11 System in idle position

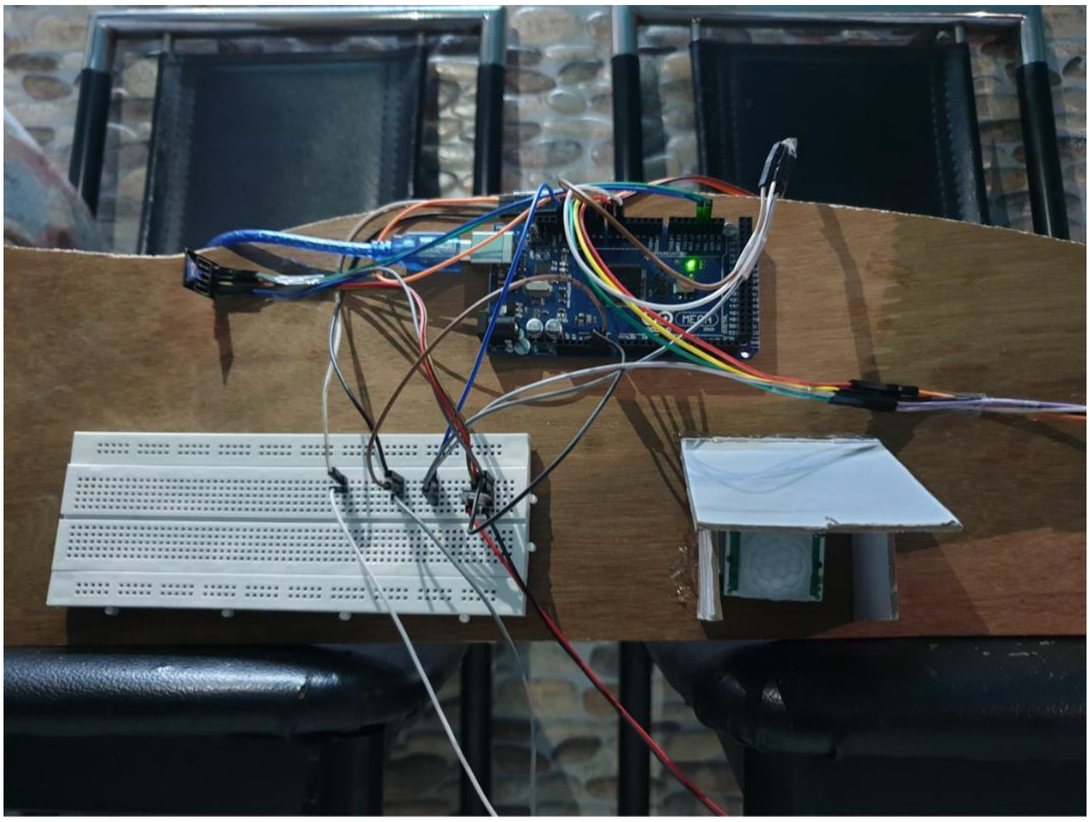


Fig 4.12 Connections of the system

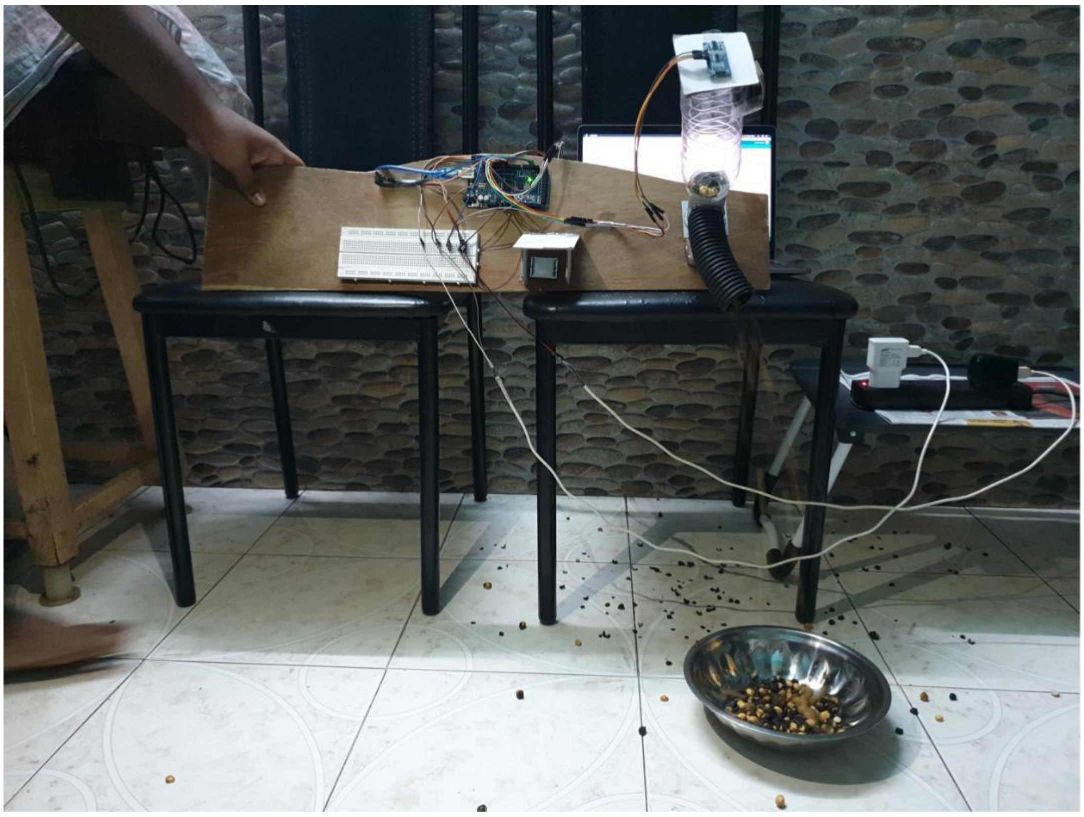


Fig 4.13 System in working position

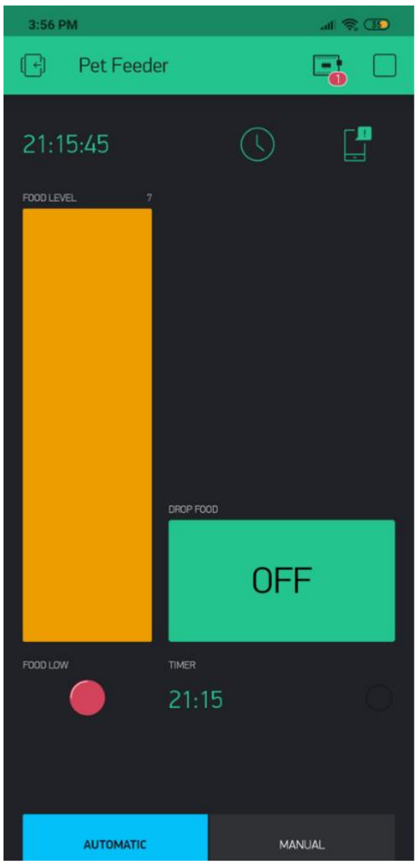


Fig 4.14: App in Automatic mode

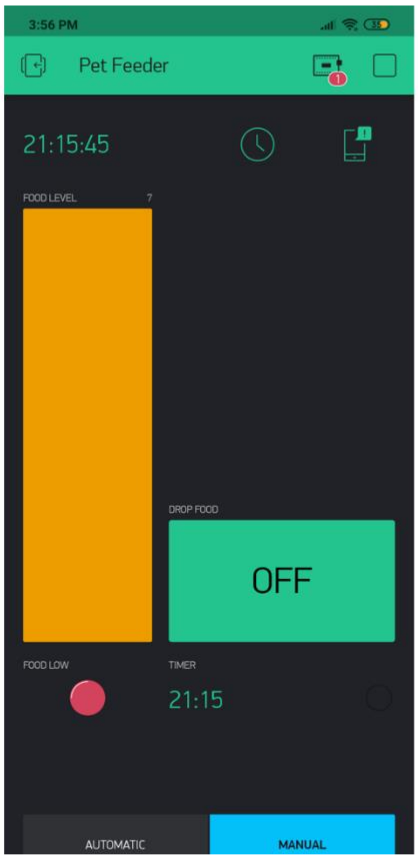


Fig 4.15: App in Manual mode

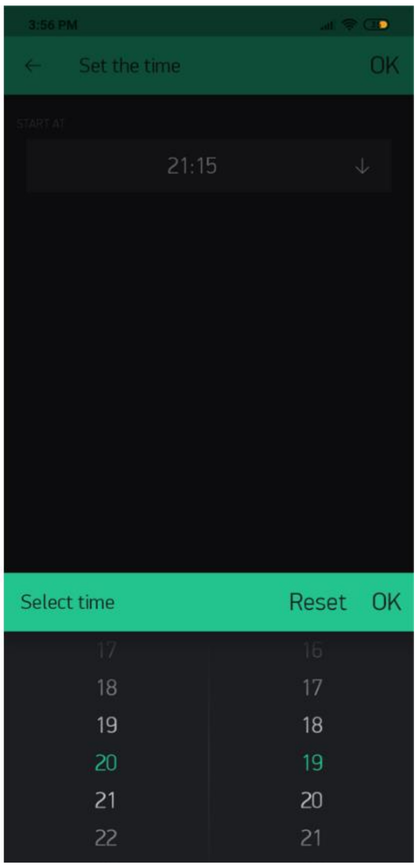


Fig 4.16: To set time in app

Chapter 5

Modifications

- **Block Diagram of Proposed Modification**

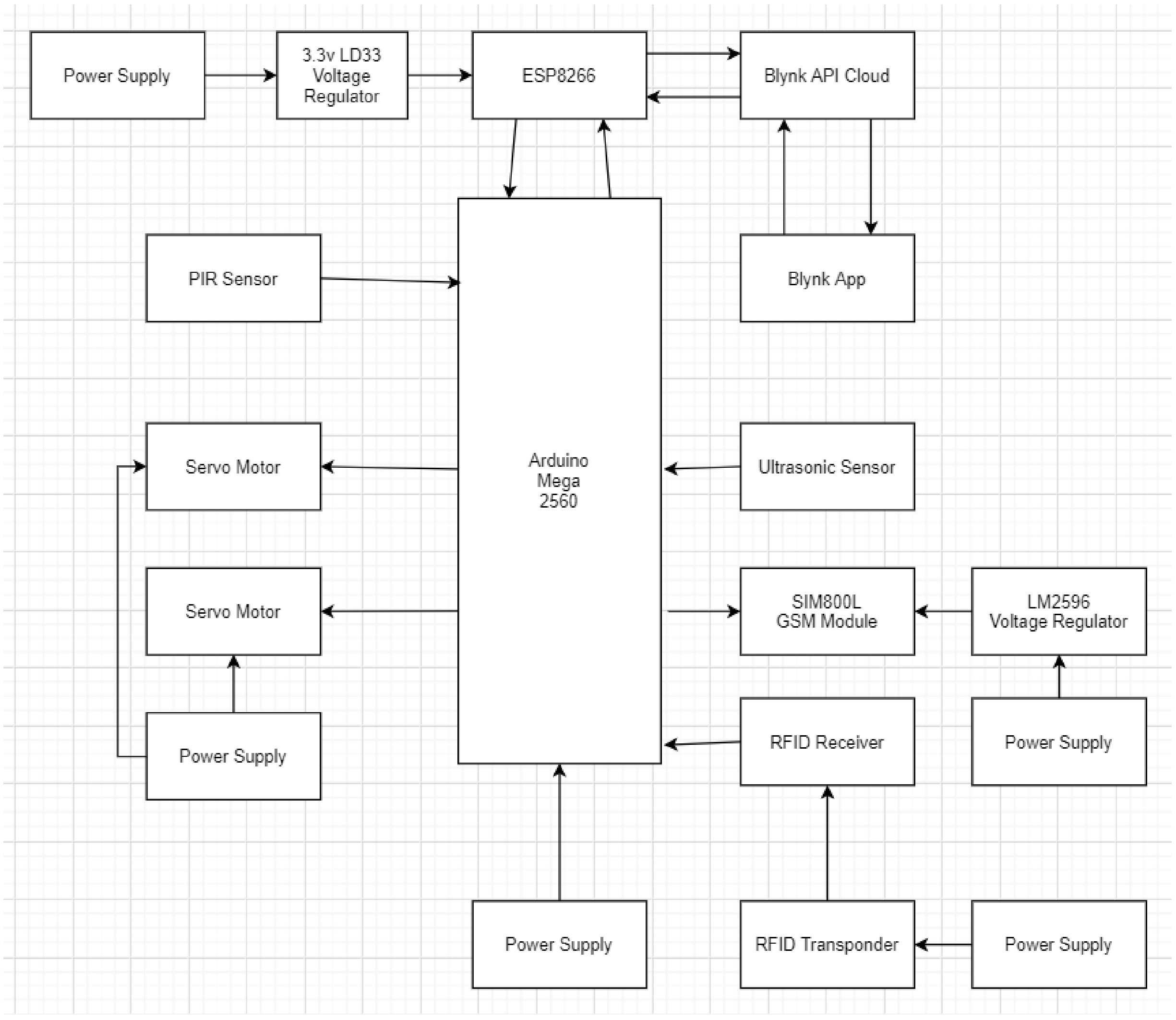


Fig 5.1 Block Diagram

- **Flowchart with Modification**

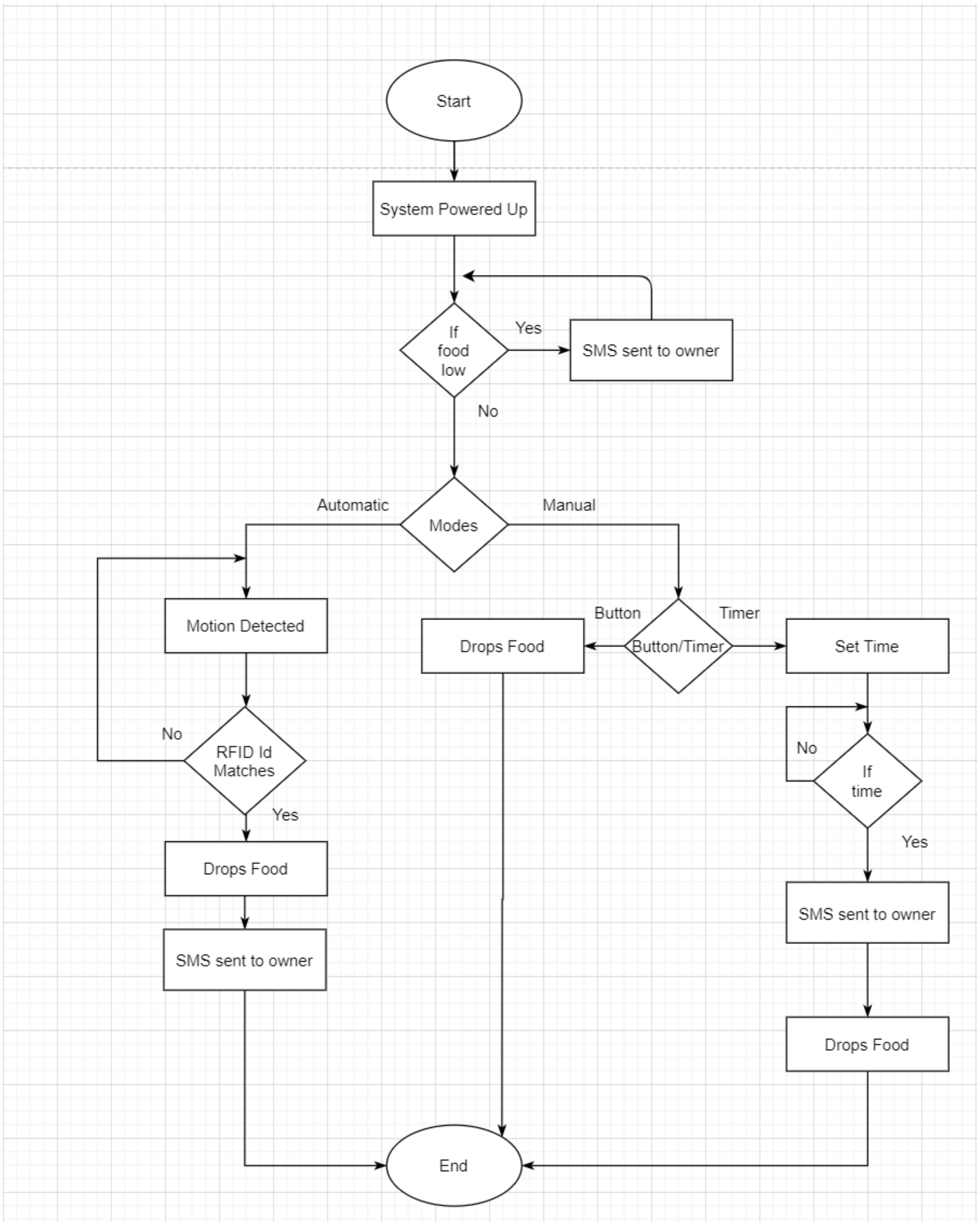


Fig 5.2 Flowchart

- **List of Additional Hardware Components Required**

➢ 180° MG995 Servo Motor

- Torque: 4.8V: 130.54 oz-in (9.40 kg-cm)
  6.0V: 152.76 oz-in (11.00 kg-cm)
- Speed: 4.8V: 0.20 sec/60°
  6.0V: 0.16 sec/60°
- Weight: 1.94 oz (55.0 g)
- Dimensions: Length:1.60 in (40.7 mm)
  Width:0.78 in (19.7 mm)
  Height:1.69 in (42.9 mm)
- Gear type: Metal
- Pulse cycle: 1ms
- Rotation range: 0-180 degrees
- Connector type: JR

Fig.5.3: 180° MG995 Servo Motor

➢ 5 V. Adapter

Fig.5.4: Power Adapter

- Gives I/P power to the system.

- SIM800L GSM Module
    - Operating voltage: 3.4v – 4.4v
    - Baud Rate: 1200bps – 115200bps
    - Supports Quad-band: GSM850, EGSM900, DCS1800 and PCS1900
    - Connect onto any global GSM network with any 2G SIM
    - Make and receive voice calls using an external 8Ω speaker & electret microphone
    - Transmit Power:
        - Class 4 (2W) for GSM850
        - Class 1 (1W) for DCS1800

Fig 5.5: SIM800L GSM Module

- LM2596
    - The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation.
    - Input Voltage: 4.5v – 40v
    - Output Voltage: 3.3v – 37v

Fig 5.6: LM2596

➢ RFID

- Active RFID systems use battery-powered RFID tags that continuously broadcast their own signal. Active RFID tags are commonly used as “beacons” to accurately track the real-time location of assets or in high-speed environments such as tolling

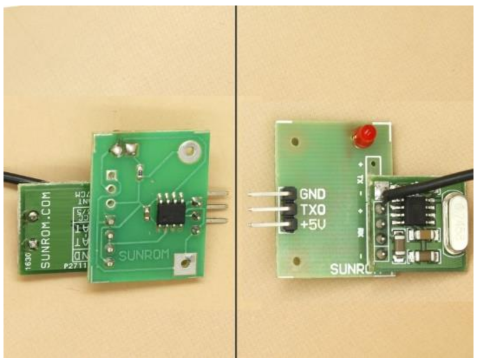


Fig 5.7: RFID Receiver

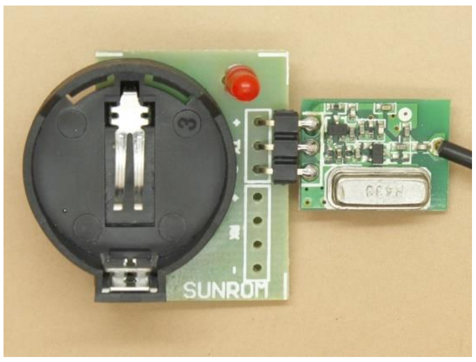


Fig 5.8: RFID Transponder

Cost Estimation-

SR NO	COMPONENT	COST
1	Servo Motor	₹400
2	5V Adapter	₹100
3	RFID	₹1200
4	SIM800L GSM Module	₹900
5	LM2596	₹90
Total		₹2690
Cost of Existing Project		₹1873
Total cost of Project after Modification		₹4563

Table 5.1

- **Circuit Diagram of Proposed Modification in Project**

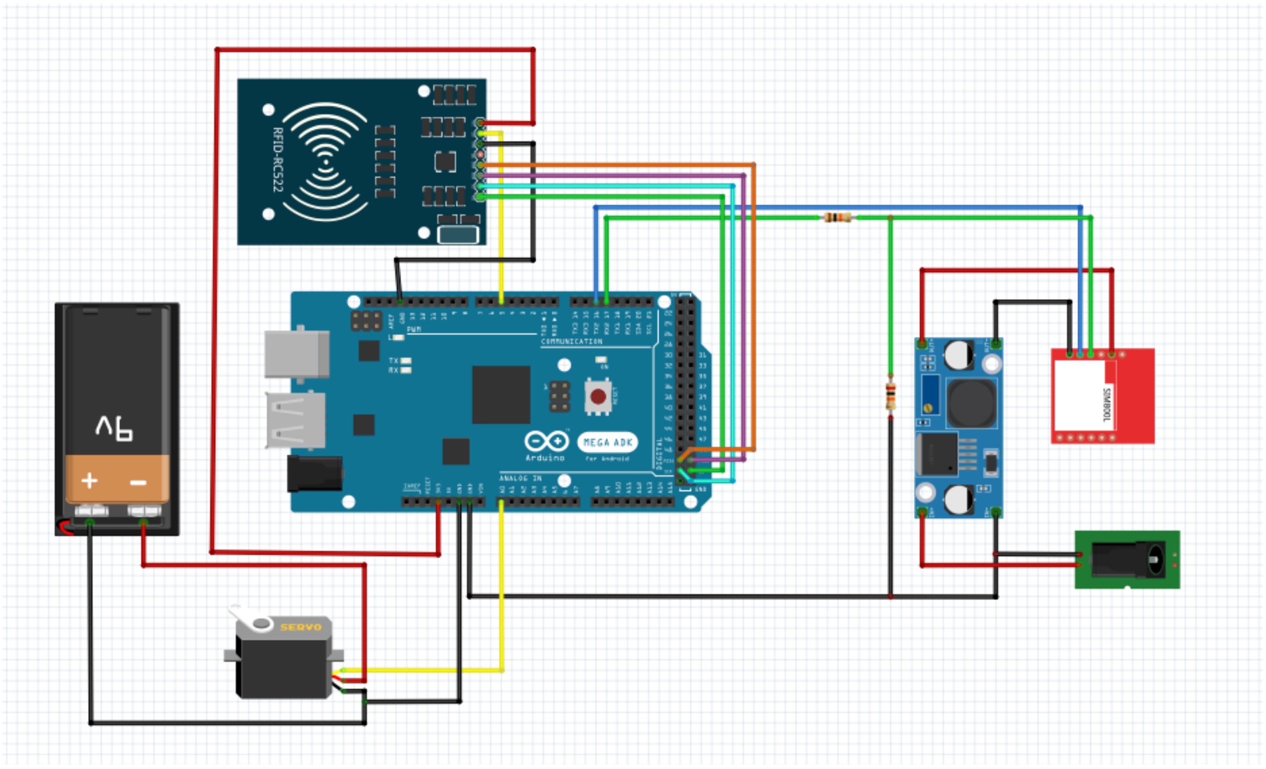


Fig 5.9: Circuit Diagram of Modified Project

Connections-

The VCC and GND of the 9V battery in connected to the VCC and GND of the servo the GND is connected to the GND of the Arduino. The pulse pin of the servo goes to pin A0 of Arduino. The VCC and GND of the power supply is connected to the VCCin an GNDin of the LM2596, the VCCout and GNDout goes to the VCC and GND of the GSM. The RXD pin of the GSM goes through a 10k resistor and then to pin D17(RX2) of the Arduino and this wire is connected to the GND of the power supply through a 20k resistor. The TXD pin of the GSM goes to pin D16(TX2) of Arduino. The VCC and GND of the RFID goes to the 3.3v VCC and GND of the Arduino respectively. The RST pin goes to the pin D5 of Arduino. The MISO, MOSI, SCK, SDK pins of the RFID connects to pin D50, pin D51, pin D52, pin D53 respectively.

- **Modification Made to the Blynk App**

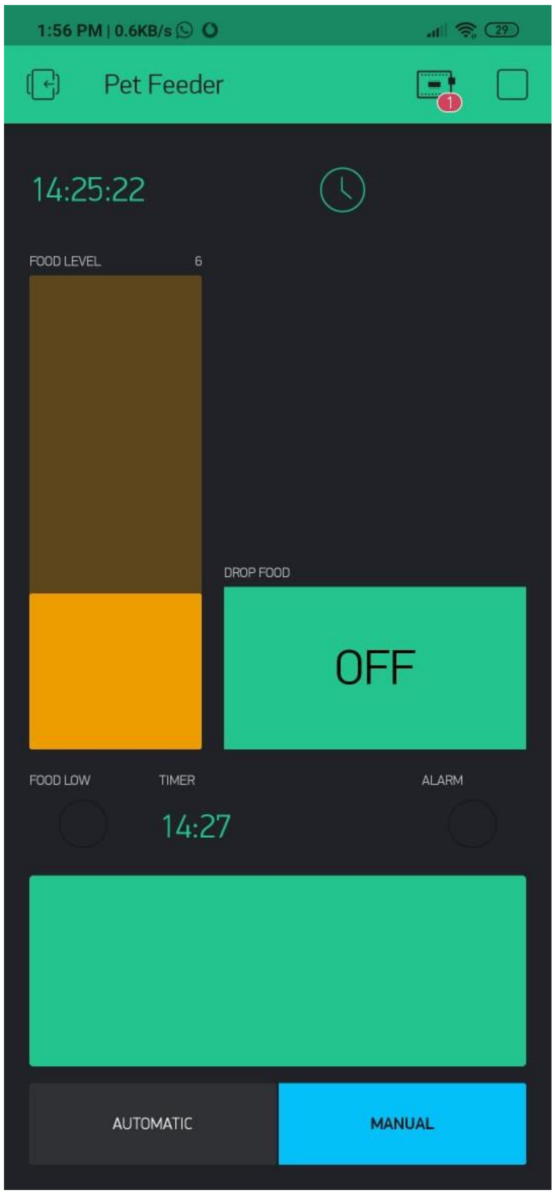


Fig 5.10: Upgraded app

• Explanation of Modified Project

▪ Integration with Existing IOT Project

The existing project has one major flaw, the system will drop food if anyone motion is detected through the PIR sensor. So, Active RFID is used to see if the motion detected by the PIR sensor is a pet of not. The pet will have the RFID Transponder in the collar which send signals at certain intervals of time, with this different type of food can be given to different pets. The SIM800L GSM Module is used to send SMS messages to the owner when the pet eats the food on its own and when the food is low. The LM2596 Module is used for voltage regulation for the GSM Module. Another servo motor is attached to facilitate two kinds of food for two different pets. The LCD Widget is added to the Blynk App to display when and which pet came and ate the food.

▪ Explanation of Working principle with Modification

This system uses a Arduino Mega, PIR Sensor, Ultrasonic sensor, Servo motors, Voltage regulators, ESP8266 WIFI Module, Active RFID and a GSM Module as the working modules.

This system is an automatic pet feeder and it is implemented in two modes AUTOMATIC and MAUNAL modes.

. In the AUTOMATIC mode there is a RFID Transponder in the collar of the pets which send signals at certain intervals of time, when the pet comes near the system the PIR sensor senses its motion then activates the RFID Receiver. The receiver takes the id from the transponder and matches the id of the associated pet and then with help of servo motors the food is dropped to the food bowl below. With the help of RFID different kinds of food and be given to different pets. When the food is dropped through the servo motor a SMS is send to the owner of the pets through the GSM module saying that your pet has eaten its food and it also displays which pet ate and when in the LCD screen in the Blynk app which is sent through the ESP8266 module, it stores and displays few previous data.

The ultrasonic sensor is used to check the levels of food in the container and is displayed the food level in the Blynk app, if the food level is low then it will send a SMS to the owner.

In the MAUNAL mode the user can control the system during this mode the PIR and RFID modules will not be activated. In this there is a real time clock which show the time. The user can drop food for the pet manually by clicking a button, he can also set a specific time at which the food should be dropped and at that given time the food will dropped without the user clicking and buttons. When the timer is set an orange led will light up showing that the timer is activated and after the set time the led will turn off.

- **Pseudo code for the Modified Project**

```
#define BLYNK_PRINT Serial

#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
#include <TimeLib.h>
#include <WidgetRTC.h>
#include <Servo.h>
BlynkTimer timer; // Announcing the timer
BlynkTimer timer1;
WidgetLED led1(V5);
WidgetLED led2(V3);
WidgetRTC rtc;

Servo myservo; // create servo object to control a servo

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "_WB1FljY_e9kRS3QJIIUeUS5uOAst0bm";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Alex's Note 9";
char pass[] = "alex1235";
```

```
int ledPin = 13;                // choose the pin for the LED
int inputPin = 8;               // choose the input pin (for PIR sensor)
int val = 0;                    // variable for reading the pin status
int t1;
String time1;
String time2;
String currentTime;
int trigPin = (11); // Add the Trig pin on pin 10.
int echoPin = (10); // Add the ECHO pin on pin 9.
int duration, distance; // Add types 'duration' and 'distance'.
int count = 0;

// Hardware Serial on Mega, Leonardo, Micro...
#define EspSerial Serial1

// or Software Serial on Uno, Nano...
//#include <SoftwareSerial.h>
//SoftwareSerial EspSerial(2, 3); // RX, TX

// Your ESP8266 baud rate:
#define ESP8266_BAUD 115200

ESP8266 wifi(&EspSerial);

void setup()
{
  // Debug console
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare sensor as input
  pinMode (trigPin, OUTPUT);// Same as above, the TRIG pin will send the ultrasonic wave.
  pinMode (echoPin, INPUT); // The ECHO pin will recieve the rebounded wave, so it must
be an input type.+
  pinMode (7 , OUTPUT);
  digitalWrite(7, HIGH);
  pinMode (6 , OUTPUT);
  digitalWrite(6, HIGH);
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  myservo.write(0);
  // Set ESP8266 baud rate
  EspSerial.begin(ESP8266_BAUD);
  delay(10);

  Blynk.begin(auth, wifi, ssid, pass);
  rtc.begin();
  // You can also specify server:
  //Blynk.begin(auth, wifi, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, wifi, ssid, pass, IPAddress(192,168,1,100),
  8080); timer.setInterval(2000, runn);

}
void up() {
  myservo.write(0);
  timer.setTimeout(2000, down);
}
void down() {
  myservo.write(180);
}
void runn() {
  val = digitalRead(inputPin); // read input value
  if (val == HIGH) {          // check if the input is HIGH
```

```
    digitalWrite(ledPin, HIGH); // turn LED ON
    Blynk.notify("Alex detected");
    timer.setTimeout(10, up);
  } else {
    digitalWrite(ledPin, LOW); // turn LED OFF
  }
}
BLYNK_WRITE(V0) {
  int buttonState = param.asInt();
  if (buttonState == 1) {
    timer.setTimeout(100, up);
  }
}
BLYNK_WRITE(V6) {
  switch (param.asInt()) {
    case 1: {
        Serial.println("Item 1");
        inputPin = 8;
        break;
      }
    case 2: {
        Serial.println("Item 2");
        inputPin = 3;
        break;
      }
  }
}
void check() {
  currentTime = String(hour()) + ":" + minute();
  Serial.println(currentTime);
  if (time1 == currentTime) {
    if (timer.isEnabled(t1) ) {
      timer1.setTimeout(10, up);
      led1.off();
      timer.disable(t1);
      Serial.println("disable");
    }
  } else {
    Serial.print("Bad");
  }
}
BLYNK_WRITE(V4) {
  TimeInputParam t(param);

  if (t.hasStartTime())
  {
    time1 = String(t.getStartHour()) + ":" + String(t.getStartMinute());
    Serial.println(time1);
    led1.on();
    t1 = timer.setInterval(5000, check);
  }
}
BLYNK_READ(V1){
  time2 = String(hour()) + ":" + minute() + ":" + second();
  Blynk.virtualWrite(V1, time2);
}
BLYNK_READ(V2)
{
  int distance = ultra();
  int level = distance;
  // This command writes Arduino's uptime in seconds to Virtual Pin (5)
  Blynk.virtualWrite(V2, level);
  if (distance >= 7) {
     if (count == 0) {
      Blynk.notify("Food is low");
    }
    led2.on();
    count = 1;
  }
  else        {
    led2.off();
    count = 0;
  }
}
int ultra()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(20);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(100);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in
  microseconds duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  return distance;
}
void loop()
{
  Blynk.run();
  timer.run();
  timer1.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```

<u>Code for RFID Tag: -</u>

```
        #include <SPI.h>
        #include <MFRC522.h>


        #define SS_PIN 5
        #define RST_PIN 52
        MFRC522 mfrc522(SS_PIN, RST_PIN);     // Create MFRC522 instance.


        void setup() {
        Serial.begin(9600);    // Initialize serial communications with the PC
        SPI.begin();    // Init SPI bus
        mfrc522.PCD_Init(); // Init MFRC522 card Serial.println("Scan PICC to see UID and
        type...");
}
void loop() {
        // Look for new cards
        if ( ! mfrc522.PICC_IsNewCardPresent()) {
                return;
        }


        // Select one of the cards
        if ( ! mfrc522.PICC_ReadCardSerial()) {
```

```
                    return;
          }


//Dump debug info about the card. PICC_HaltA() is automatically called.

mfrc522.PICC_DumpToSerial(&(mfrc522.uid));

}
```

<u>Code for GSM module:</u>

```
          #include <SoftwareSerial.h>



          SoftwareSerialmySerial(16,17);



          void setup()

          {

mySerial.begin(9600); // Setting the baud rate of GSM Module Serial.begin(9600); // Setting
the baud rate of Serial Monitor (Arduino) delay(100);





void loop()

{

  if (Serial.available()>0)

switch(Serial.read())

  {

    case 's':

SendMessage();

      break;

    case 'r':

RecieveMessage();
```

```
      break;
  }


 if (mySerial.available()>0)
Serial.write(mySerial.read());
}



 void SendMessage()
{
mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text
Mode delay(1000); // Delay of 1000 milli seconds or 1 second
        mySerial.println("AT+CMGS=\"+91xxxxxxxxxx\"\r"); // Replace x with
        mobile number delay(1000);
        mySerial.println("I am SMS from GSM Module");// The SMS text you
        want to send delay(100);
        mySerial.println((char)26);// ASCII code of CTRL+Z
        delay(1000);
        }



         void RecieveMessage()
        {
        mySerial.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS
        delay(1000);
         }
```

AT Commands to Send SMS using Arduino and GSM Module

AT+CMGF=1 // Set the GSM module in text mode

AT+CMGS=\"+YYxxxxxxxxxx\"\r // Input the mobile number| YY is country code “the message” with stopping character (char)26 // ASCII of ctrl+z

AT Commands to Receive SMS using Arduino and GSM Module

AT+CMGF=1 // Set the GSM Module in text mod

AT+CNMI=2,2,0,0,0 // AT Command to receive live sms.

Chapter 6

Conclusion

This project is an automated as well as manual system with a very user-friendly interface and easy to use system which assists people who are busy with work and are not able to take care of their pets. And at the same time, they can set certain times when your pet should eat.

- **Chapter 7**

Future Scope

The system can be further enhanced in the future by adding

- Machine Learning algorithm where camera module will be used to detect different animals and differentiate between pets like cats and dogs and can also differentiate with respect to their ages to provide variable amount of food.
- Different varieties of food like milk, water, wet and dry food according to animal's needs.
- Local servers to provide wireless connection within a limited range in remote areas where internet access cease to exist.
- Monitoring feature which be used to analyse the behaviour of pets and collect data. With the help of the data generated, the machine can provide food adequately and can also keep track of its diet.

Chapter 8

References

[1]. Mritunjay Subhashchandra Tiwari and Sahil Manoj Hawal and Nikhil Navanath Mhatre and Akshay Ramesh Bhosale and Mainak Bhaumik "Automatic Pet Feeder Using Arduino" International Journal of Innovative Research in Science, Engineering and Technology, Vol. 7 (3), March 2018.

[2]. Chung-Ming Own, Haw-Yunshin, Chen-Ya Teng (2013), 'The Study & Application of the IoT in Pet Systems' Vol.1, No.3, PP 1-8, http://dx.doi.org.

[3]. Vania and Kanisius Karyono and Hargyo Tri Nugroho I. "Smart Dog Feeder Design Using Wireless Communication, MQTT and Android Client" International Conference on Computer, Control, Informatics and its Applications, October 2016.

[4]. Tessema Gelila Berhan and Worku Toyiba Ahemed and Tessema Zelalem Birhan "Programmable Pet Feeder" International Journal of Scientific Engineering and Research, 2014.

[5]. Prashant Singh and Amit Kumar Sharma and Payal Sood and Paramdeep Singh "Remote Controlled and GSM based Automated Pet Feeder" Lovely Professional University, Vol. 2 (2), April, 2015.

[6]. http://docs.blynk.cc/

[7]. https://community.blynk.cc/

[8]. https://examples.blynk.cc/?board=Arduino%20Mega%202560&shield=ESP8266%20 WiFi%20Shield&example=GettingStarted%2FBlynkBlink

[9]. http://help.blynk.cc/en/articles/605485-esp8266-with-at-firmware

[10]. Rachel Hail, Kristine McCarthy, Filip Rege, Alexix Rodrigvez Carlson (2008), 'The Smart Pet Feeder'.