Smart Aquarium using Raspberry PICO

Abstract: In comparison to other pets, fish require more intensive care. Fish had been abandoned due to a lack of maintenance, such as water level or timely feeding in the fish tank. One of the solutions to the problems is an Io T-based smart aquarium system. This research exhibits a prototype of an IOT-based Smart Aquarium that has been developed.

1. Introduction

The aquarium system now on the market is a standard aquarium. The aquarium system on the market today is already interesting, with a collection of plants and fish in the tank, but the issues if the aquarium is kept indoors the issue is still-manual distribution of fish feed. At least once a day, aquarium owners must personally feed their fish. This is especially problematic if the aquarium owner is extremely busy, as the danger of forgetting to feed the fish is very high.To overcome the existing issues, an aquarium system that is both visually appealing and entirely automated in terms of water-filling and fish feeding when required. A smart aquarium system that can accommodate aquarium hobbyists and all of the aquarium owner's activities.One of the technologies designed to make it easier for humans to monitor and control an internet-based system is IoT technology. The Internet of Things (IoT) is a system in which things are given unique identities and the ability to move data over a network without requiring human-to-human or human-to-computer interaction in both directions. This technology is supposed to aid in the monitoring of fish they grow.The goal of this project is to make aquarium monitoring and maintenance easier. Using the Picoboard, tasks like, testing water level, fish feeding,water pumping etc., can be automated. This project has 3main functionalities: timed fish feeding using a real-time clock, water pumping,and water level monitoring.. There is Ethernet connectivity, where the Ethernet module acts as a server and it communicates with the picoboard over serial and mobile device over local network. You can control and monitor the smart aquarium via the android app MQTT voice. The app will allow you to control the feeding the fish automatically when its required.

2. Methodology

Tools used to make this model are:

- Picoboard.
- Android device.
- Servo Motor.
- Single channel Relay – 5V.
- Breadboard.
- Aquarium Bowl (For implementation).
- Wires, Resistors, led etc.

- Male to Female, Female to Female Breadboard Jumper wires.
- Water Level Sensor.
- Automatic feeder system.
- 5v pump
- Dc jack
- 5v adaptor

3.1.1. Picoboard

The W5100S-EVB-Pico is a micro controller evaluation board based on the Raspberry Pi RP2040andthefully hardwired TCP/IP controller W5100S. It functions similarly to the Raspberry Pi Pico board but adds Ethernet via the W5100S.

Fig..1. Pico board

2.1.2 Android Device

Basically an android phone as an interface to work with the application to control the feeding functions of the smart aquarium.

2.1.3. Single Channel Relay Module

The 1-Channel Relay Driver Module is a load tripping mechanism used to derive load based on setting. It can operate on 5V, this makes it compatible. A 1-channel relay is an elector-mechanical relay operated on an electrical signal provided to it. Switches inside the relay make it useful for quick switching of load.

Fig..2:- Single channel relay

2.1.4 Servo motor

A servo motor is a rotary actuator that allows for precise control of angular position. It consists of a motor coupled to a sensor for position feedback.

Fig.3:Servo motor

2.1.5 Water level Sensor

To detect the required water level in the aquarium.

Fig. 4:. Water level Sensor

2.1.6 (5V)Water pump

The 5v pump helps in pumping the water to the aquarium.

Fig .5

2.1.6 Resistor

330k resistor has been used to blink the led light.

Fig. 6

3. Circuit Diagram

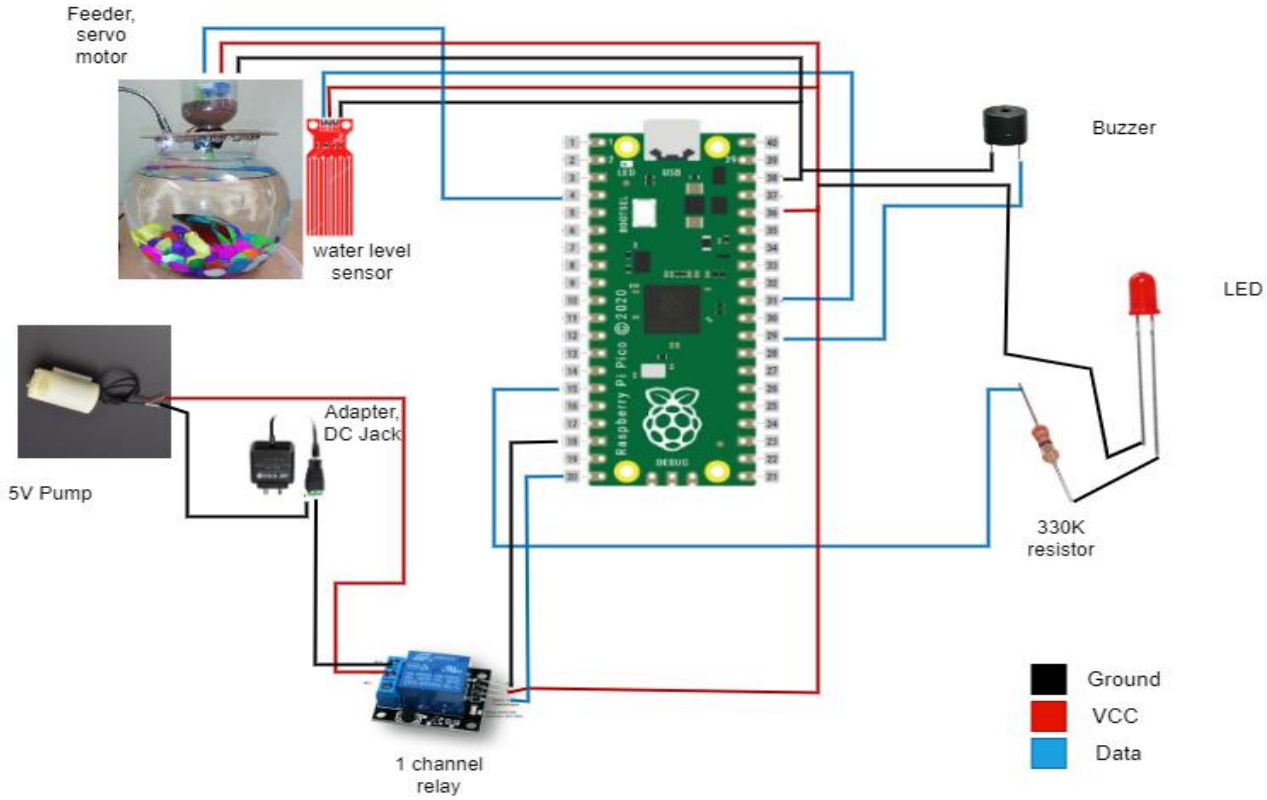

Fig.7

4. AI connection

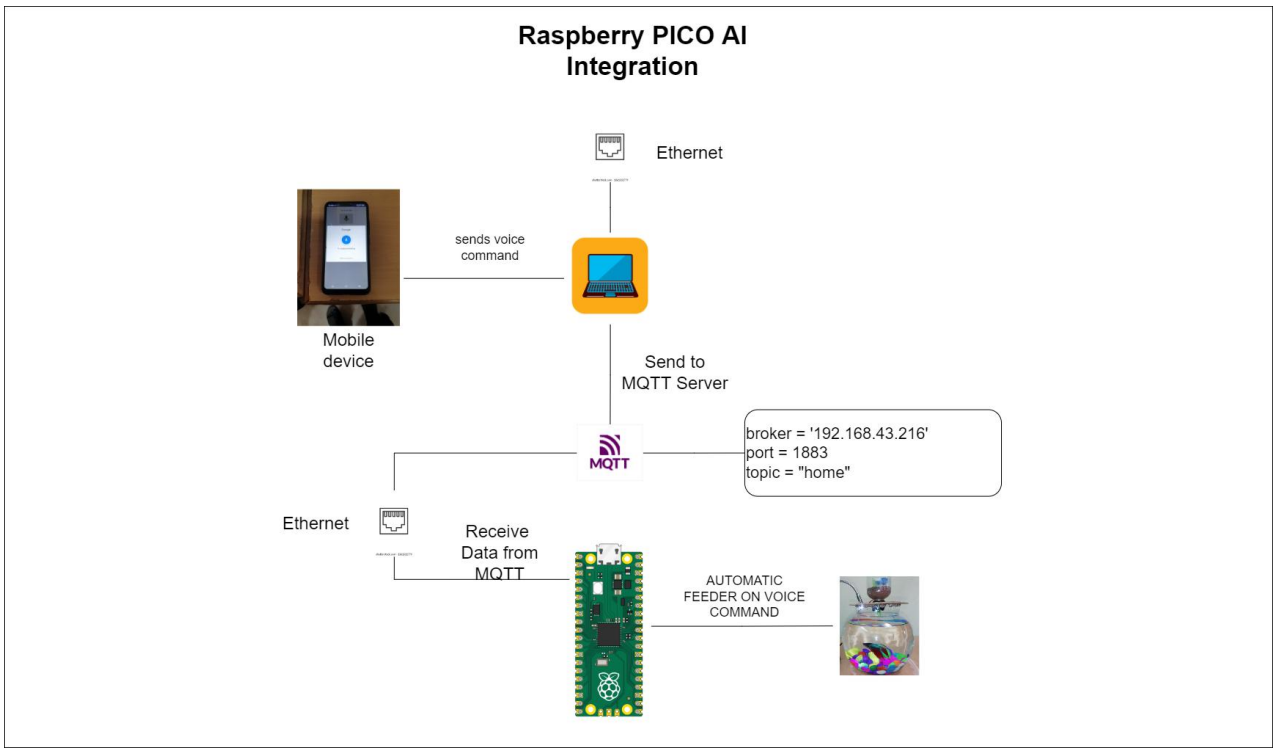

Fig.8

5. Code

```
//----------------------------------------//
//--------SMART AQUARIUM-------------------//
//----------------------------------------//
#define sensorPin 26
#include <Servo.h>
#include <SPI.h>
#include <Wire.h>
#include <Ethernet.h>
#include <PubSubClient.h>
#include <Servo.h>
int val2 = 0;
int servo=0;
int pos;

Servo myservo;  // create servo object to control a servo
//MQTT Global Variables
byte mac[]    = {  0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
IPAddress ip(172, 16, 0, 100);
IPAddress myDns(192, 168, 0, 1);
IPAddress server(10, 5, 15, 103);
EthernetClient ethClient;
PubSubClient client(ethClient);

//MQTT callback funtion
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  char messageBuffer[30];
  memcpy(messageBuffer, payload, length);
  messageBuffer[length] = '\0';
  Serial.println(messageBuffer);
  for (int i=0;i<length;i++) {
    Serial.print((char)payload[i]);
  }

  //For convering payload to string
  char message[30];
  for(int i=0;i<length;i++){
```

```
37    message[i]=payload[i];
38    }
39   String msg = String(message);
40   if((msg.indexOf("feed")>-1)||(msg.indexOf("Feed")>-1))
41   servo=1;
42   else
43   servo=0;
44 }
45
46 //MQTT Reconnect callback
47 void reconnect() {
48   while (!client.connected()) {
49   // Loop until we're reconnected
50     Serial.print("Attempting MQTT connection...");
51     // Attempt to connect
52     if (client.connect("arduinoClient")) {
53       Serial.println("connected");
54       client.subscribe("home/aqua");
55     } else {
56       Serial.print("failed, rc=");
57       Serial.print(client.state());
58       Serial.println(" try again in 5 seconds");
59       // Wait 5 seconds before retrying
60       delay(5000);
61     }
62   }
63 }
64 void ethstart()
65 {
66   //ethernet codes
67   Ethernet.init(17);  // WIZnet W5100S-EVB-Pico
68   Serial.begin(115200);
69   while (!Serial) {
70     ; // wait for serial port to connect. Needed for native USB port only
71   }
72
73     Serial.println("Initialize Ethernet with DHCP:");
74   if (Ethernet.begin(mac) == 0) {
75     Serial.println("Failed to configure Ethernet using DHCP");
76     // Check for Ethernet hardware present
```

```
77    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
78      Serial.println("Ethernet shield was not found.  Sorry, can't run without hardware. :(");
79      while (true) {
80        delay(1); // do nothing, no point running without Ethernet hardware
81      }
82    }
83    if (Ethernet.linkStatus() == LinkOFF) {
84      Serial.println("Ethernet cable is not connected.");
85    }
86    // try to congifure using IP address instead of DHCP:
87    Ethernet.begin(mac, ip, myDns);
88  } else {
89    Serial.print("  DHCP assigned IP ");
90    Serial.println(Ethernet.localIP());
91  }
92  // give the Ethernet shield a second to initialize:
93  delay(1000);
94  Serial.print("connecting to ");
95  Serial.print(server);
96  Serial.println("...");
97  client.setClient(ethClient);
98  client.setServer(server, 1883);
99  client.setCallback(callback);
100   // Allow the hardware to sort itself out
101   delay(1500);
102
103 }
104
105 void setup() {
106   ethstart(); //calling ethernet connection function
107   myservo.attach(2); //data pin of servo
108   pinMode(15,OUTPUT);//pump
109  pinMode(sensorPin, INPUT);
110 //  pinMode(9,OUTPUT);
111
112   pinMode(12,OUTPUT); //red led glows when the water is reached maximum level
113   pinMode(13,OUTPUT); //buzzer - when the water is low in fish bowl
114    myservo.attach(2);
115  Serial.begin(9600);
116 }
```

```
117
118 int start_servo()
119   {
120    if(servo==1)
121   {
122    for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
123    // in steps of 1 degree
124    myservo.write(pos);
125    delay(2);// tell servo to go to position in variable 'pos'
126                          // waits 15ms for the servo to reach the position
127   }
128   for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
129    myservo.write(pos);
130    delay(2);// tell servo to go to position in variable 'pos'
131                          // waits 15ms for the servo to reach the position
132   }
133   }
134   servo=0;
135   return 0;
136   }
137
138
139 void loop() {
140   int pumpstatus=0;
141   if (!client.connected()) {
142    reconnect();
143    }
144    //Subscribe to topic
145    client.subscribe("home");
146
147    start_servo();
148
149    val2 = analogRead(sensorPin);
150   Serial.print("Water level: ");
151   Serial.println(val2);
152    if((val2>=400)){
153    digitalWrite(12,HIGH); //red
154    digitalWrite(13,LOW); //buzzer
155    delay(100);
156    pumpstatus=1;
```

```
157    }
158   else{
159    pumpstatus=0;
160    digitalWrite(12,LOW);
161    digitalWrite(13,HIGH);
162    delay(100);
163    }
164   if(pumpstatus==0)
165   {
166    digitalWrite(15,HIGH);
167    delay(5000);
168    digitalWrite(15,LOW);
169    }
170    else
171    digitalWrite(15,LOW);
172
173    client.loop();
174 }
```

6. Code Explanation:

The code starts with including all the libraries required for the components. All global variables and definitions for the MQTT server are initialized(Line no 14-19). Followed by form line no 22 - 44 MQTT callback function which is used for checking for new messages which is sent by publisher. In line 40 passing a keyword feed/Feed so when the publisher/user gives the command feed/Feed then the feeder will start working. Line no 47-60 MQTT reconnect callback. Line no 64-103 is of Ethernet connection, It tries to connect to the Ethernet, but if there are any problems, it sends a warning message to the serial port. Line 105-116 void setup(), this function in which we are setting the data pin for servo motor. Pin mode for pump, red led, buzzer. Line 118-136, int start_servo function telling servo to go to position in variable 'pos'. To for loop conditions one goes from 0 degrees to 180 degrees another goes from 180 degrees to 0 degrees. Line 139-174 void loop() function, it contains the if condition to check for MQTT connection is connected to subscribe "home". Then if condition to pump the water to start if water is less in fish bowl with buzzer high and to stop pumping water if water has reached maximum level with red led glowing to indicate maximum water level has reached.

7. AI Implementation

The AI implemented by voice recognition for the smart aquarium, is to automate the feeding system for the fish.

We have used a raspberry pi local MQTT server for in-taking voice command and to activate feeder system. Firstly downloaded a MQTT voice app from play store. Then we have give the credentials MQTT Server:Port and MQTT Topic. MQTT Server:Port where the IP address and port number has to given. MQTT topic foe this project is “home”. The figure for this as follows

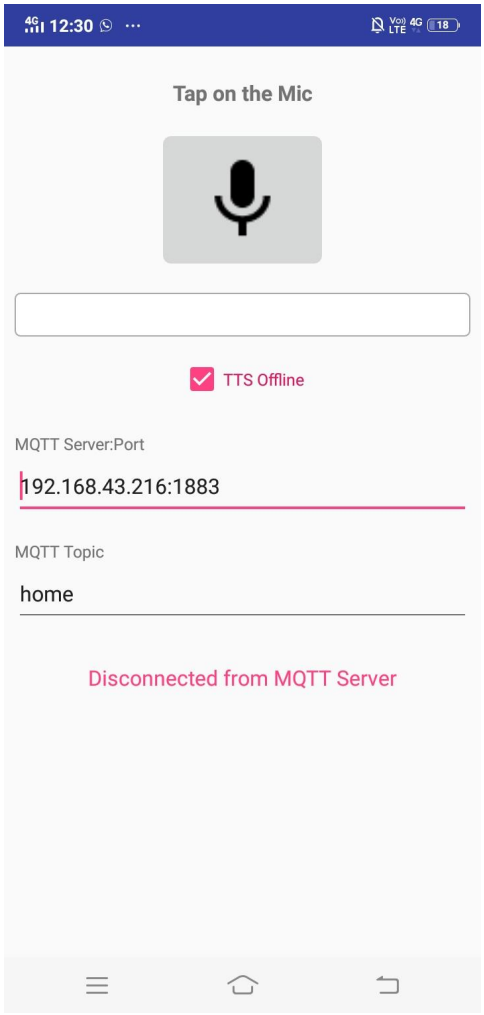

Fig. 9

By clicking on the mic we need to send command feed/Feed so that the feeder gets activated. The given command will be checked if its matching with the command given in code, when it matches feeder gets activated.

8. Images

Fig.10

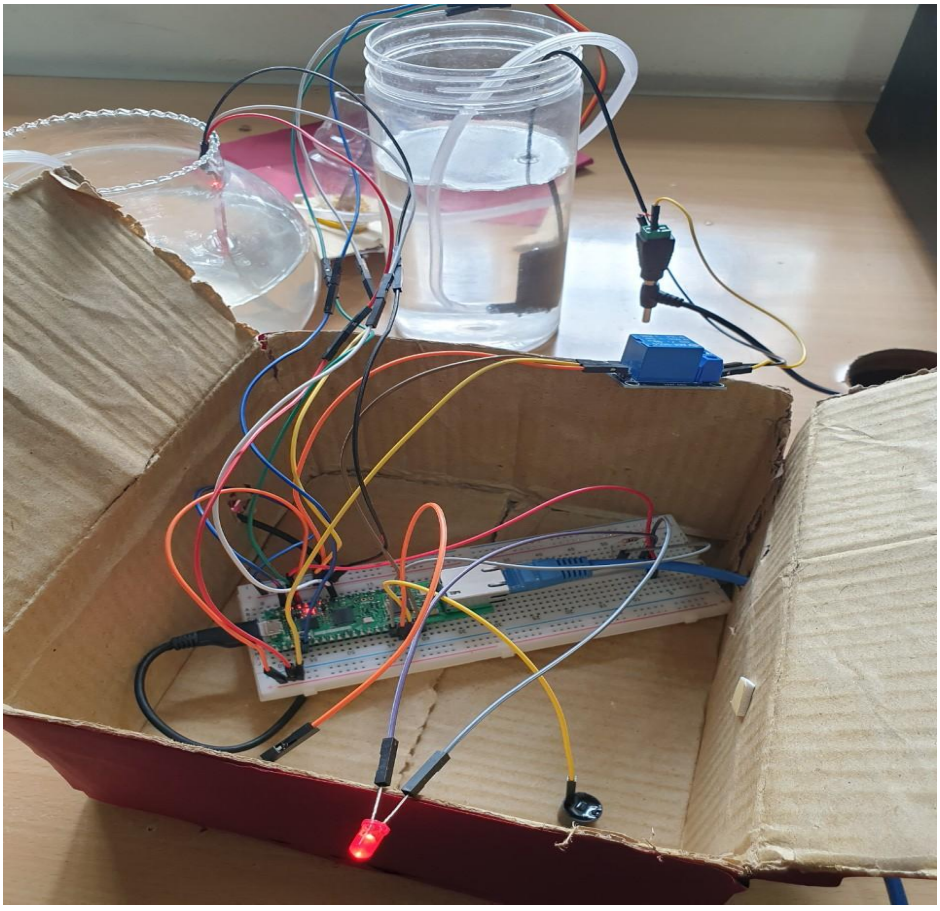

Fig.11

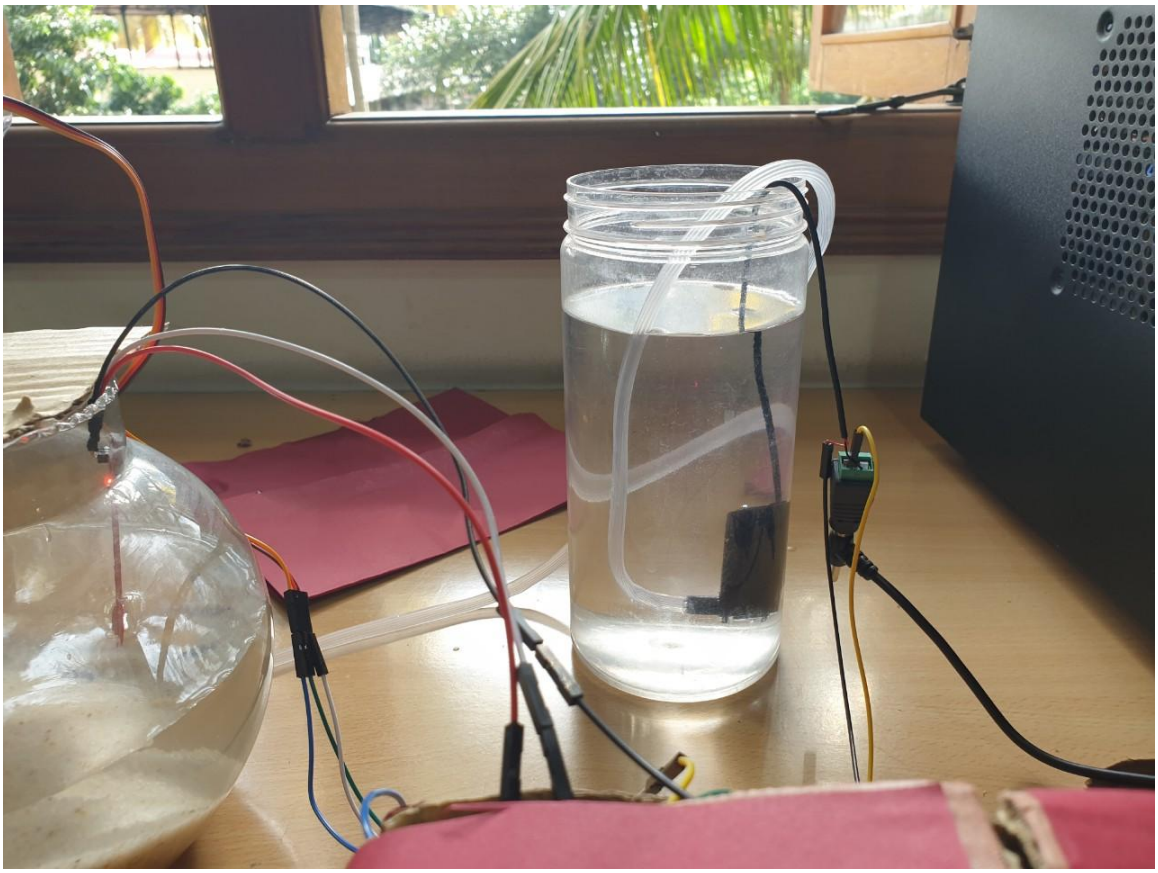

Fig.12

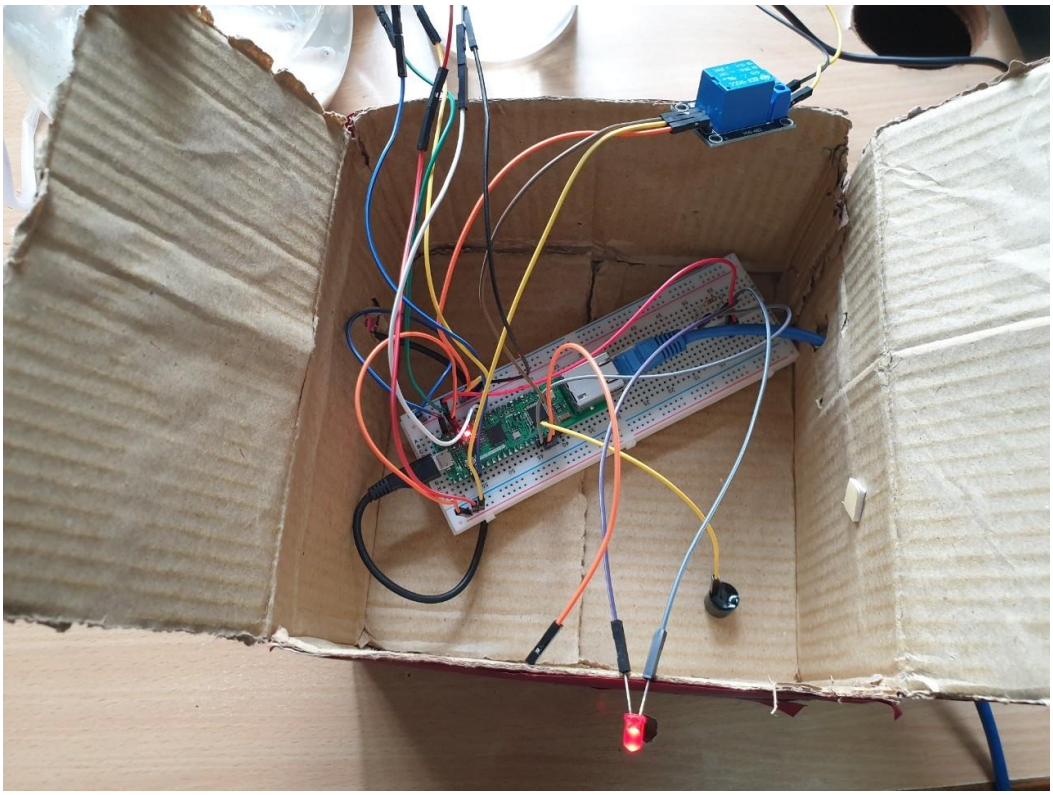

Fig.13

Fig.14

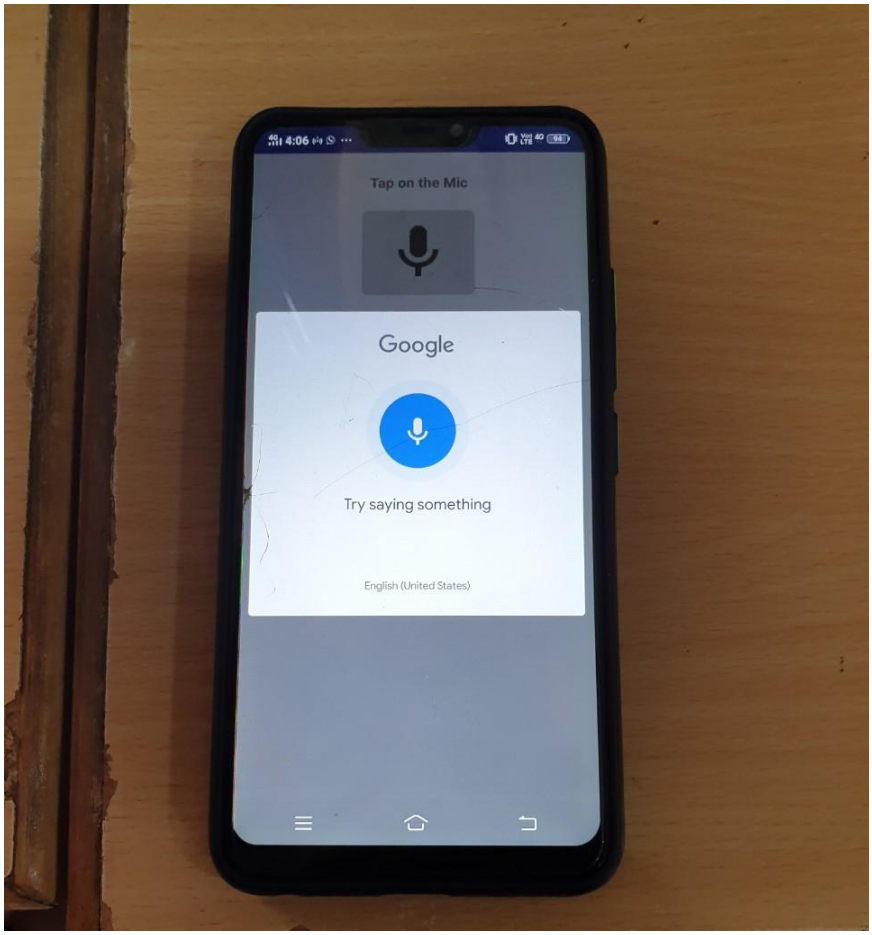
Tap on the Mic
Google
Try saying something
English (United States)

Fig.15

9. Conclusions

The project was inspired by an idea, to create a system that could automatically take care of the fish and the ability to be able to control these devices remotely using android devices.. Now most of the operations happen by themselves, water level,automatic feeding,automatic pumping of water.. By using an IoT platform, it can monitor these variables, visualize the data and even control some features manually, over the internet which is successfully implemented in this project. Another important feature was the mechanical design and implementation of the fish feeding system, which is an original design. It is a rather simple design, but it efficiently does the job. There are other designs but they are complicated, so this project achieves simplicity, efficiency, time saving and cost saving. This project serves as a way to practically implement our skills to solve a very important management related problem and assist in achieving an ideal environment for fishes in the aquarium.

10. References

[1] https://create.arduino.cc/projecthub/fadytarek1995/aquarium-assistant-b7ff20

[2]https://www.researchgate.net/publication/337567490_Implementation_of_Smart_Aquarium_System_Supporting_Remote_Monitoring_and_Controlling_of_Functions_using_Internet_of_Things

[3]https://www.researchgate.net/publication/345690191_Smart_Aquarium_Management_System[4]https://all3dp.com/2/raspberry-pi-pico-projects/