**Project Report**

**on**

**PET FEEDER**

in partial fulfilment for the award of the degree of

**BACHELOR OF ENGINEERING**

IN

**COMPUTER SCIENCE(AIML)**

**Submitted by:**

Parampreet Kaur(24BAI70286)

Navkiran Kaur(24BAI70312)

Swara Verma(24BAI70493)

Runali Behera (24BAI70591)

**Under the Guidance of**

Dr. Divneet Singh Kapoor

**ACADEMIC UNIT-I**

****

**Chandigarh University**

**17April 2025**

**Index**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Content** | **Page No.** |
| 1 | Project Overview | 1 |
| 2 | Objective and Problem Statement | 2 |
| 3 | Proposed Solution & Methodology | 3 |
| 4 | Key Findings / Results | 4,5,6,7 |
| 5 | Conclusion & Learnings | 8 |
| 6 | References | 9 |
| 7 | Appendix (if required) | 10,11 |

**PROJECT OVERVIEW**

1. What is your project about?

Our project is a **Smart Pet Feeder** that dispenses food manually using a mobile app. It allows pet owners to feed their pets remotely with voice assistant.

1. Why is it important?

Pet owners may not always be home or near their pets. This project helps them feed their pets from anywhere, anytime, ensuring convenience and better care.

1. Brief background or context:

Feeding pets on time is often a challenge for busy owners. Manual control through a mobile app provides flexibility and control, especially when feeding times vary day to day.

**OBJECTIVE AND PROBLEM STATEMENT**

1)What problem are you trying to solve?

Difficulty in feeding pets while away from home or on irregular schedules.

2)What are your main goals?

* Enable remote, manual control of pet feeding.
* Provide a simple interface for the user.
* Display real-time data and feedback on feeding status.

**Proposed Solution & Methodology**

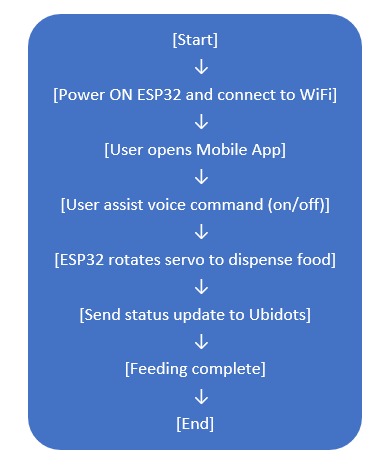
1)What did you plan to do and how?

We built a pet feeder controlled manually via a mobile app made with MIT App Inventor, connected to the ESP32 microcontroller, with data logging using Ubidots.

2)Tools/Software/Materials used:

* ESP32 microcontroller
* Servo motor for food dispensing
* Ubidots (for IoT cloud dashboard)
* MIT App Inventor (for mobile app)
* Arduino IDE (for programming)
* Food container, food dispenser, wood base, pipe, wiring
* power supply (using USB wire via laptop)

Step wise demonstration



**Key Findings / Results**

1)What did you observe, build, or find out?

* The servo motor responds correctly to app commands.
* Food is dispensed instantly when the button is pressed.
* Ubidots dashboard shows successful feed status.
* App works smoothly with WiFi-enabled ESP32.

IMAGES

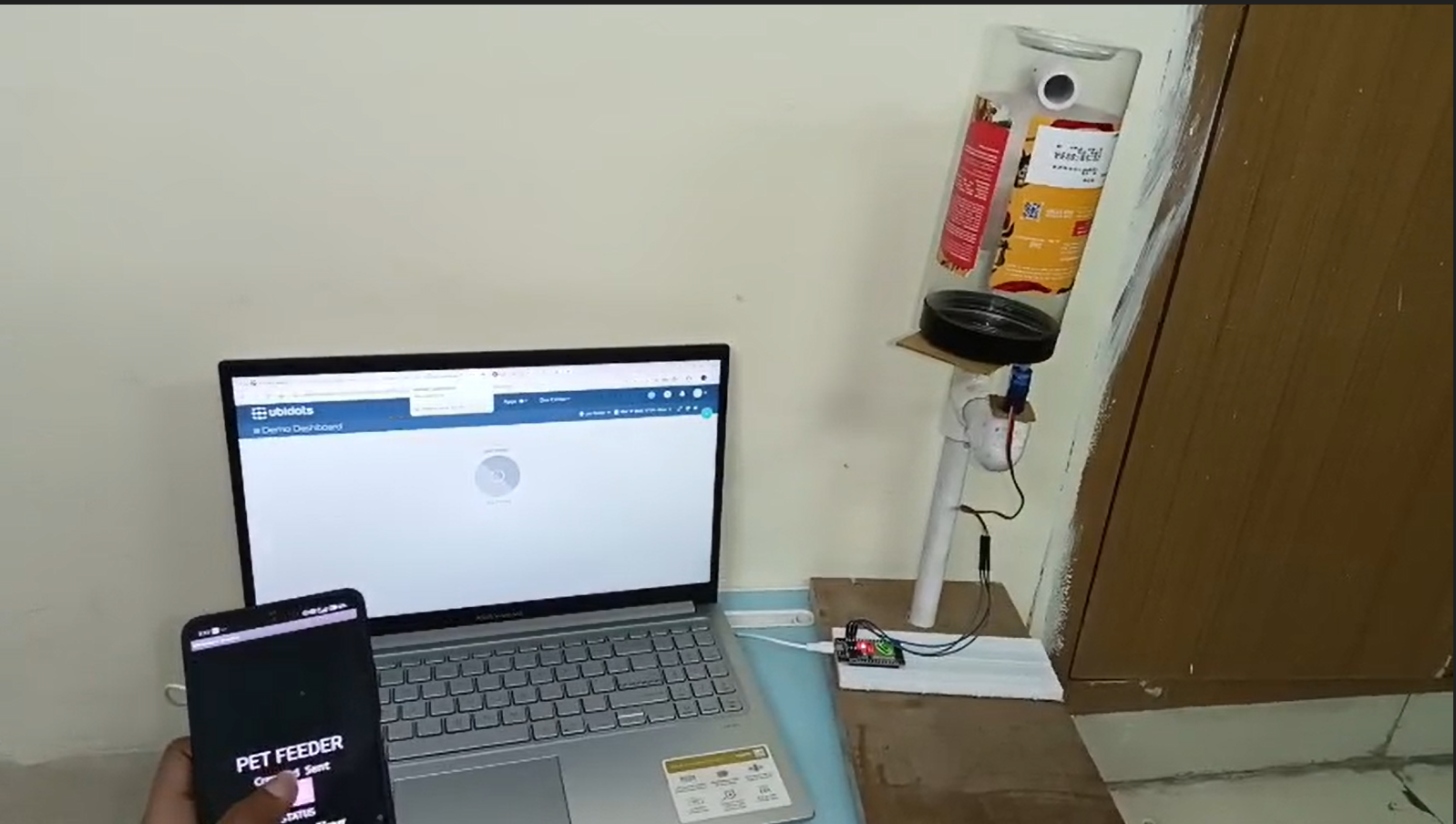
**

Figure 1: During the OFF command

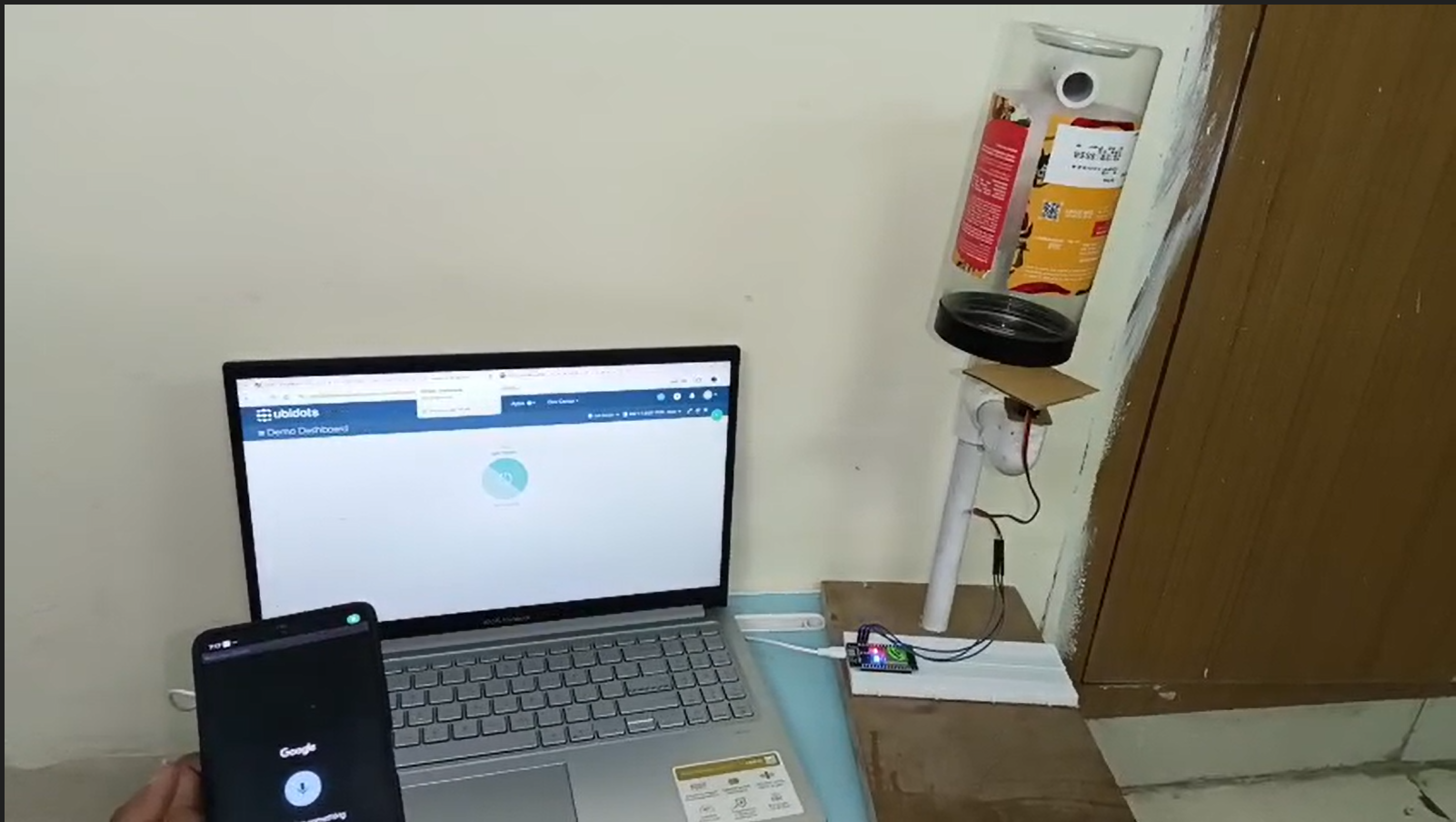
**

Figure2: During the ON command

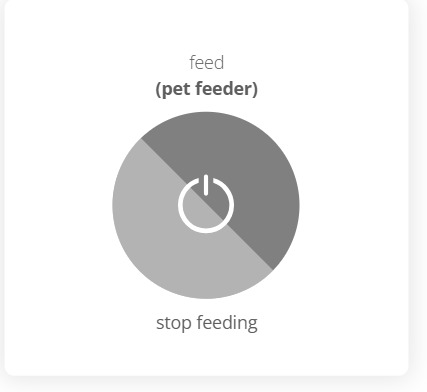
**

Figure4: Ubidots during OFF command

Figure3: MIT App Interface



Figure6: Final Model

**Conclusion & Learnings**

1)What did you learn through this project?

* + How to use ESP32 for IoT applications
  + App creation using MIT App Inventor
  + Real-time communication between mobile app and hardware
  + Use of Ubidots for data visualization

2)Any improvements or next steps?

* + Add feeding schedule automation
  + Notification system after feeding
  + Pet food level detection
  + Integration with Google Assistant or Alexa

**REFRENCES**

[EM & DT](https://sites.google.com/cumail.in/dt2-2024/) (for understanding the concept )

[https://www.arduino.cc/en/software ( arduino](https://www.arduino.cc/en/software%20(%20arduino) uno)

[IoT - Google Drive](https://drive.google.com/drive/folders/11Hwk3zPujy-jY2mgyJEMdJZRDc9_QwEm) (code)

[Ubidots | Dashboards](https://stem.ubidots.com/app/dashboards/) (dashboard for petfeeder)

<https://ai2.appinventor.mit.edu> (app)

**APPENDIX**

#include <UbidotsESPMQTT.h>

#include <ESP32Servo.h>

#define RELAY 2

#define SERVO\_PIN 13 // Servo ka pin

#define TOKEN "BBUS-mKqfeexWXEpA5Gfdma395KxXktnqkt" // Ubidots TOKEN

#define WIFISSID "swara" // WiFi SSID

#define WIFIPASS "9876543210" // WiFi Password

Ubidots client(TOKEN);

Servo myServo;

void callback(char\* topic, byte\* payload, unsigned int length) {

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

for (int i = 0; i < length; i++) {

Serial.print((char)payload[i]);

}

Serial.println();

bool command = \*payload - '0'; // Convert char to int

Serial.print("Command: ");

Serial.println(command);

digitalWrite(RELAY, command);

if (command == 1) {

Serial.println("Turning Servo to 90 degrees...");

myServo.write(0); // Servo 90 degree ghoomega

delay(1000); // Thoda rukne de, dekh raha hai kya ho raha hai

} else {

Serial.println("Turning Servo to 0 degrees...");

myServo.write(210); // Wapas 0 degree par

delay(1000);

}

}

void setup() {

Serial.begin(9600);

Serial.println("Init... T4\_Smart\_Home");

pinMode(RELAY, OUTPUT);

myServo.attach(SERVO\_PIN);

myServo.write(0); // Start position

Serial.print("Connecting to SSID: ");

Serial.print(WIFISSID);

client.wifiConnection(WIFISSID, WIFIPASS);

Serial.println("Done");

Serial.println("Initializing Ubidots Connection...");

client.ubidotsSetBroker("industrial.api.ubidots.com");

client.setDebug(true);

client.begin(callback);

client.ubidotsSubscribe("pet-feeder","feed");

Serial.println("Done");

}

void loop() {

if (!client.connected()) {

client.reconnect();

client.ubidotsSubscribe("pet-feeder","feed");

}

client.loop();

delay(1000);

}