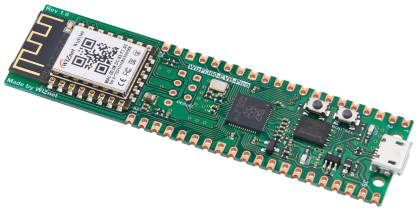
SMART PET HOME

I.Abstract

Smart technology is developing swiftly, opening up to exceptional uses that will further streamline and enhance daily life. While owners are out of their house, dogs can stay in smart doghouses in a safe and comfortable environment.The internet-connected smart doghouse has internal camera that allow for remote monitoring using a device.the smart pet home consist of automated light system which works on the condition of human presence. The light turns on when human is close to a certain distance with respect to the distance to a smart pet home. The distance is measured and condition is set with the ultrasonic sensors. And the smart pet home consist of feature of automatic water feeder to the dog when its required. The automatic water feeder works with the sensor readings from the water level sensor and the water pumping works by respective to the sensor reading with conditions provided. The AI Implementation is done with detection of human near the smart pet home makes the light turn on inside the smart pet home.

1. Requirements

1.1 Hardware

Name	Image	Description
WizFi360-EVB-Pico		Based on the Raspberry Pi RP2040, WizFi360-EVB-Pico offers Wi-Fi connectivity. It is suitable for use in the creation of IoT solutions and is pin compatible with the Raspberry Pi Pico board.

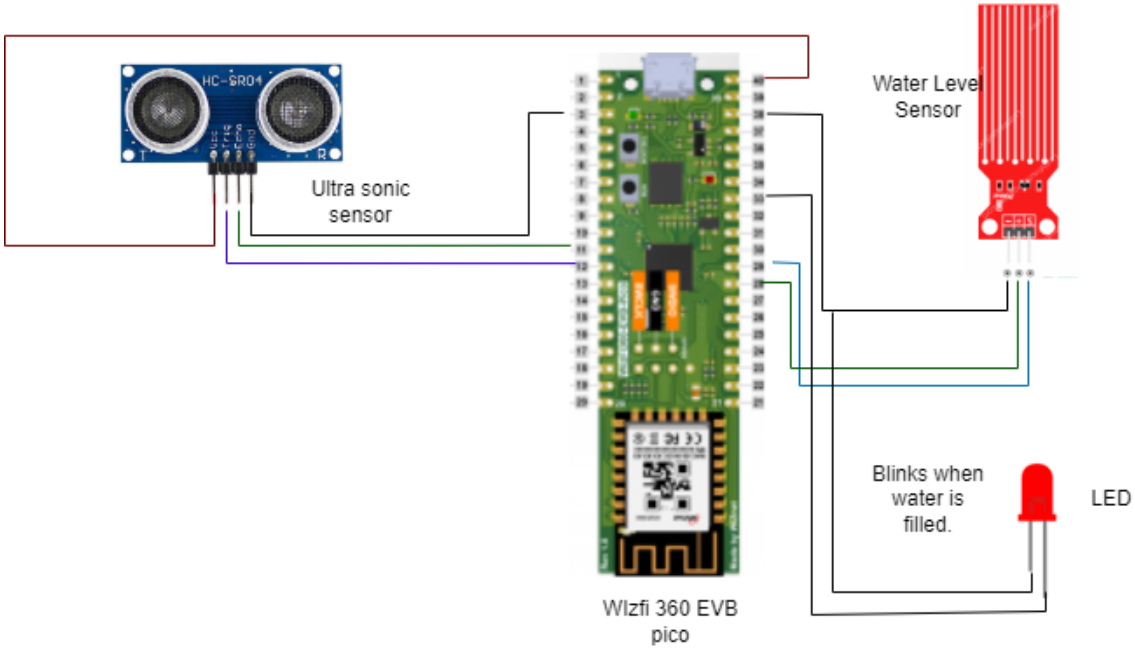
Water-Level-Sensor		The water level sensor is a device that measures the liquid level in a fixed container that is too high or too low.
Ultrasonic-Sensor		The Ultrasonic Sensor is used to measure the distance from a pet home and according to distance all the automatic functions will operate.
Jumper-Wires		It is nothing, just an electric wire that connects remote electric circuits It is used to connect all the given sensors and Pico Board together.
Breadboard		It is a solderless device for temporary prototype with electronics and test circuit designs

1.2 SOFTWARE

- Arduino-IDE
- Thingspeak

- WizFi360 Libraries

2. Circuit diagram



2.1 Pin configuration

Ultrasonic Sensor

VCC - Connected to pin VBUS.

Trig - Connected to pin GP12.

Echo - Connected to pin GP11.

GND - It can be Connected to any ground of pico board.

Water Level Sensor

Consists of 3 pins Positive , Negative and Signal.

Positive - Connected to VCC.

Negative - Connected to GND.

Signal - Its a data pin connected to analog pin ADC0.

LED

It consists of 2 pins positive and negative. Where positive is connected to pin GP13 and negative is connected to GND.

3. Code

3.1 Code Explanation

```
1. /*
2.  WizFi360 example: Thingspeak
3.
4.  This sketch Request to thingspeak using a WizFi360 module to
5.  perform a simple store and search data.
6. */
7.
8. #include <stdlib.h>
9. #include <DHT.h>
10.
11. #include "WizFi360.h"
12. const int trig = 12; //Trigger pin of ultrasonic Sesnor
13. const int echo = 11; //Echo pin of ultrasonic Sesnor
14. long time_taken;
15. int dist,distance;
16.
17. const int analogInPin = A0;
18. int sensorValue = 0;
19. int a;
20. const int buzzer = 15;
21.
22. // setup according to the device you use
23. #define WIZFI360_EVB_PICO
24.
25. // Emulate Serial1 on pins 6/7 if not present
26. // remember that the wiznet is connected to 4 and 5 not 6 and 7
27. #ifndef HAVE_HWSERIAL1
28. #include "SoftwareSerial.h"
29. #if defined(ARDUINO_MEGA_2560)
30. SoftwareSerial Serial1(4, 5); // RX, TX
31. #elif defined(WIZFI360_EVB_PICO)
32. SoftwareSerial Serial2(4, 5); // RX, TX
33. #endif
34. #endif
35.
36. /* Baudrate */
37. #define SERIAL_BAUDRATE   115200
```

```
38. #if defined(ARDUINO_MEGA_2560)
39. #define SERIAL1_BAUDRATE  115200
40. #elif defined(WIZFI360_EVB_PICO)
41. #define SERIAL2_BAUDRATE  115200
42. #endif
43.
44. /* Sensor */
45. #define DHTTYPE DHT11
46. #define DHTPIN SDA
47. #define CDSPIN A0
48.
49. /* Wi-Fi info */
50. char ssid[] = "LAPTOP-GRDK2B0B 4358";       // your network SSID
(name)
51. char pass[] = "R223u51*";   // your network password
52.
53. int status = WL_IDLE_STATUS;  // the Wifi radio's status
54.
55. char server[] = "api.thingspeak.com"; // server address
56. String apiKey ="9HM2EATBZJ1UPFGD";                    // apki
key
57.
58. // sensor buffer
59. char temp_buf[10];
60. char humi_buf[10];
61. char cds_buf[10];
62.
63. unsigned long lastConnectionTime = 0;         // last time you
connected to the server, in milliseconds
64. const unsigned long postingInterval = 30000L; // delay between
updates, in milliseconds
65.
66. // Initialize the Ethernet client object
67. WiFiClient client;
68. // Initialize the DHT object
69. DHT dht(DHTPIN, DHTTYPE);
70.
71. void setup() {
72.   //initialize sensor
73.   pinMode(CDSPIN, INPUT);
```

```
74.
75.   pinMode(13, OUTPUT);//tells the arduino to get ready to
exchange messages with the serial monitor  at a data rate of 9600
bits/sec.
76.   pinMode(trig, OUTPUT); //configures the specified pin to
behave as output/input.
77.   pinMode(echo, INPUT);
78.   pinMode(buzzer, OUTPUT);
79.   dht.begin();
80.
81.   // initialize serial for debugging
82.   Serial.begin(SERIAL_BAUDRATE);
83.   // initialize serial for WizFi360 module
84. #if defined(ARDUINO_MEGA_2560)
85.   Serial1.begin(SERIAL1_BAUDRATE);
86. #elif defined(WIZFI360_EVB_PICO)
87.   Serial2.begin(SERIAL2_BAUDRATE);
88. #endif
89.   // initialize WizFi360 module
90. #if defined(ARDUINO_MEGA_2560)
91.   WiFi.init(&Serial1);
92. #elif defined(WIZFI360_EVB_PICO)
93.   WiFi.init(&Serial2);
94. #endif
95.
96.   // check for the presence of the shield
97.   if (WiFi.status() == WL_NO_SHIELD) {
98.     Serial.println("WiFi shield not present");
99.     // don't continue
100.     while (true);
101.   }
102.
103.   // attempt to connect to WiFi network
104.   while ( status != WL_CONNECTED) {
105.     Serial.print("Attempting to connect to WPA SSID: ");
106.     Serial.println(ssid);
107.     // Connect to WPA/WPA2 network
108.     status = WiFi.begin(ssid, pass);
109.   }
110.   Serial.println("You're connected to the network");
```

```
111.   printWifiStatus();
112.   thingspeakTrans();
113. }
114.
115. void calculate_distance(int trigger, int echo)
116. {
117.   digitalWrite(trigger, LOW);
118.   delayMicroseconds(2);
119.   digitalWrite(trigger, HIGH);
120.   delayMicroseconds(10);
121.   digitalWrite(trigger, LOW);
122.
123.   time_taken = pulseIn(echo, HIGH);
124.   dist= time_taken*0.034/2;
125.
126. }
127.
128. void loop() {
129.
130.   calculate_distance(trig, echo);
131.   distance=dist; //getting distance of ultrasonic sensor
132.   if(distance < 15 )
133.   {
134.     digitalWrite(13, HIGH);   // turn the LED on (HIGH is the
voltage level)
135.     delay(1000);
136.   }
137.   else
138.   {
139.     digitalWrite(13, LOW);    // turn the LED off by making the
voltage LOW
140.     delay(1000);
141.   }
142.   Serial.println(distance);
143.   delay(1000);
144.   water();
145.   // if 30 seconds have passed since your last connection,
146.   // then connect again and send data
147.   if (millis() - lastConnectionTime > postingInterval) {
148.     //sensorRead();
```

```
    thingspeakTrans();
  }
}


//Transmitting sensor value to thingspeak
void thingspeakTrans() {

  // close any connection before send a new request
  // this will free the socket on the WiFi shield
  client.stop();
  calculate_distance(trig, echo);

  distance=dist;

  // if there's a successful connection
  if (client.connect(server, 80)) {
    Serial.println("Connecting...");
    Serial.println(distance);
    client.print(F("GET /update?api_key="));
    client.print(apiKey);
    client.print(F("&field1="));
    client.print(distance);
    client.print(F("&field2="));
    client.print(water());
    client.println();
    lastConnectionTime = millis();
  }
  else {
    // if you couldn't make a connection
    Serial.println("Connection failed");
  }
}

void printWifiStatus() {
  // print the SSID of the network you're attached to
  Serial.print("SSID: ");
    //IPAddress ip = WiFi.localIP();
  Serial.println(WiFi.SSID());
 Serial.println("STOPED HERE  .... ");
```

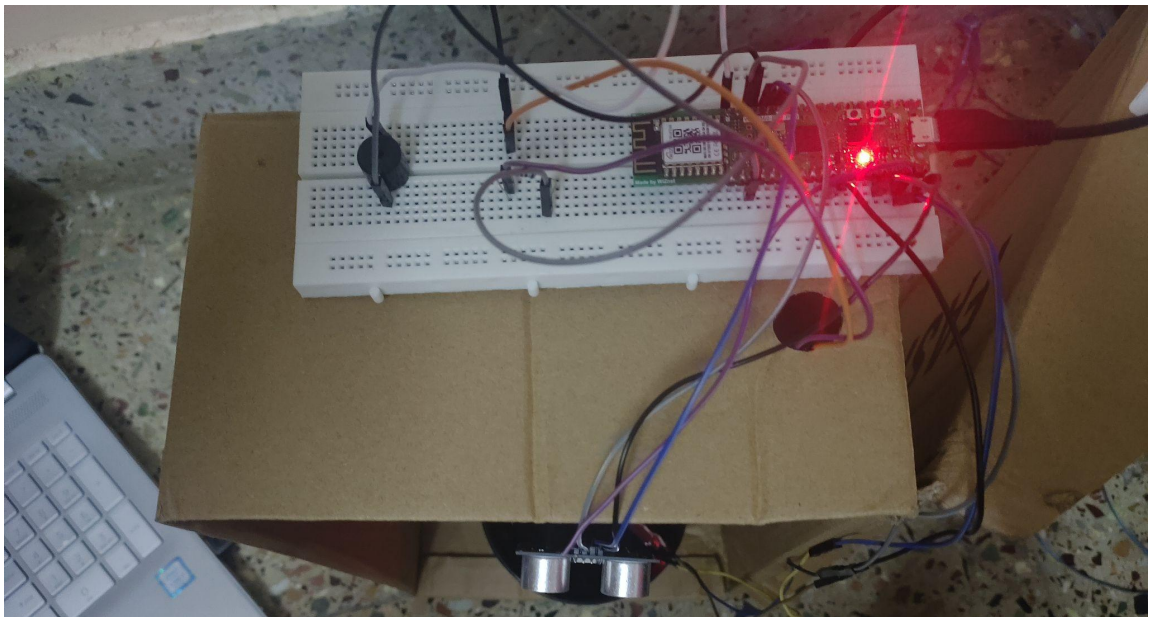
```
189.   // print your WiFi shield's IP address
190.   //IPAddress ip = WiFi.localIP();
191.   Serial.print("IP Address: ");
192.   //Serial.println(ip);
193.
194.   // print the received signal strength
195.   //long rssi = WiFi.RSSI();
196.   Serial.print("Signal strength (RSSI):");
197.   //Serial.print(rssi);
198.   Serial.println(" dBm");
199. }
200.
201. int water(){
202.   sensorValue = analogRead(analogInPin);
203. Serial.print("Sensor = " );
204. a=sensorValue*100/1024;
205. Serial.print(a);
206. Serial.println("%");
207. if(a < 20){
208.   digitalWrite(buzzer,HIGH);
209. }
210. else
211. {
212.   digitalWrite(buzzer,LOW);
213. }
214. return a;
215. }
216.
217.
```

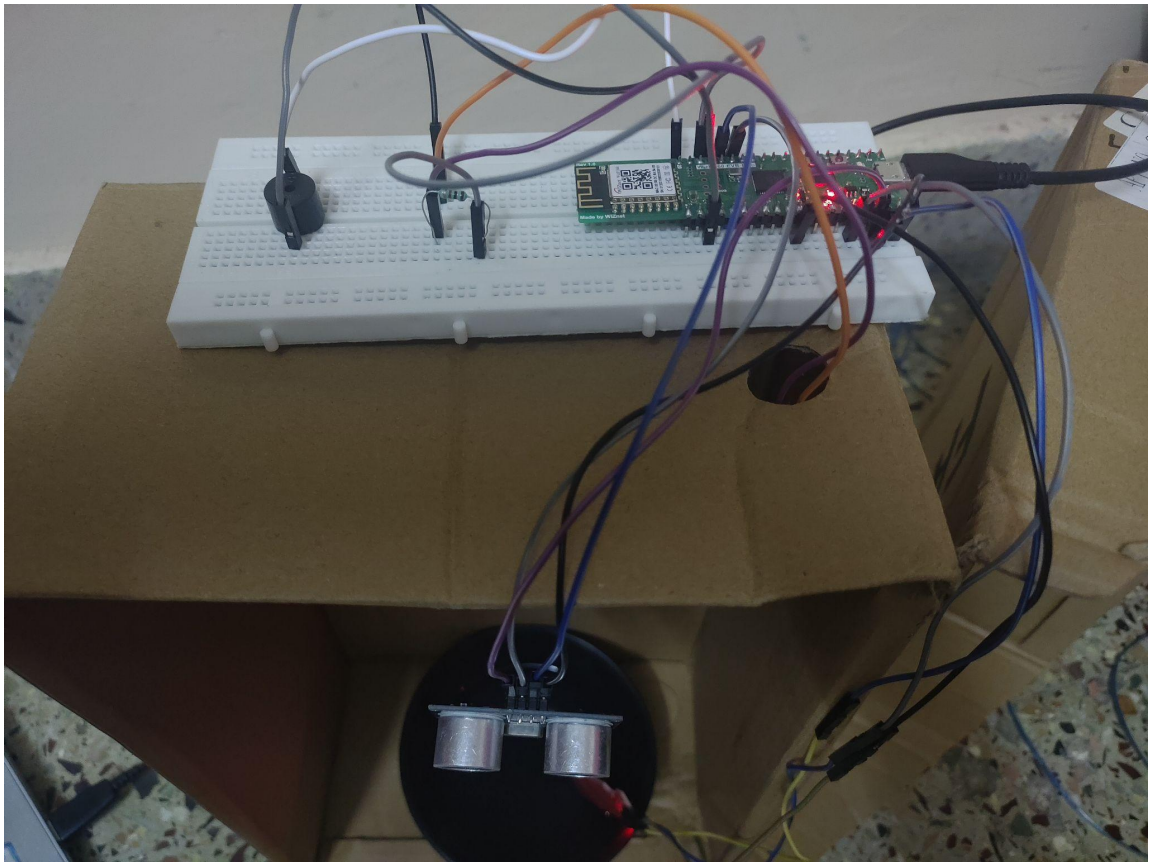
Explanation:

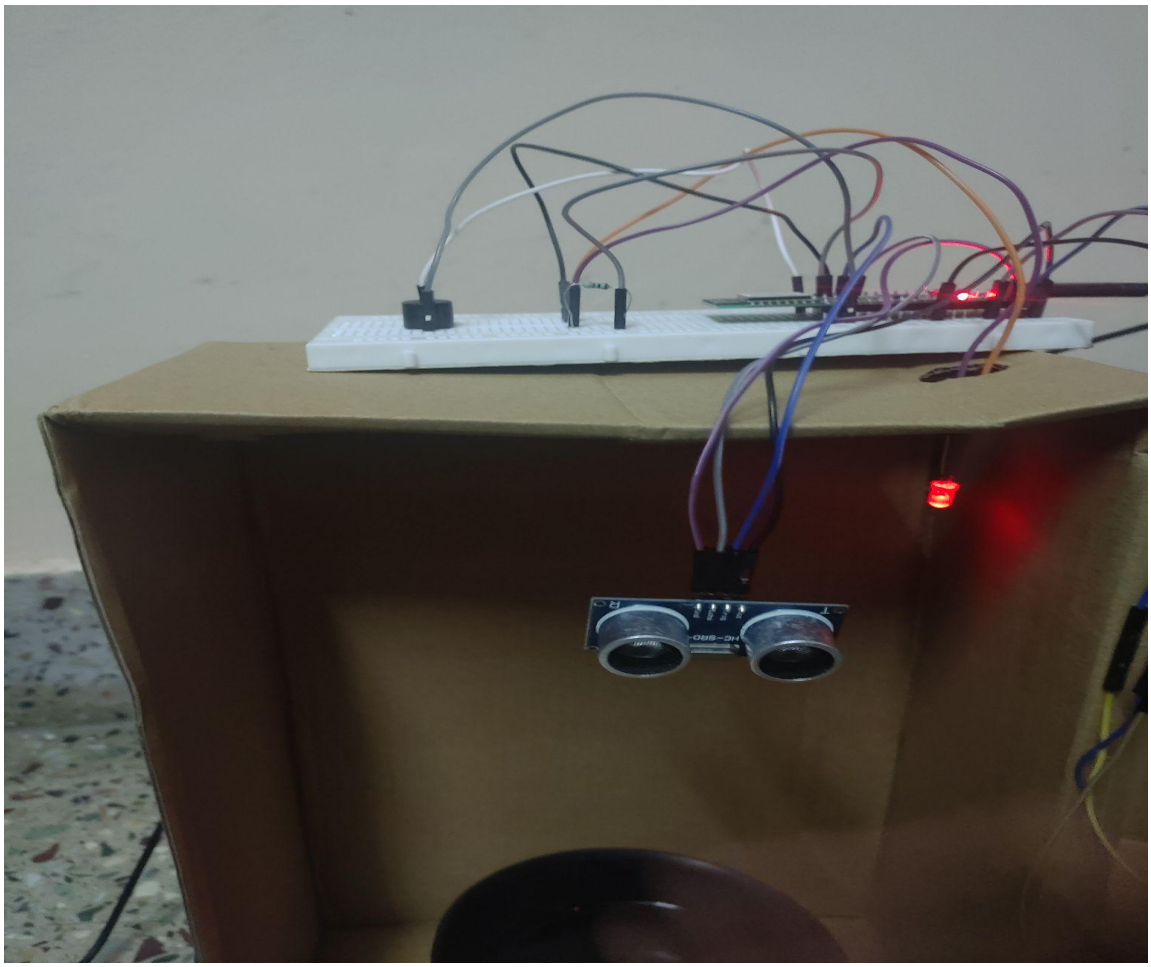
From the above code Line[12 - 20] initializing the variables for led, buzzer distance and analog sensor for water level sensor. Line[50 - 51] is the WIFI ID and Password. Line[55 - 56] thingspeak server and api key. Line[115 - 126] calculate distance function in which the data read from ultrasonic sensor is taken and calculate the distance. Line[128 - 151] void loop where distance and water level function is called. Line[201 - 215] analog data

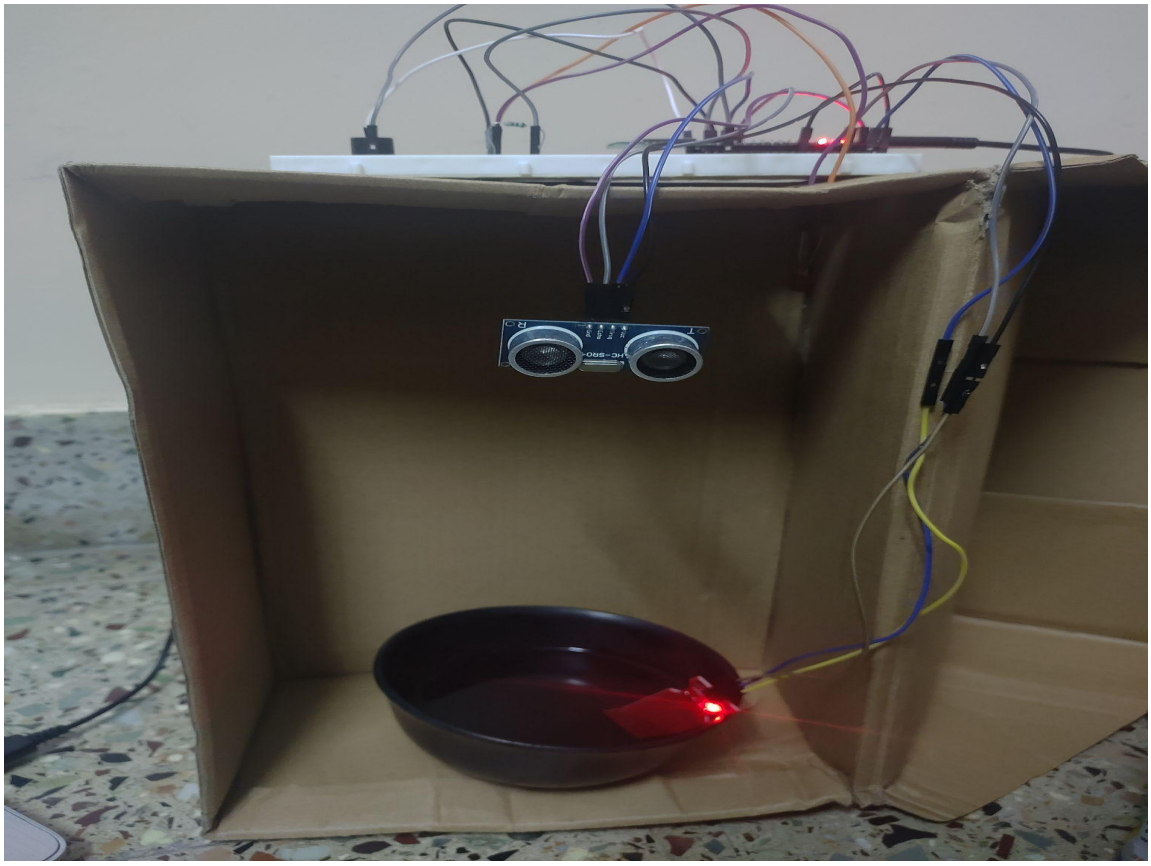
received from water level sensor based on data is less than 20 buzzer is high else buzzer is low.

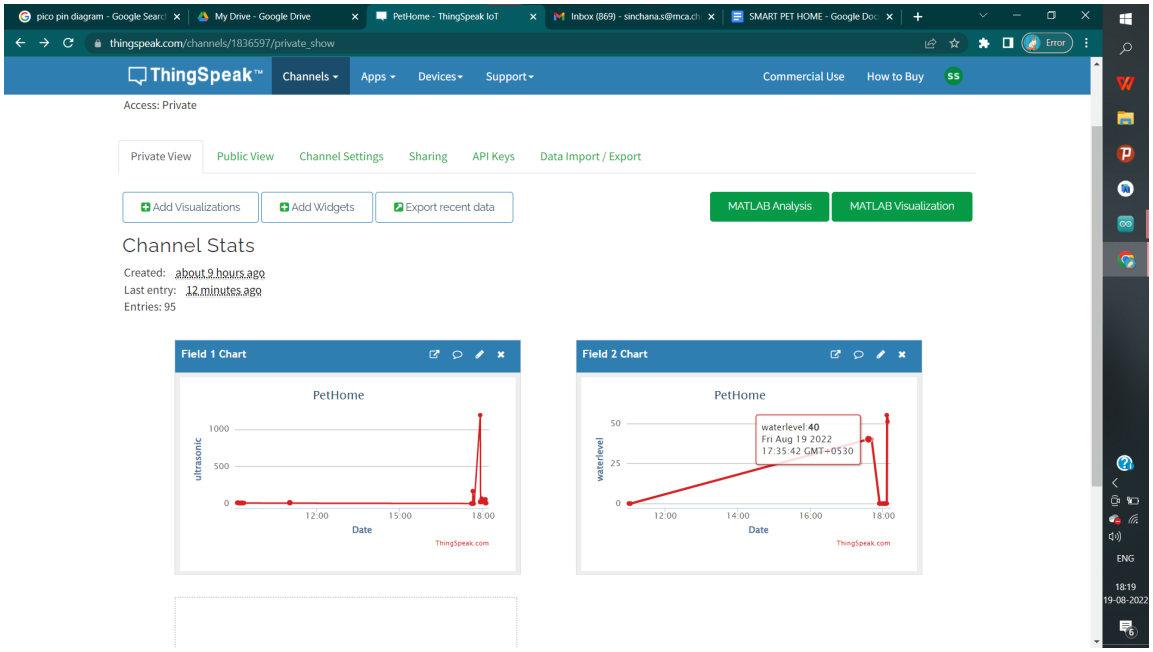
4. Images










pico pin diagram - Google Searc
My Drive - Google Drive
PetHome - ThingSpeak IoT
Inbox (869) - sinchana.s@mca.ch
SMART PET HOME - Google Doc
thingspeak.com/channels/1836597/private_show
ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
Access: Private
Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Add Visualizations
Add Widgets
Export recent data
MATLAB Analysis
MATLAB Visualization
Channel Stats
Created: about 9 hours ago
Last entry: 12 minutes ago
Entries: 95
Field 1 Chart
PetHome
ultrasonic
1000
500
0
12:00
15:00
18:00
Date
ThingSpeak.com
Field 2 Chart
PetHome
waterlevel
50
25
0
12:00
14:00
16:00
18:00
Date
waterlevel: 40
Fri Aug 19 2022
17:35:42 GMT+0530
ThingSpeak.com
ENG
18:19
19-08-2022

5. References

[1]Luayon, Alexis Anne A., et al. "PetCare: a smart pet care IoT mobile application." *Proceedings of the 10th International Conference on E-Education, E-Business, E-Management and E-Learning*. 2019.

[2] Kim, Seungcheon. "Smart pet care system using internet of things." *International Journal of Smart Home* 10.3 (2016): 211-218.

[3] Tsai, Ming-Fong, et al. "Multiple Feature Dependency Detection for Deep Learning Technology—Smart Pet Surveillance System Implementation." *Electronics* 9.9 (2020): 1387.