SMART HOME

A Project report submitted in partial fulfilment

of the requirements for the degree of B. Tech in Electronic and Communication Engineering by

ANUDEEP BATHINA(18BEC006)
JEEVAN REVANEPPA HIRETHANAD(18BEC017)
KIRAN D(18BEC022)
NIKHIL S A(18BEC033)
NITHIN R(18BEC034)

Under the guidance of

DR. JAGADEESHA R BHATT, ASST.PROFESSOR

DEPARTMENT

OF

ELECTRONICS AND COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,DHARWAD

JANUARY 2021 - MAY 2021

CONTENTS :

SL.NO	TOPIC
1	Abstract
2	Introduction
3	Problem Definition
4	Proposed Model
5	Advantages of Home automation systems
6	Components
7	Block Diagram
8	Features of the our home automation system
9	Codes
10	Implementation on software using ML
11	Results
12	References

ABSTRACT :

-Withadvancement of Automation technology, life is getting simpler and easier in all aspects. In today's world Automatic systems are being preferred over manual systems. With the rapid increase in the number of users of the internet over the past decade has made the Internet a part and parcel of life, and IoT is the latest and emerging internet technology. The Internet of things is a growing network of everyday objects-from industrial machines to consumer goods that can share information and complete tasks while you are busy with other activities. Wireless Home Automation system(WHAS) using IoT is a system that uses computers or mobile devices to control basic home functions and features automatically through the internet from anywhere around the world, an automated home is sometimes called a smart home. It is meant to save electric power and human energy. The home automation system differs from other systems by allowing the user to operate the system from anywhere around the world through internet connection.

INTRODUCTION :

Overview:

Homes of the 21st century will become more and more self controlled and automated due to the comfort it provides, especially when employed in a private home. A home automation system is a means that allows users to control electric appliances of varying kinds. Many existing, well-established home automation systems are based on wired communication. This does not pose a problem until the system is planned well in advance and installed during the physical construction of the building. But for already existing buildings the implementation cost goes very high.
In contrast, Wireless systems can be of great help for automation systems. With the advancement of wireless technologies such as Wi-Fi, cloud networks in the recent past, wireless systems are used every day and everywhere.

Problem Definition:

Home automation systems face four main challenges, these are high cost of ownership, inflexibility, poor manageability, and difficulty in achieving security. The main objective of this project is to design and implement a home automation system using IoT that is capable of controlling and automating most of the house appliances through an easy manageable web interface. The proposed system has a great flexibility by using Wi-Fi technology to interconnect its distributed sensors to home automation servers. This will decrease the deployment cost and will increase the ability of upgrading, and system reconfiguration.

Proposed Model :

Raspberry pi is used to collect the data from all the sensors like MQ05 - temperature & humidity sensor , HC-SR04 -ultra sonic sensor, Gas sensor. And all the data will be accessed through the internet from thingspeak cloud platform. Cayenne is used to control the relay module.And arduino uno is used to convert analog data into digital.

Advantages of Home automation systems:

In recent years, wireless systems like Wi-Fi have become more and more common in home networking. Also in home and building automation systems, the use of wireless technologies gives several advantages that could not be achieved using a wired network only.

1) Reduced installation costs: First and foremost, installation costs are significantly reduced since no cabling is necessary. Wired solutions require cabling, where material as well as the professional laying of cables (e.g. into walls) is expensive.

2) System scalability and easy extension: Deploying a wireless network is especially advantageous when, due to new or changed requirements, extension of the network is necessary. In contrast to wired installations, in which cabling extension is tedious. This makes wireless installations a seminal investment.

3) Aesthetical benefits: Apart from covering a larger area, this attribute helps to full aesthetical requirements as well. Examples include representative buildings with all-glass architecture and historical buildings where design or conservatory reasons do not allow laying of cables.

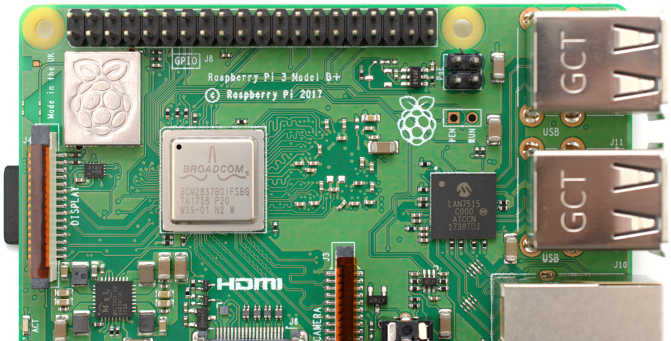
4) Integration of mobile devices: With wireless networks, associating mobile devices such as PDAs and Smartphones with the automation system becomes possible everywhere and at any time, as a device's exact physical location is no longer crucial for a connection (as long as the device is in reach of the network).

For all these reasons, wireless technology is not only an attractive choice in renovation and refurbishment, but also for new installations.

Components Required for design:

RASPBERRY PI BOARD (MODEL B):

The Raspberry Pi 3 is the third generation RaspberryPi. It replaced the Raspberry Pi 2 Model B inFebruary 2016. Compared to the Raspberry Pi 2 it has:
• A 1.2GHz 64-bit quad-core ARMv8 CPU
• 802.11n Wireless LAN • Bluetooth 4.1
• Bluetooth Low Energy (BLE)
• Like the Pi 2, it also has:
• 1GB RAM
• 4 USB ports
• 40 GPIO pins
• Full HDMI port
• Ethernet port
• Combined 3.5mm audio jack and composite video
• Camera interface (CSI)
• Display interface (DSI)
• Micro SD card slot (now push-pull rather than
push-push)
• VideoCore IV 3D graphics core The Raspberry Pi
3 has an identical form factor to the previous Pi 2 (and Pi 1 Model B+) and has complete compatibility with Raspberry Pi 1 and 2. We recommend the Raspberry Pi 3 Model B for use in schools, or for any general use. Those wishing to embed their Pi in a project may prefer the Pi Zero or Model A+, which are more useful for embedded projects, and projects which require very low power.
Voltages: Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V), which are un configurable. The remaining pins are all general purpose 3V3 pins, meaning outputs are set to 3V3 and inputs are 3V3-tolerant.



Outputs:
A GPIO pin designated as an output pin can be set to high (3V3) or low (0V).
Inputs:
A GPIO pin designated as an input pin can be read as high (3V3) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors. Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be configured in software.
More:
As well as simple input and output devices, the GPIO pins can be used with a variety of alternative functions, some are available on all pins, others on specific pins.
>PWM (pulse-width modulation)
>Software PWM available on all pins
>Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19
SPI:
>SPI0: MOSI (GPIO10); MISO (GPIO9);
SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
>SPI1: MOSI (GPIO20); MISO (GPIO19);
SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17);
CE2 (GPIO16)
I2C:
>Data: (GPIO2); Clock (GPIO3)
>EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
Serial:
>TX (GPIO14); RX (GPIO15)

DHT 11 (DIGITAL HUMIDITY AND TEMPERATURE SENSOR)

A humidity sensor senses, measures and regularly reports the relative humidity in the air. It measures both moisture and air temperature. Relative humidity, expressed as a percent, is theratioofactualmoistureintheairtothehighestamountofmoistureairatthattemperaturecan hold. The warmer the air is, the more moisture it can hold, so relative humidity changes with fluctuations in temperature.

Humidity Sensors Detect The Relative Humidity Of The Immediate environments in which they are placed.They Measure Both The Moisture And Temperature

In the air and express relative humidity as a percentage of the ratio of moisture in air themaximumamountthatcanbe held in the air at the current temperature. As air becomes hotter, it holds more moisture, so the relative humidity changes with the temperature.

Ultrasonic sensor:

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

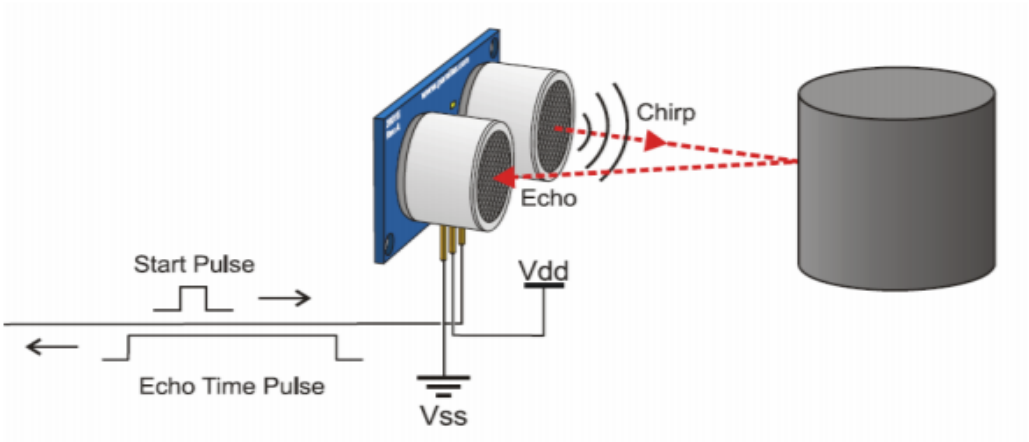
In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is D = ½ T x C (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be:

Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology. In comparison to infrared (IR) **sensors** in proximity sensing applications, ultrasonic sensors are not as susceptible to interference of smoke, gas, and other airborne particles (though the physical components are still affected by variables such as heat).

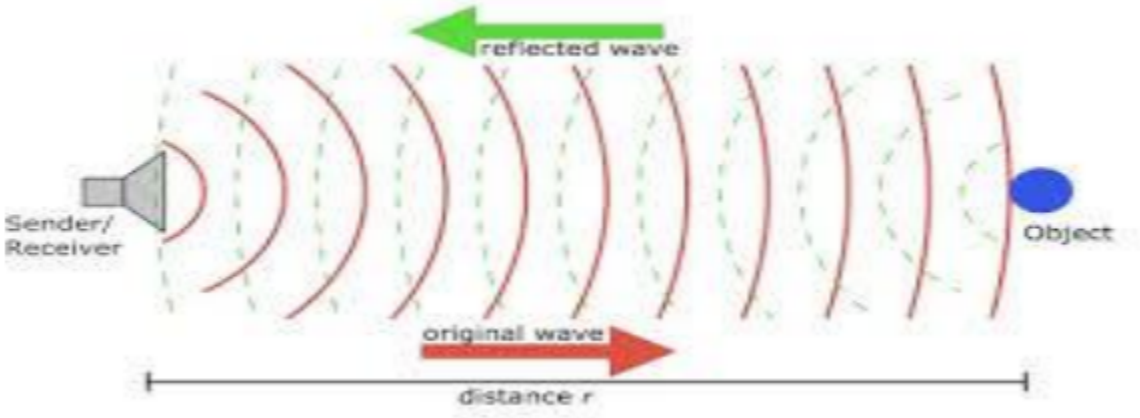
Ultrasonic sensors are also used as level sensors to detect, monitor, and regulate

liquid levels in closed containers (such as vats in chemical factories). Most notably, ultrasonic technology has enabled the medical industry to produce images of internal organs, identify tumors, and ensure the health of babies in the womb.

Working Of Ultrasonic sensor:



Ultrasonic sensors generate high-frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval



between sending the signal and receiving the echo to determine the distance to an object.

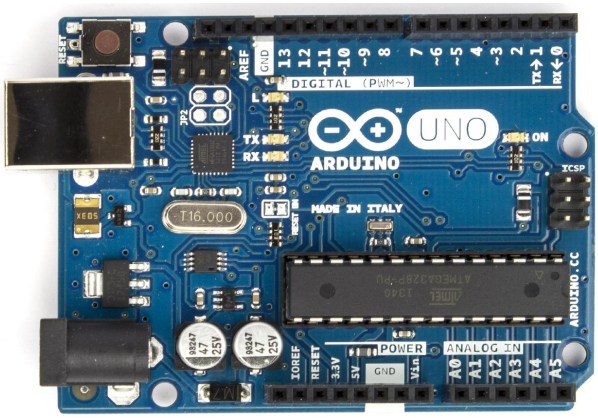
- Not affected by object colour and transparency as it detects distance through sound waves.
- Works well in places that are dim.
- Tend to consume lower current/power.
- Multiple interface options for pairing with a microcontroller, etc.

Arduino Uno Board:

Arduino is a software company, project, and user community that designs and manufactures computer open-source hardware,open-source software, and microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices.

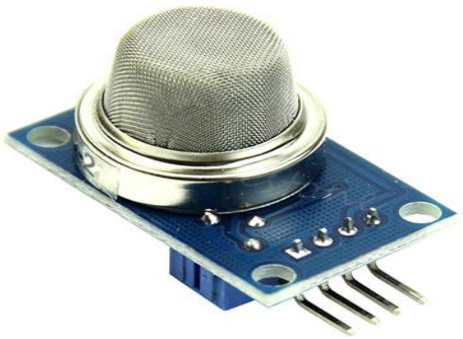
The project is based on microcontroller board designs, produced by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can interface to various expansion boards (termed shields) and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment(IDE) based on a programming language named Processing, which also supports the languages C and C++.

The first Arduino was introduced in 2005, aiming to provide a low cost, easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.



Gas sensor:

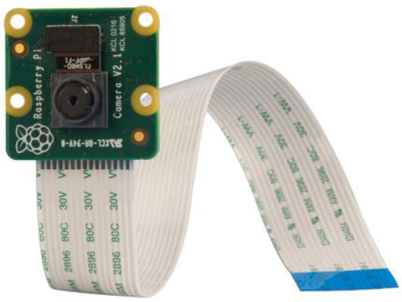
A gas sensor is a device which detects the presence or concentration of gases in the atmosphere. Based on the concentration of the gas the sensor produces a corresponding potential difference by changing the resistance of the material inside the sensor, which can be measured as output voltage. Based on this voltage value the type and concentration of the gas can be estimated.



The type of gas the sensor could detect depends on the sensing material present inside the sensor. Normally these sensors are available as modules with comparators as shown above. These comparators can be set for a particular threshold value of gas concentration. When the concentration of the gas exceeds this threshold the digital pin goes high. The analog pin can be used to measure the concentration of the gas.

Raspberry Pi Camera:

The v2 Camera Module has a Sony IMX219 8-megapixel sensor (compared to the 5-megapixel OmniVision OV5647 sensor of the original camera).

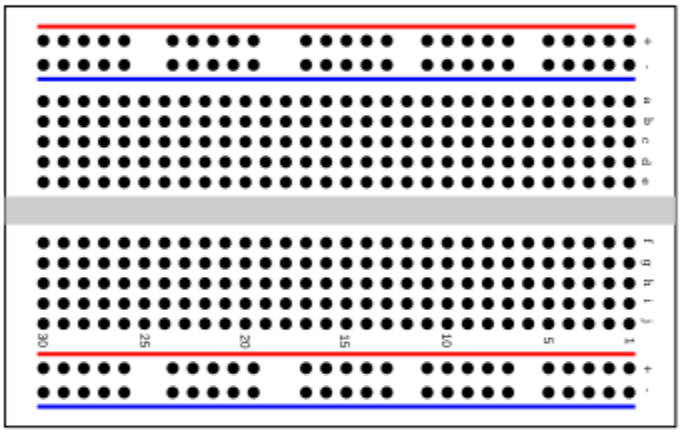


The Camera Module can be used to take high-definition video, as well as stills photographs. It's easy to use for beginners, but has plenty to offer advanced users if you're looking to expand your knowledge. There are lots of examples online of people

using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries we bundle with the camera to create effects.

Bread Board:

A breadboard is a construction base for prototyping of electronics. Originally it was literally a breadboard, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard (AKA plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these. "Breadboard" is also a synonym for "prototype". Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also extremely popular with students and in technological education. Older breadboard types did not have this property. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).



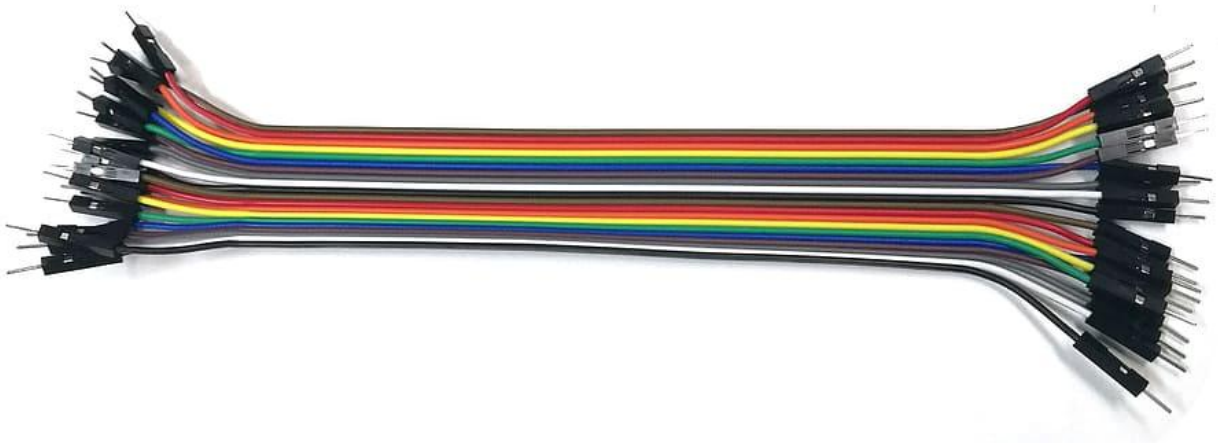
A modern solderless breadboard consists of a perforated block of plastic with numerous tin plated phosphor bronze or nickel silver alloy spring clips under the perforations. The clips are often called tie points or contact points. The number of tie points is often given in the specification of the breadboard.

The spacing between the clips (lead pitch) is typically 0.1 in (2.54 mm).

Integrated circuits (ICs) in dual in-line packages (DIPs) can be inserted to straddle the centerline of the block. Interconnecting wires and the leads of discrete components (such as capacitors, resistors, and inductors) can be inserted into the remaining free holes to complete the circuit. Where ICs are not used, discrete components and connecting wires may use any of the holes.

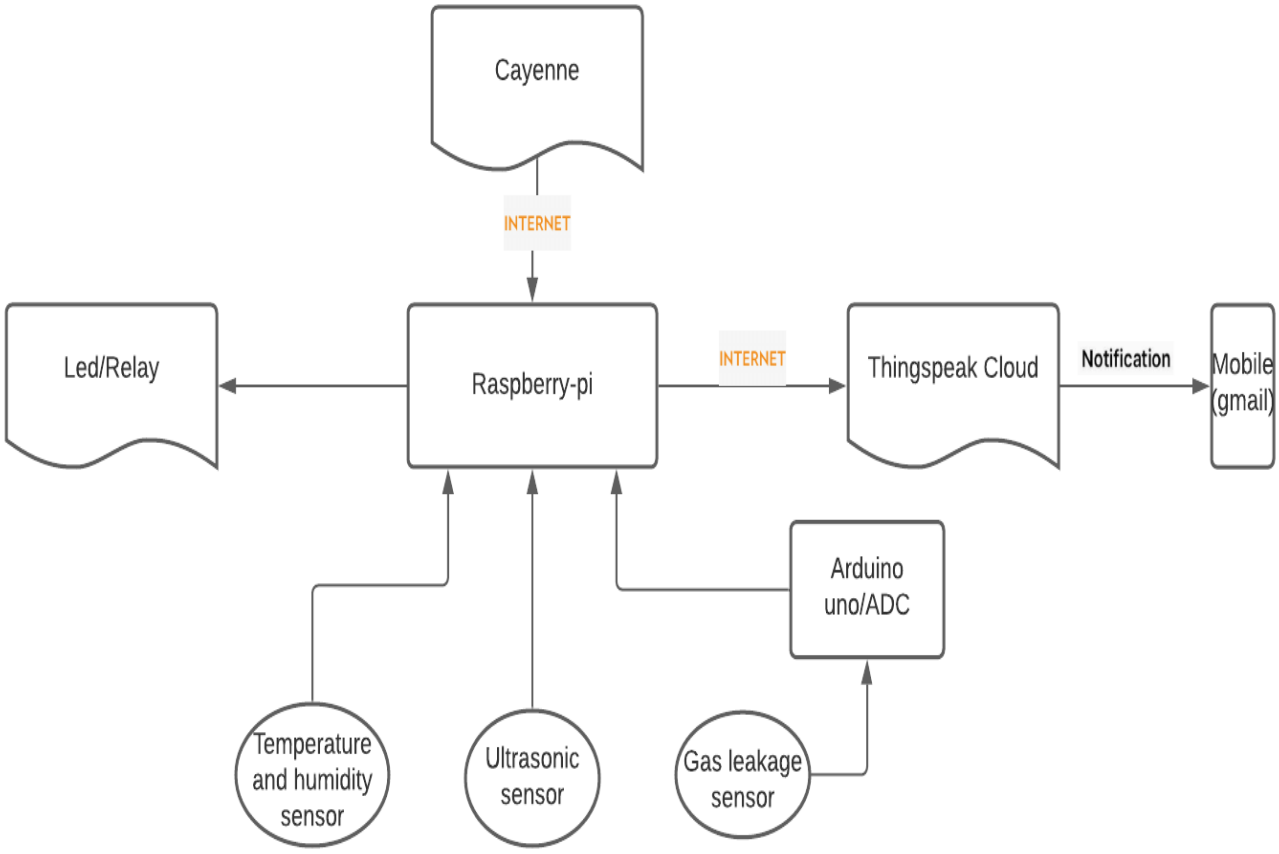
Jumper Wires:

Jump wires (also called jumper wires) for solderless breadboarding can be obtained in ready-to-use jump wire sets or can be manually manufactured. The latter can become tedious work for larger circuits. Ready To-use jump wires come in different qualities, some even with tiny plugs attached to the wire ends. Jump wire material for ready-made or homemade wires should usually be 22 AWG (0.33 mm2) solid copper, tin-plated wire - assuming no tiny plugs are to be attached to the wire ends. The wire ends should be stripped 3⁄16 to 5⁄16 in (4.8 to 7.9 mm).

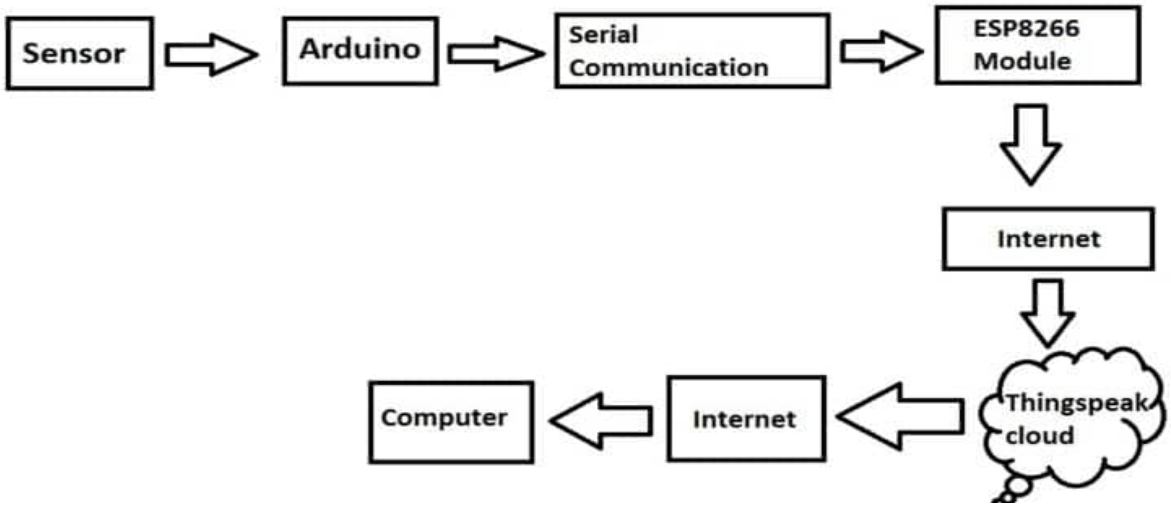


Differently colored wires and color-coding discipline are often adhered to for consistency. However, the number of available colors is typically far fewer than the number of signal types or paths. Typically, a few wire colors are reserved for the supply voltages and ground (e.g., red, blue, black), some are reserved for main signals, and the rest are simply used where convenient.

Block Diagram:



Block Diagram of Smart Dustbin:



Firstly the Arduino will trigger the ultrasonic sensor and collects the data, after calculating the distance arduino will convey this data to ESP8266 module via UART / serial communication. The serial data consists of distance data in cm with 2 to 3 decimal places. The ESP8266 will send this data to your Thingspeak account via the internet.

Since we are using a generic ESP8266 module we have to upload an appropriate program code that makes ESP8266 to accept serial data and send it to Thingspeak. We also need to upload another program code to the Arduino board, so that it will convert the measured ultrasonic sensor data into serial data.

Ultrasonic Sensor data ---> Arduino Uno ---> ESP8266 Wireless Transceiver module ---> Thingspeak cloud.

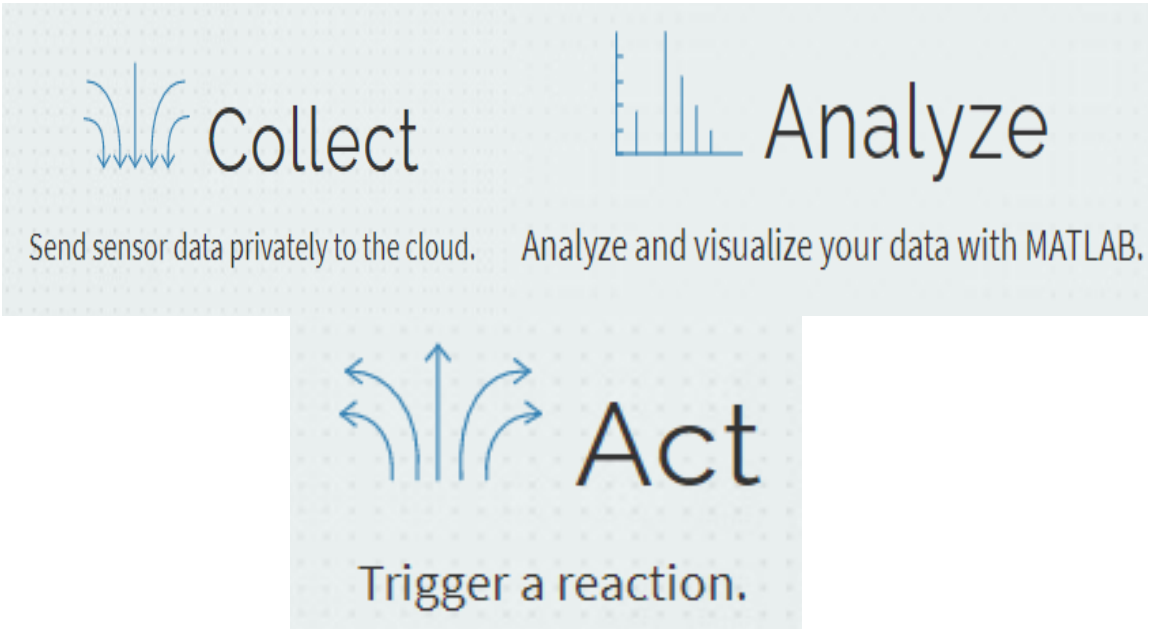
ThingSpeak

Features of Home automation system:

1. Temperature monitoring.
2. Humidity.
3. Gas leakage detection.
4. Water level indication.
5. Security Camera.
6. Smart home dustbin.

About Thingspeak cloud:

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.



ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of ThingSpeak include the ability to:

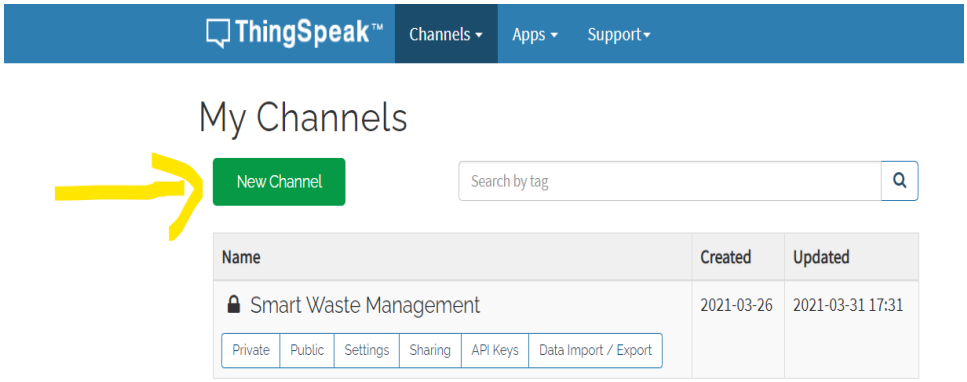
- Easily configure devices to send data to ThingSpeak using popular IoT protocols.
- Visualize your sensor data in real-time.
- Aggregate data on-demand from third-party sources.
- Use the power of MATLAB to make sense of your IoT data.
- Run your IoT analytics automatically based on schedules or events.
- Prototype and build IoT systems without setting up servers or developing web software.
- Automatically act on your data and communicate using third-party services like Twilio® or Twitter®.
-

❖Creating Channel in Thingspeak cloud:

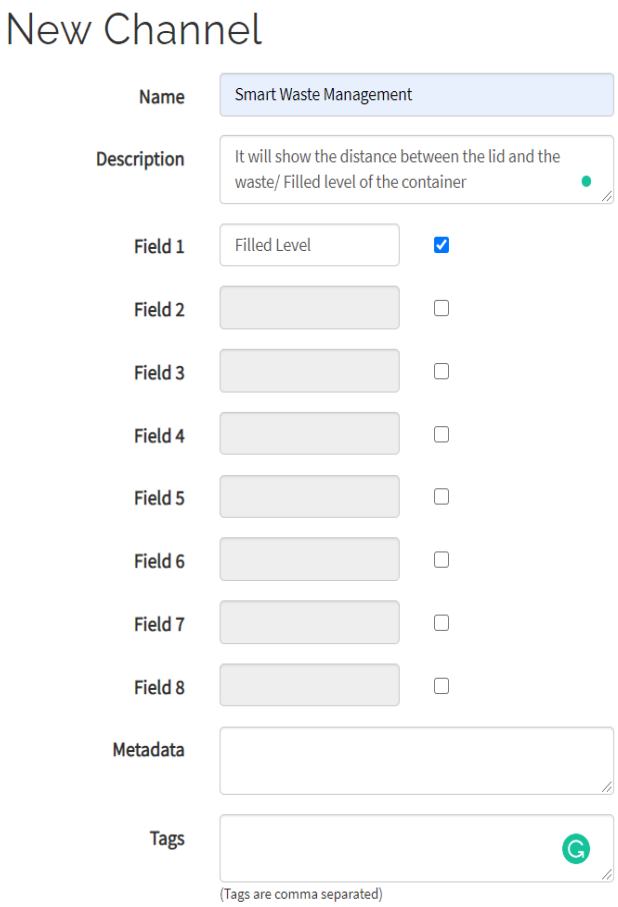
Step 1: Open the browser and visit thingspeak cloud using the following link.https://thingspeak.com/login?skipSSOCheck=true

Step 2: Login to your account.

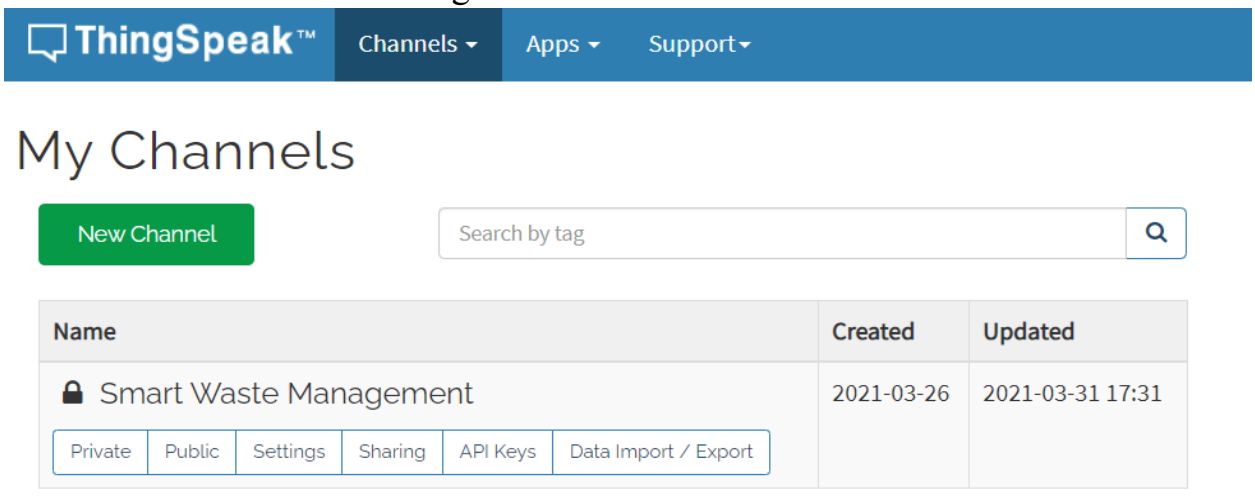
Step 3: Click on the new channel.



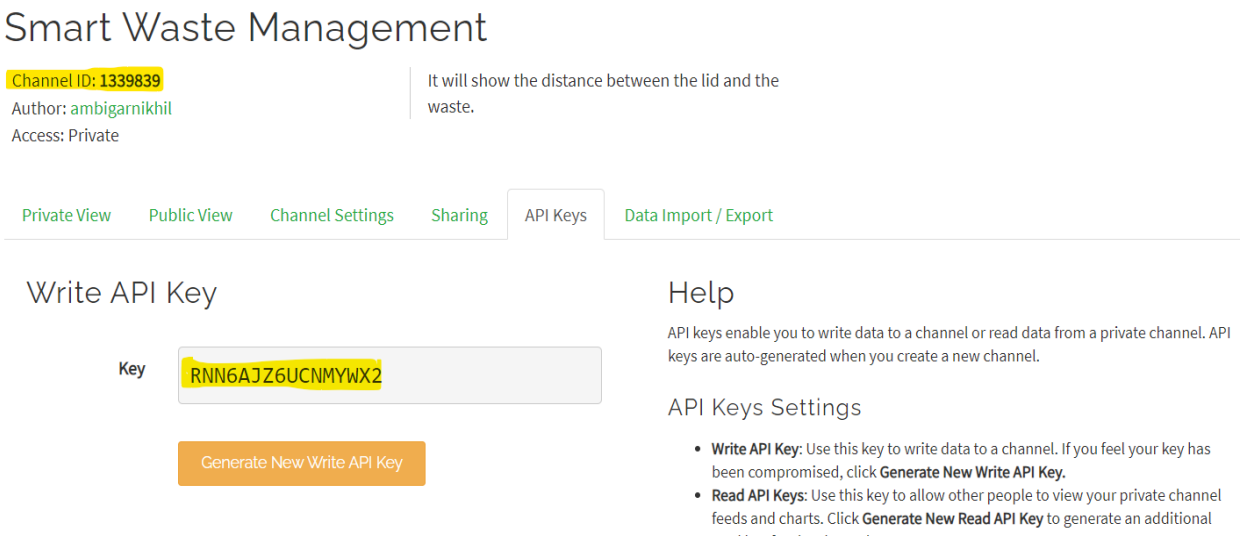
Step 4: Fill all the required details.



Step 4: After filling all the details click on the save channel, the channel will be created and looks something like this.

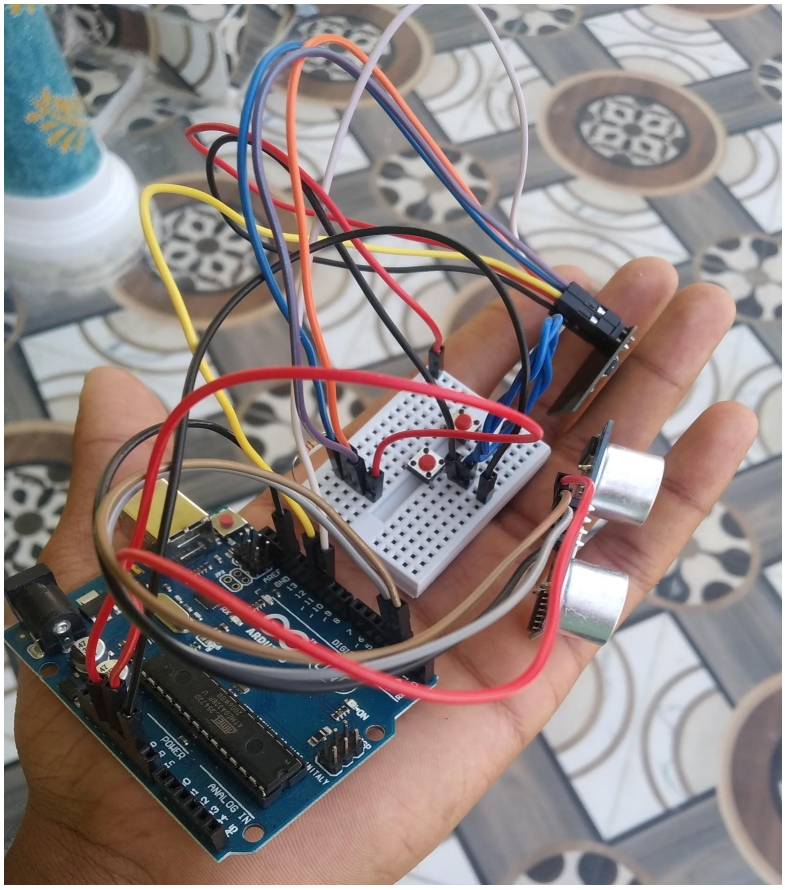


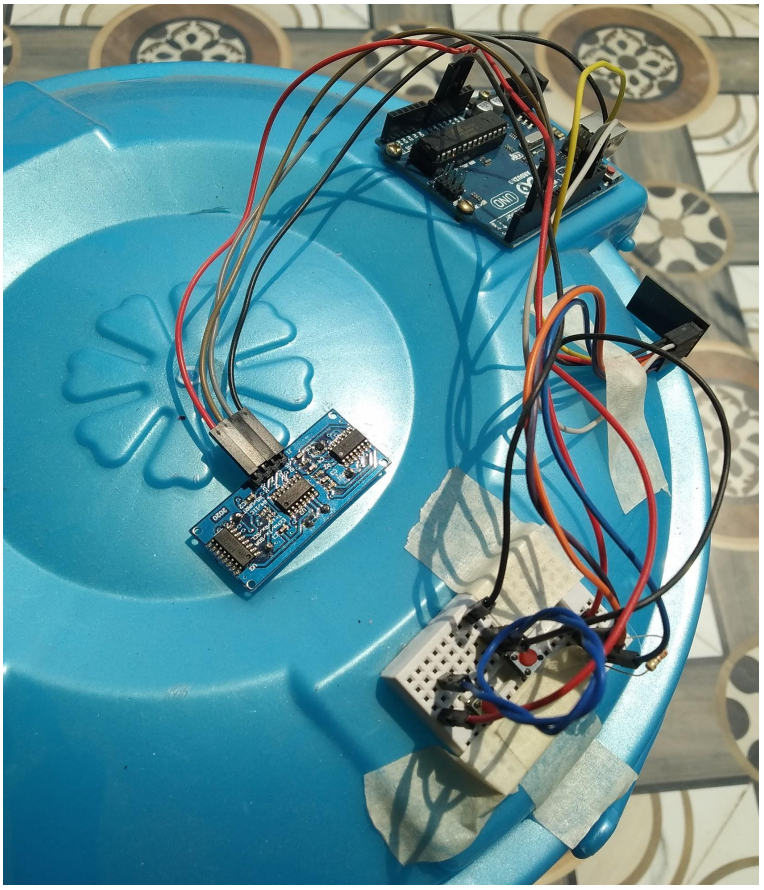
Step 5:After creating we need to copy channel id, and write api key as we need it for programming esp8266 module.



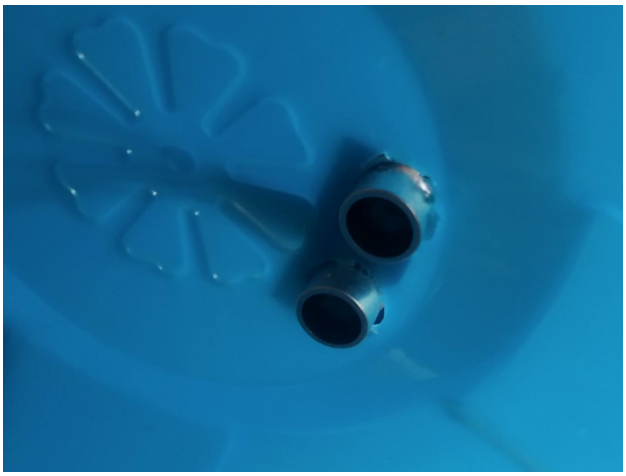
We made a video on the entire process of creating a thingspeak channel , it can be accessed via the following link. https://youtu.be/ehq28VdjM8Y

❖ For demo purposes we fixed the module onto the small dustbin and checked its performance.





Interior view of the dustbin:



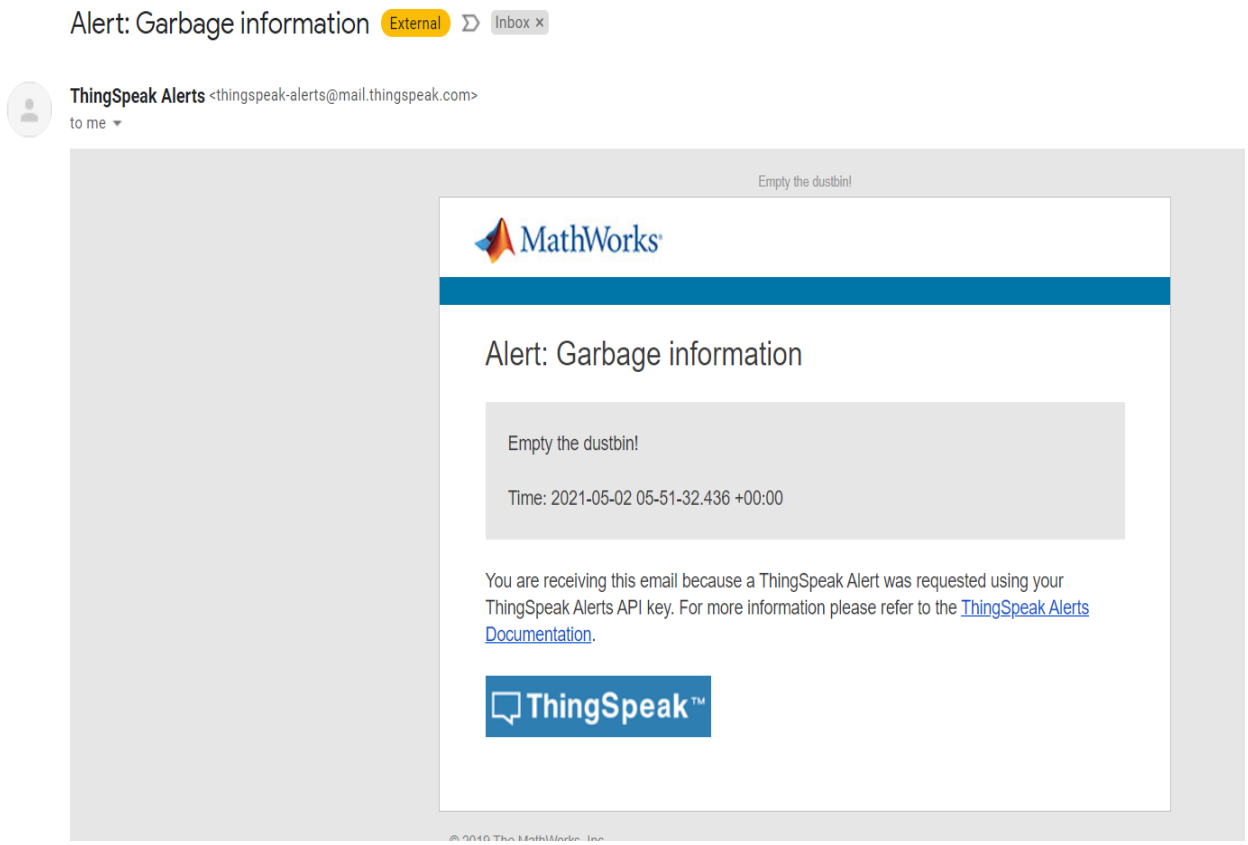
Demonstration video link:https://drive.google.com/file/d/1ZDDyE3r1UvYeT_7pusQywcre3h5chCBE/view?usp=sharing

❖ Alert email on reaching the threshold value:

For demo purposes we used a small dustbin of size ~27cm.

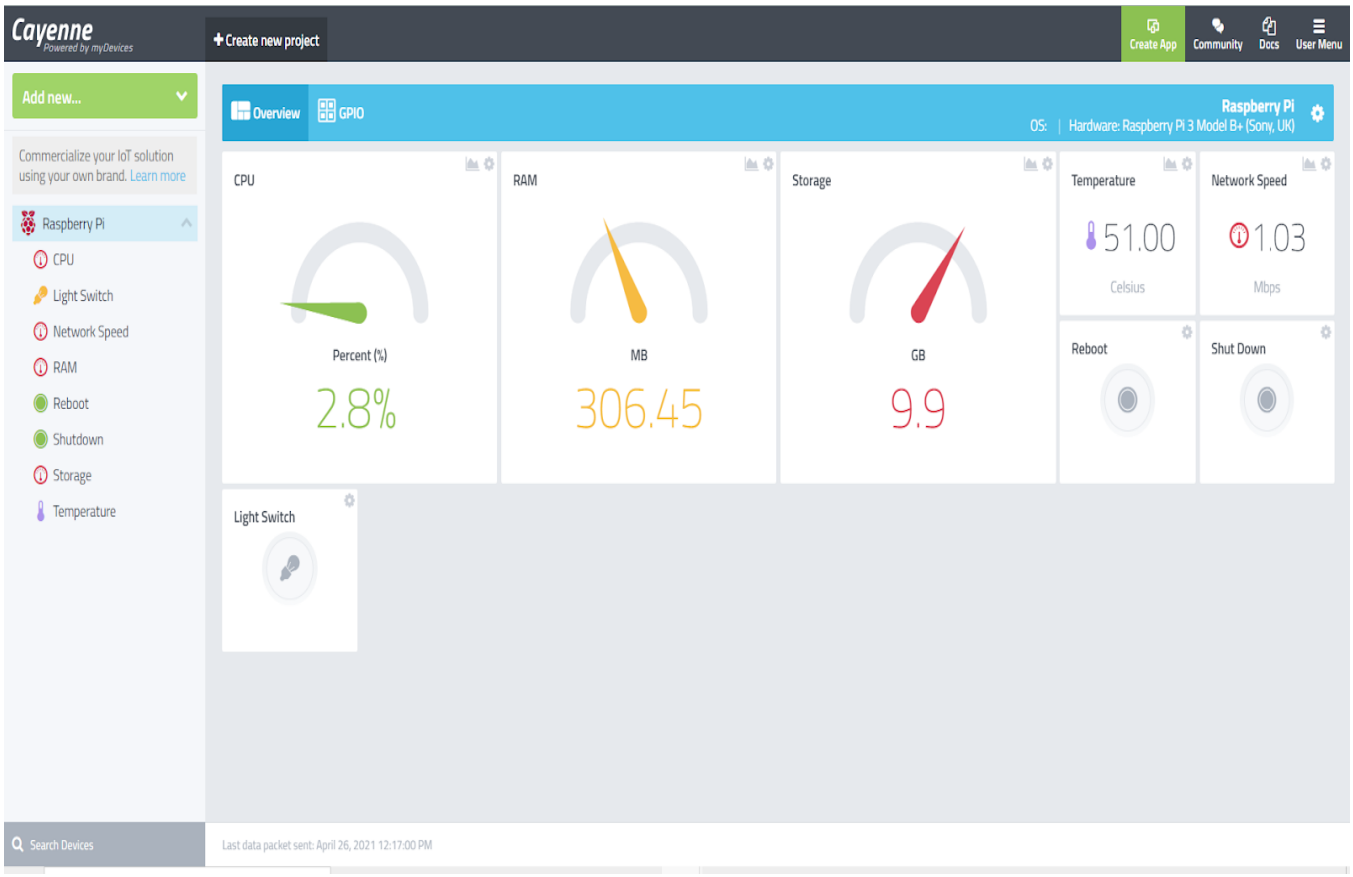
And Selected threshold value of 15 cm.

So, whenever the thingspeak channel receives the data, compares it with the threshold value. If the latest value received is less than the threshold value then it sends the trigger mail as shown below.



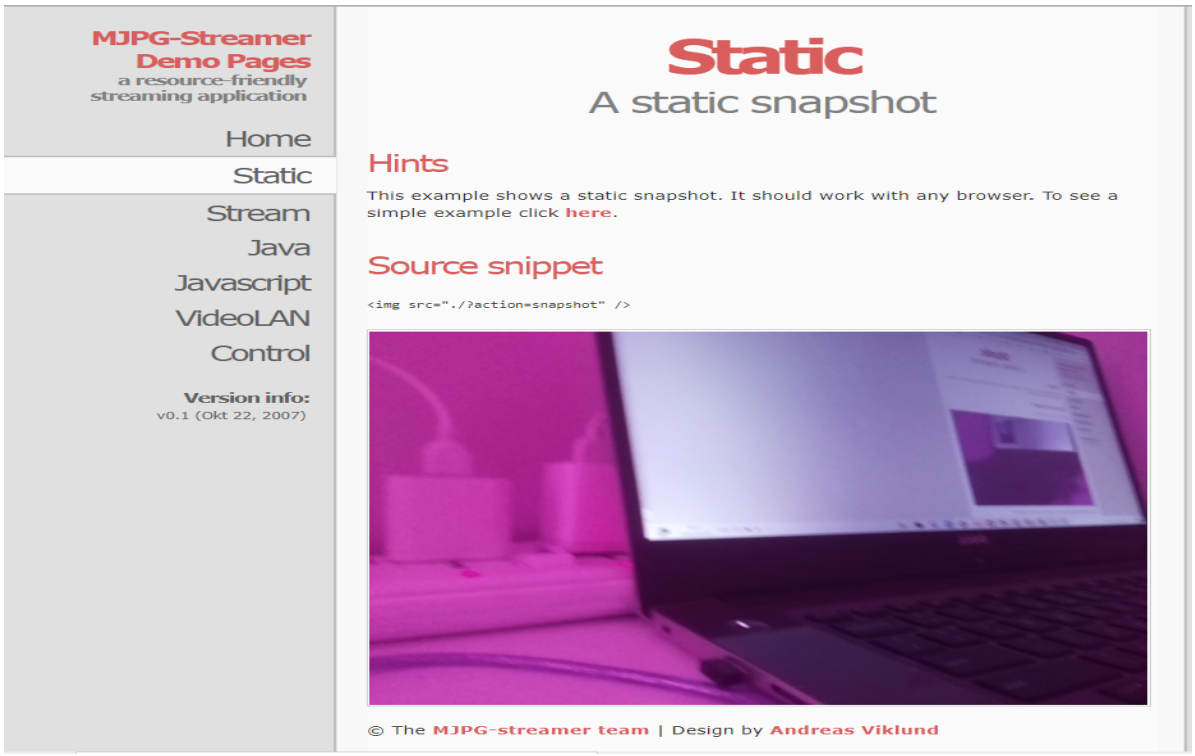
Cayenne :

_____Cayenne IoT cloud platform which helps to control the raspberry pi from anywhere of the world.It uses MQTT protocol to send messages from cloud to raspberry pi.



MPEG Streamer:

It is online steamer of content of raspberry pi camera.We need install **MJPG-streamer application** on raspberry pi using git file `https://github.com/jacksonliam/mjpg-streamer.git`



Codes:

Code for Measuring temperature, water level, humidity, gas leakage detection:

```
import Adafruit_DHT
import sys
import urllib
import urllib2
from time import sleep
import serial
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)

#set GPIO Pins
GPIO_TRIGGER = 24
GPIO_ECHO = 23

#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)

def distance():
    GPIO.output(GPIO_TRIGGER, True)
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)
    StartTime = time.time()
    StopTime = time.time()
     while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()
     TimeElapsed = StopTime - StartTime
    # multiply with the sonic speed (34300 cm/s)
    # and divide by 2, because there and back
```

```
    distance = (TimeElapsed * 34300) / 2

    return distance

# Enter Your API key here
myAPI = '20EO6EXRT1HXYTYV'
# URL where we will send the data, Don't change it
baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI
# Sensor should be set to Adafruit_DHT.DHT11,
# Adafruit_DHT.DHT22, or Adafruit_DHT.AM2302.
sensor = Adafruit_DHT.DHT11
pin = 15

ser = serial.Serial('/dev/ttyUSB0',9600)
s = [0,1]
i=0
while True:
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
    dist = distance()
    print ("Measured Distance = %.1f cm" % dist)
    dist=str(dist)
    r=ser.readline()
    a=str(r)
    if(i>0):
        a=a.split(" ")
        aa=a[2].split("\\")
        aa=aa[0].split(":")
        aa=aa[1].split("ppm")
        aa[0]=float(aa[0])
    i=i+1

    if humidity is not None and temperature is not None:
        humi = '%0.1f' % humidity
        temp = '%0.1f' % temperature
            # Sending the data to thingspeak
```

```
        try:
            conn = urllib2.urlopen(baseURL + '&field1=%s&field2=%s&field3=%s&field4=%s' %
(temp, humi,aa[0],dist))


            print('Temp={0:0.1f}*C  Humidity={1:0.1f}%  LPG content:{2:0.5f}
ppm'.format(temperature, humidity,aa[0]))
        except:
            pass


    else:
        print('Failed to get reading. Try again!')
```

Matlab code to trigger mail:-

```
% Store the channel ID for the moisture sensor channel.
channelID = 1339592;

% Provide the ThingSpeak alerts API key.  All alerts  API keys start with TAK.
alertApiKey = 'TAKQC0YNKJ4CGQ2WVA40R';

% Set the address for the HTTTP call
alertUrl="https://api.thingspeak.com/alerts/send";

% webwrite uses weboptions to add required headers.  Alerts needs a
ThingSpeak-Alerts-API-Key header.
options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", alertApiKey ]);

% Set the email subject.
alertSubject = sprintf("Gas Leakage Alaram");

% Read the recent data.
moistureData = thingSpeakRead(channelID,'NumDays',1,'Fields',3);

% Check to make sure the data was read correctly from the channel.
if isempty(moistureData)
```

```
    alertBody = ' No data read from sensor. ';
  else

    lastValue = moistureData(end);

    % Set the outgoing message
    if (lastValue>=1)
      try
        webwrite(alertUrl , "body", lastValue, "subject", alertSubject, options);
      catch someException
        fprintf("Failed to send alert: %s\n", someException.message);
      end
    end
  End
```

Code for smart dustbin:

1. **Uploading code to ESP8266 wireless transceiver module.**

   Code:

```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>

//-------- Enter Your Wi-Fi Details -----//
char ssid[] = "Nikhil";  //SSID
char pass[] = "6364077543"; //Password
//--------------------------------------//

WiFiClient  client;

unsigned long myChannelField = 1339839; // Channel ID
const int ChannelField = 1; // Which To Field Write
const char * myWriteAPIKey = "RNN6AJZ6UCNMYWX2"; // Write API Key

String value = "";
void setup()
{
 Serial.begin(9600);
 WiFi.mode(WIFI_STA);
 ThingSpeak.begin(client);
```

```
}
void loop()
{
 if (Serial.available() > 0)
 {
  while (Serial.available() > 0)
  {
   int inChar = Serial.read();
   value += (char)inChar;
  }
 }
 if (WiFi.status() != WL_CONNECTED)
 {
  while (WiFi.status() != WL_CONNECTED)
  {
   WiFi.begin(ssid, pass);
   delay(5000);
  }
 }
 ThingSpeak.writeField(myChannelField, ChannelField, value, myWriteAPIKey);
 value = "";
}
```

2. Uploading code to Arduino Uno:

Code:

```
#include <SoftwareSerial.h>
SoftwareSerial ESP(10,11);
const int trigger = 2;
const int echo = 3;
long T;
float distanceCM;
void setup()
{
 pinMode(trigger, OUTPUT);
 pinMode(echo, INPUT);
 Serial.begin(9600);
 ESP.begin(9600);
}
void loop()
{
 digitalWrite(trigger, LOW);
 delay(1);
 digitalWrite(trigger, HIGH);
 delayMicroseconds(10);
```

```
  digitalWrite(trigger, LOW);
  T = pulseIn(echo, HIGH);
  distanceCM = T * 0.034;
  distanceCM = distanceCM / 2;
  Serial.print("Distance in cm: ");
  Serial.println(distanceCM);
  ESP.print(distanceCM);
  delay(10000);
}
```

- **Matlab analysis to read the channel data and send a trigger mail after reaching the threshold value:**

Code:

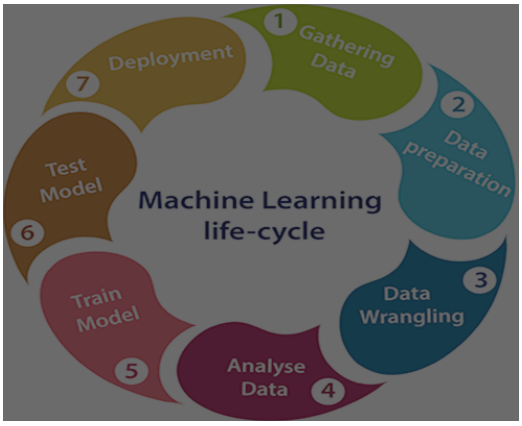
```
channelID = 1339839;

% Provide the ThingSpeak alerts API key.  All alerts  API keys start with TAK.
alertApiKey = 'TAKCFCGEX6WK129Z0B878';

% Set the address for the HTTP call
alertUrl="https://api.thingspeak.com/alerts/send";

% webwrite uses web options to add required headers.  Alerts needs a
ThingSpeak-Alerts-API-Key header.
options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", alertApiKey ]);

% Set the email subject.
alertSubject = sprintf("Garbage information");

% Read the recent data.
moistureData = thingSpeakRead(channelID,'NumDays',30,'Fields',1);

% Check to make sure the data was read correctly from the channel.
if isempty(moistureData)
  alertBody = ' No data';
else
  latest Value = moistureData(end);
  % Set the outgoing message
  if (latestValue<=15)

    alertBody = 'Empty the dustbin!';
    try
      webwrite(alertUrl , "body", alertBody, "subject", alertSubject, options);
    catch someException
      fprintf("Failed to send alert: %s\n", someException.message);
```

```
    end
  end
End
```

Future Prediction Using Machine learning.

Steps for building machine learning model:

1. Data Collection.
2. Data Pre-processing.
3. Choose a model.
4. Train the model.
5. Evaluate the model.
6. Make predictions.



Link for the dataset used for training the model:https://github.com/ambigarnikhil/Smart-waste-management-system-based-on-IoT-Platform/blob/main/Dataset%20-%20Sheet3.csv

We are using **Decision Tree Classifiers** for training our model.

- We are using Decision-Tree-Classifier for training the model. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.
- The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
- A tree can be seen as a piecewise constant approximation. DecisionTreeClassifier takes as input two arrays: an array X, sparse or dense, of shape (n_samples, n_features) holding the training samples, and an array Y of integer values, shape (n_samples,), holding the class labels for the training samples.
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm**,** which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further splits the tree into subtrees.

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follows while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase.

Link of the Jupyter notebook/Python code for future prediction:https://github.com/ambigarnikhil/Smart-waste-management-system-based-on-IoT-Platform/blob/main/Final.ipynb

Result :

```
import sklearn.metrics as metrics
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
accuracy = metrics.accuracy_score(y_pred,y_test)
print("Accuracy : %s" % "{0:.3%}".format(accuracy))            #Accuracy of the model
print("r2_score:",r2_score(y_test, y_pred))                      #R2 value
print("mse:",mean_squared_error(y_test, y_pred)/49)                 #MSE(Mean Square Error) of the model
```

```
Accuracy : 77.551%
r2_score: 0.7691828120397397
mse: 2.344439816743024
```

Trained model is almost 80% accurate.

MSE is calculated by the sum of squares of prediction error which is real output minus predicted output and then divided by the number of data points. It gives you an absolute number on how much your predicted results deviate from the actual number.

The **r2 score** varies between 0 and 100%. It is closely related to the **MSE** (see below), but not the same. **R2** is

" ...the proportion of the variance in the dependent variable that is predictable from the independent variable(s)."

Another definition is "(total variance explained by model) / total variance." So if it is 100%, the two variables are perfectly correlated, i.e., with no variance at all. A low value would show a low level of correlation, meaning a regression model that is not valid, but not in all cases.

Machine learning model accuracy is the measurement used to determine which

model is best at identifying relationships and patterns between variables in a dataset based on the input, or training, data. The better a model can generalize to 'unseen' data, the better predictions and insights it can produce, which in turn deliver more business value.

Our Trained model is almost 80% accurate.

Related work:

[1] Sirsath N. S, Dhole P. S, Mohire N. P, Naik S. C & Ratnaparkhi IS
This paper proposes a Home Automation system that employs the integration of multi-touch mobile devices, cloudnet working, wireless communication, and power-line communication to provide the user with remote control of various lights and appliances within their home. This system uses a consolidation of a mobile phone application, handheld wireless remote,and PCbase program to provide means of user interface to the consumer.

[2] Basil Hamed
ThemainobjectiveofthisPaper is todesignand implement a control and monitor system for smart houses. Smart house system consists of many systems that are controlled by LabVIEW software as the main controlling system in this paper. Also, the smart house system was supported by remote control system as a sub controlling system. The system also is connected to the internet to monitor and control the house equipment's from anywhere in the world using LabVIEW.

[3] Deepali Javale, Mohd. Mohsin, ShreerangNandanwar
The prime objective of this paper is to assist handicapped/old aged people. It gives basic idea of how to control various home appliances and provide a security using Android phone/tab. The design consists of Android phone with home automation application, Arduino Mega ADK. User can interact with the android phone and send control signal to the Arduino ADK which in turn will control other embedded devices/sensors.