SMART WASTE MANAGEMENT SYSTEM BASED ON A IoT PLATFORM

A PROJECT REPORT

Submitted by

SHIVA SAI(18BEC012)
KIRAN D(18BEC022)
NIKHIL S A(18BEC033)
NITHIN R(18BEC034)

Under the guidance of : Dr . Prakash Pawar

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

Ittigatti Road, Near Sattur Colony, Dharwad 580009,
DHARWAD
JANUARY 2021/MAY 2021

Table of Contents:

Introduction

1.1 Overview:

Waste management is all the activities and actions required to manage waste from its inception to its final disposal . This includes collection, transportation, treatment and disposal of waste together with monitoring and regulation. Waste collection methods vary widely among different countries and regions. Domestic waste collection services are often provided by local government authorities.

Curbside collection is the most common method of disposal in most countries, in which waste is collected at regular intervals by specialised trucks. Waste collected is then transported to an appropriate disposal area.

Nowadays, cities with developing economies experience exhausted waste collection services, inadequately managed and uncontrolled dumpsites and the problems are worsening. Waste collection method in such countries is an on-going challenge and many struggle due to weak institutions and rapid urbanization.

1.2 Need for improvement in smart waste management:

- By 2030, almost two-third of the world's population will be living in cities. This fact requires the development of sustainable solutions for urban life, managing waste is a key issue for health.
- Efficient and energy-saving waste management, reducing CO2, air pollution and vehicle exhaust emissions are just a few examples for the demands of future cities. In view of that, the efficient use and responsible handling of resources become more important.
- Effectively managing waste is important in developed countries. Waste management may swallow upto 50% of a city's budget, but only serve a small part of the population.
- Sometimes, upto 60%of waste is not being collected, it is often simply burned by the roadside. It can pollute drinking water, it can spread disease to people living nearby.
- Lack of the information about the collecting time and area.
- Lack of the proper system for monitoring, tracking the trucks and trash bins that have been collected in real time.
- One of the main concerns with our environment has been solid waste management which impacts the health and environment of our society.
- There is no quick response to urgent cases like truck accidents, breakdown,

longtime idling.

- There is no quick way to respond to client's complaints about uncollected waste.
- Even with great route optimization, the worker must still physically go to the dustbin to check waste levels. Because of this, trucks often visit containers that do not need emptying, which wastes both time and fuel.
- Waste management prevents harm to human health and the environment by reducing the volume and hazardous character of residential and industrial waste.
- Improving proper waste management will reduce pollution, recycle useful materials and create more green energy.

1.3 Features of smart waste management:

- The smart, sensor based dustbin will judge the level of waste in it and send the message directly to the municipal corporation.
- It can sense all the types of waste material either in the form of solid or liquid.
- According to the filled level of the dustbin, the vehicles from the municipal corporation will choose the shortest path with the help of the "TRANSPORTATION SOFTWARE", which will save their time.
- It emphasizes on "DIGITAL INDIA".
- The system is simple. If there is any problem with any equipment in the future, that part is easily replaceable with new one without any difficulty and delay

1.4 Advantages of smart waste management:

- Less time and fuel consumption as the trucks go only to the filled containers.
- Decreased noise, traffic flow and air pollution as a result of less trucks on the roads.
- Our smart operating system enables two way communication between the dustbin deployed in the city and service operator. Therefore the focus is only on collection of route based fill level of the containers.
- The sensors installed in the containers provide real time information on the fill level. This information helps determine when and where to prioritise collection.
- In this way both service providers and citizens benefit from an optimized system which results in major cost savings and less urban pollution.
- Reduces the infrastructure (trucks, containers), operating (fuel) and maintenance costs of the service by upto 30%.

- Applying this technology to the city optimises management, resources and costs, and makes it a “SMART CITY”.
- Historical information on collections helps adapt the deployment of containers to the actual needs of the city, therefore reducing the number of containers that clutter up the road and increasing public parking spaces.
- It keeps the surroundings clean and green, free from bad odour of wastes, emphasizes a healthy environment and keeps cities more beautiful.
- Reducing manpower required to handle the garbage collection.

1.5 Applications of smart waste management:

- This can be best used by municipal corporations for their betterment of management regarding collection of wastes.
- With the help of proper technology (GPS & SOFTWARE APPLICATIONS) we can guide the trucks to choose the shortest path.
- It also favours the “SMART CITY” project and “DIGITAL INDIA”.

Review of Literature:

An inevitable consequence of development and industrial progress is generation of waste. Therefore, efficient waste management is a matter of international concern and countries have set up robust regulatory waste management regimes for balancing the objectives of development and environment sustainability. In India, the national environment policy, 2006 while suggesting measures for collection of wastes and safe disposal of residues. The metro cities and major economic hubs generate the maximum volume of waste, but a survey of 20 smaller cities selected

to be developed as smart cities show that most are struggling to manage waste. So, there should be an improvement in the waste management techniques.

Following the onset of industrialisation and the sustained urban growth of large population centres, the build-up of waste in the cities caused a rapid deterioration in levels of sanitation and the general quality of urban life. The streets became choked with filth due to the lack of waste clearance regulations .
In the UK, London, The Metropolitan Board of Works was the first city-wide authority that centralized sanitation regulation for the rapidly expanding city and the Public Health Act 1875 made it compulsory for every household to deposit their weekly waste in "moveable receptacles: for disposal—the first concept for a dust-bin.
Early garbage removal trucks were simply open bodied dump trucks pulled by a team of horses. They became motorized in the early part of the 20th century and the first close body trucks to eliminate odours with a dumping lever mechanism were introduced in the 1920s in Britain .

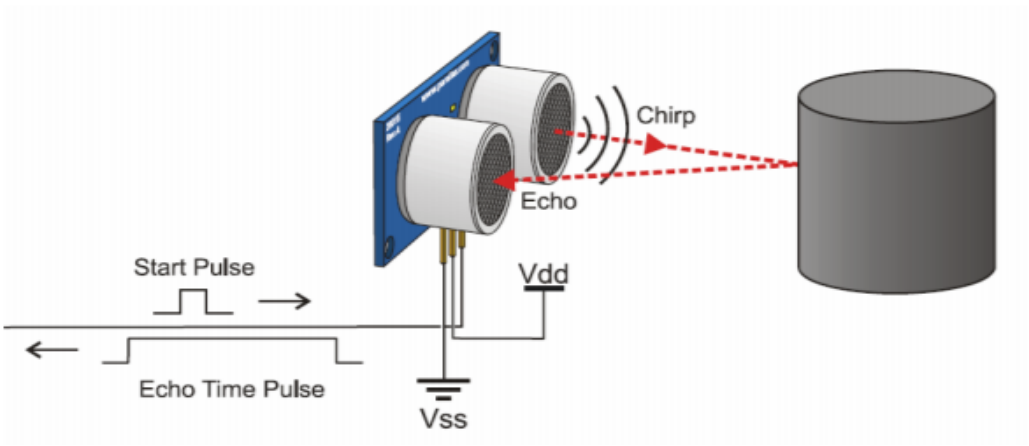
Components Required for designing:

1. Garbage Container:
   A waste container is a container for temporarily storing waste, and is usually made out of metal or plastic.
   The curbside dustbins usually consist of three types: trash cans (receptacles made of metal or plastic), dumpsters (large receptacles similar to skips) and wheelie bins (light, usually plastic bins that are mobile). All of these are emptied by collectors, who will load the contents into a garbage truck and drive it to a landfill, incinerator or consuming crush facility to be disposed of.

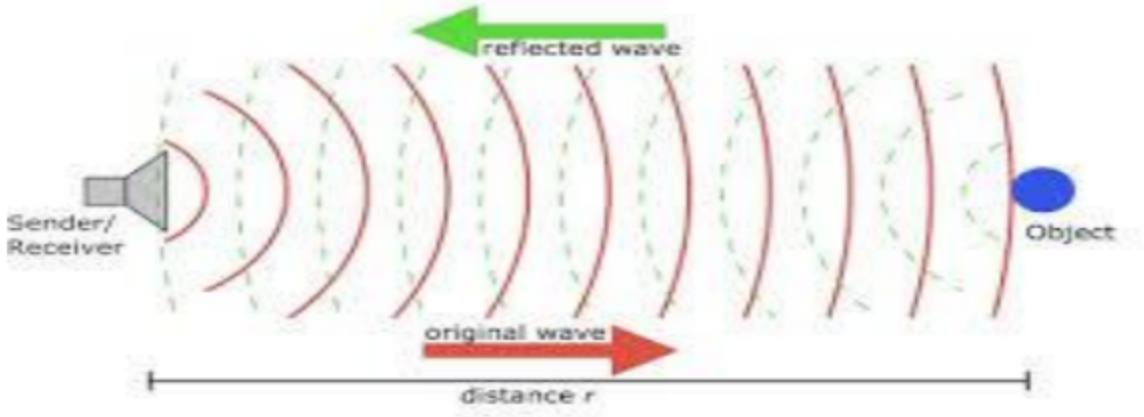
2. Ultrasonic Sensor:

A special sonic transducer is used for the ultrasonic proximity sensors, which allows for alternate transmission and reception of sound waves. The sonic waves emitted by the transducer are reflected by an object and received back in the transducer. After having emitted the sound waves, the ultrasonic sensor will switch to receive mode. The time elapsed between emitting and receiving is proportional to the distance of the object from the sensor.

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.



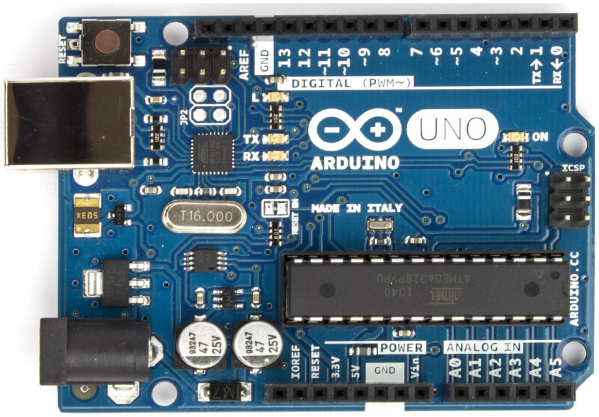
- Not affected by object colour and transparency as it detects distance through sound waves.
- Works well in places that are dim.
- Tend to consume lower current/power.
- Multiple interface options for pairing with a microcontroller, etc.

3. 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.

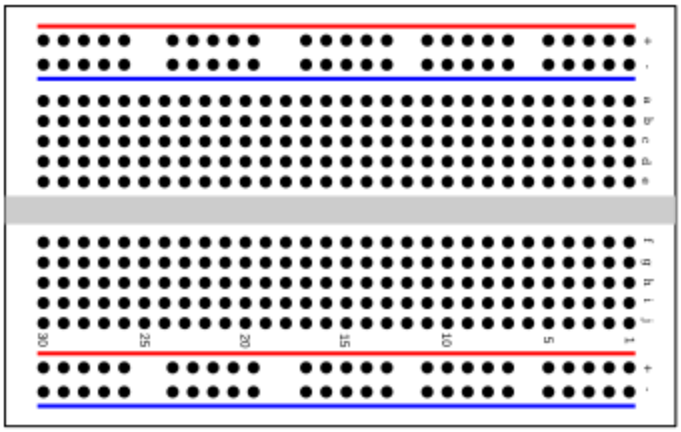


Arduino Software:

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides a simple one-click mechanism to compile and load programs to an Arduino board. A program written with the IDE for Arduino is called a "sketch". The Arduino IDE supports the languages C and C++ using special rules to organize code.

4. 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).

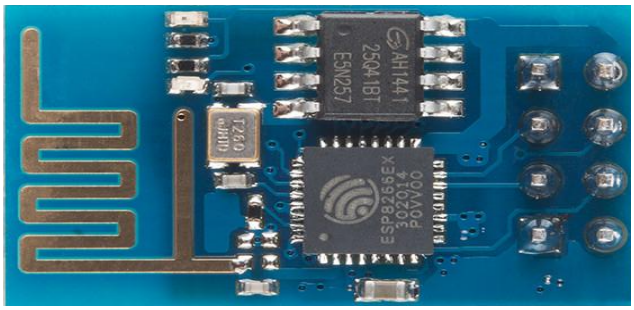


A modern solderless breadboard consists of a perforated block of plastic with numerous tin plated phosphor bronze or nickel silver alloy spring clips under the perforations. The clips are often called tie points or contact points. The number of tie points is often given in the specification of the breadboard.

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.

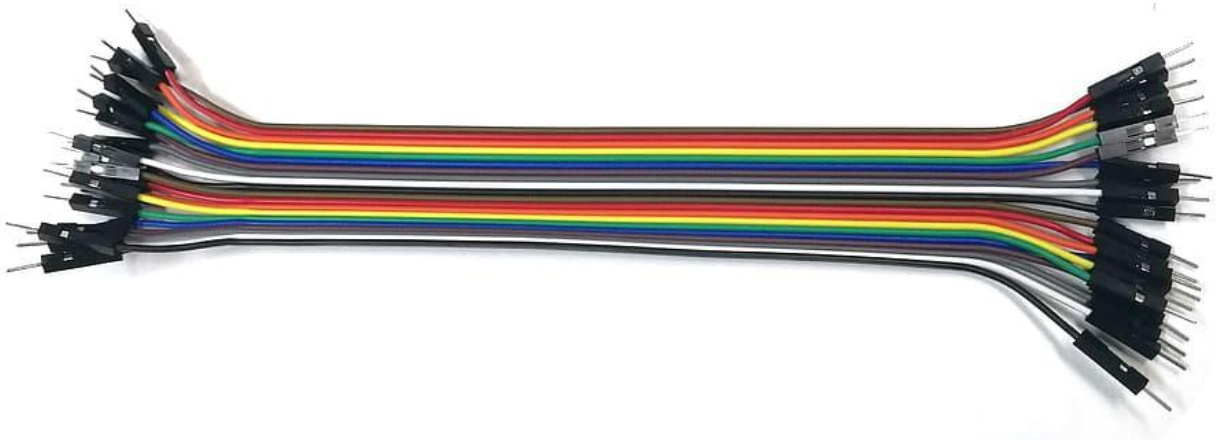
5. ESP8266 Wifi Module:

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers. The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.



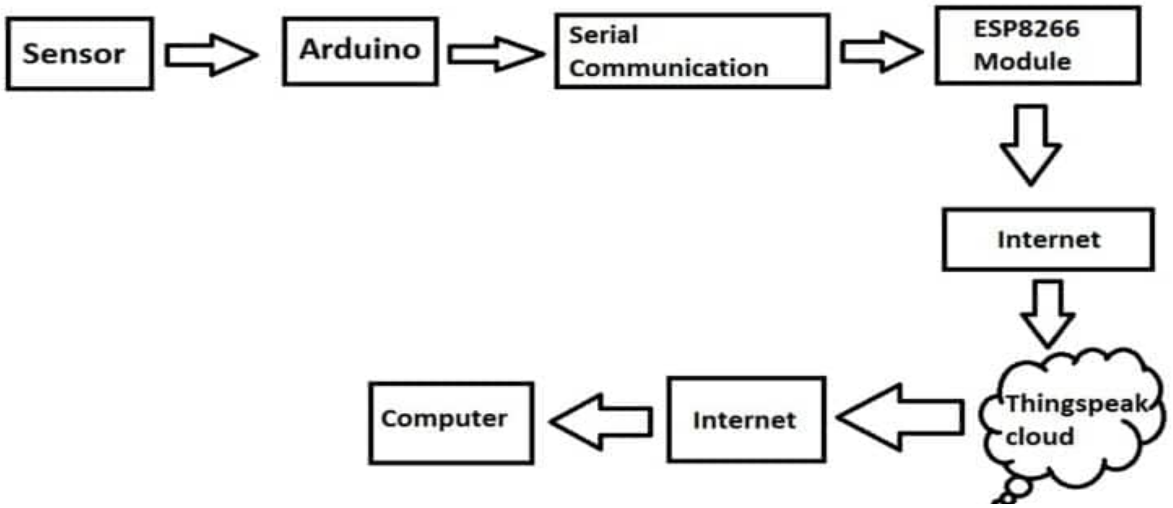
6. Jumper Wires:

   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). Shorter stripped wires might result in bad contact with the board's spring clips (insulation being caught in the springs). Longer stripped wires increase the likelihood of short-circuits on the board. Needle-nose pliers and tweezers are helpful when inserting or removing wires, particularly on crowded boards.



   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:



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 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

❖ 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:

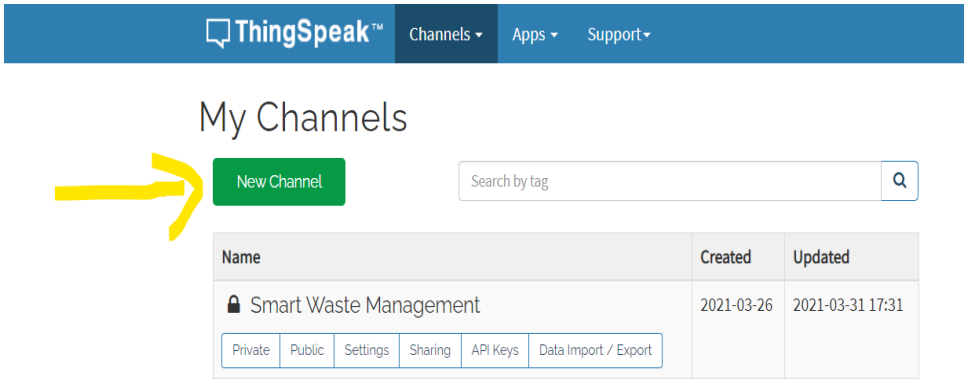
- Easily configure devices to send data to ThingSpeak using popular IoT protocols.
- Visualize your sensor data in real-time.
- Aggregate data on-demand from third-party sources.
- Use the power of MATLAB to make sense of your IoT data.
- Run your IoT analytics automatically based on schedules or events.
- 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:

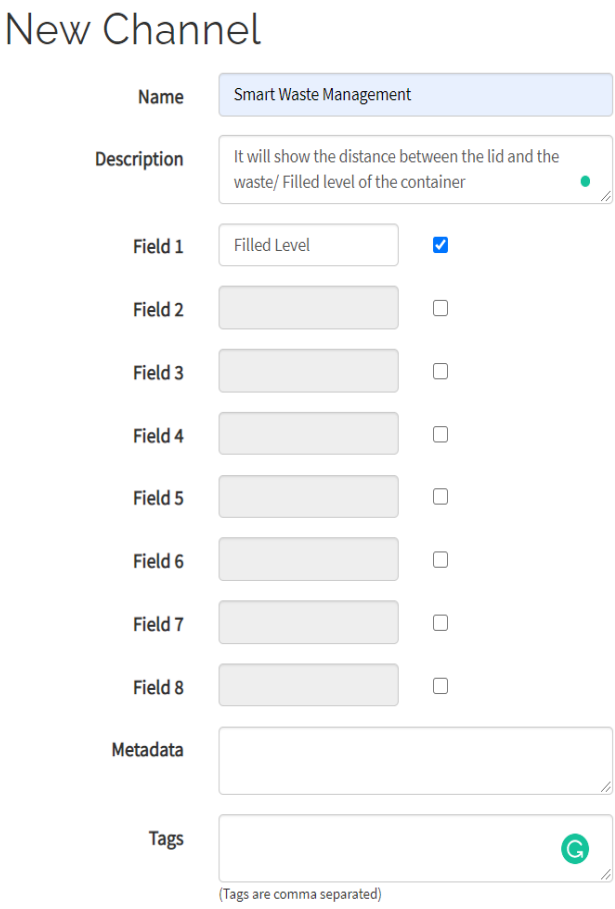
Step 1: Open the browser and visit thingspeak cloud using the following link.https://thingspeak.com/login?skipSSOCheck=true

Step 2: Login to your account.

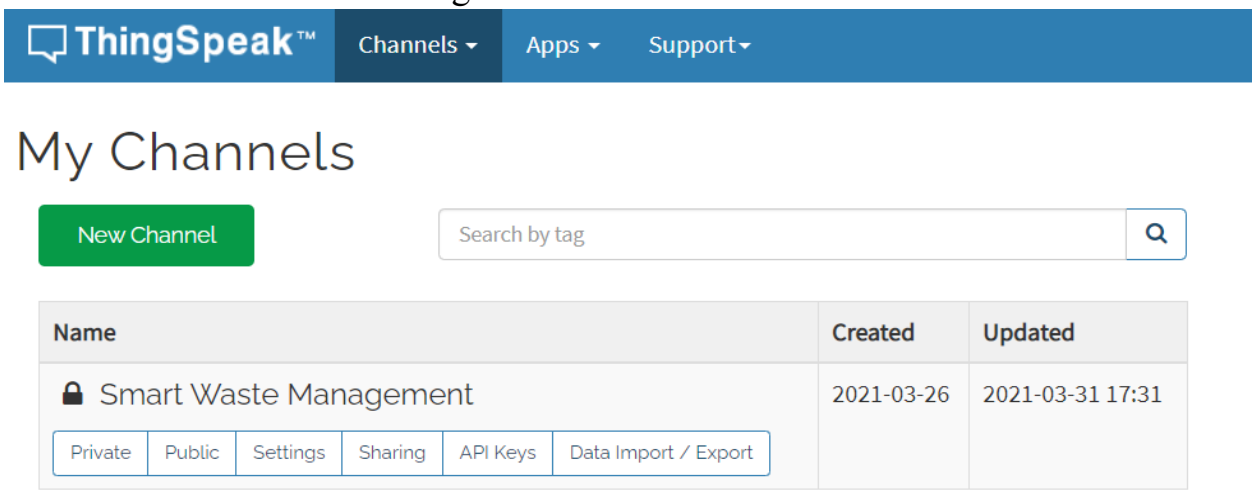
Step 3: Click on the new channel.



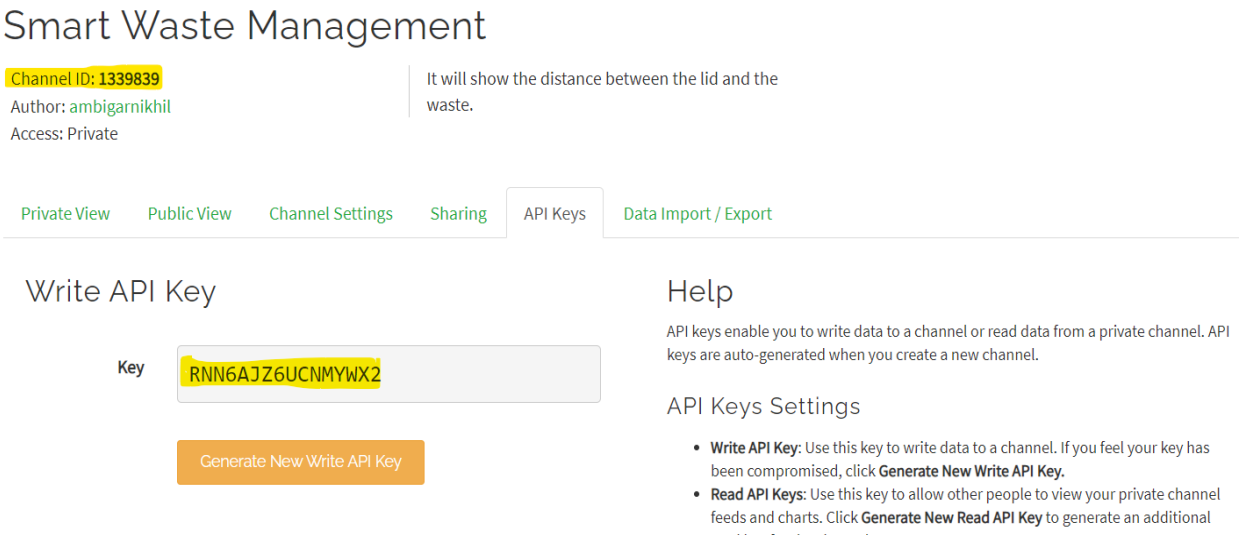
Step 4: Fill all the required details.



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

Code:

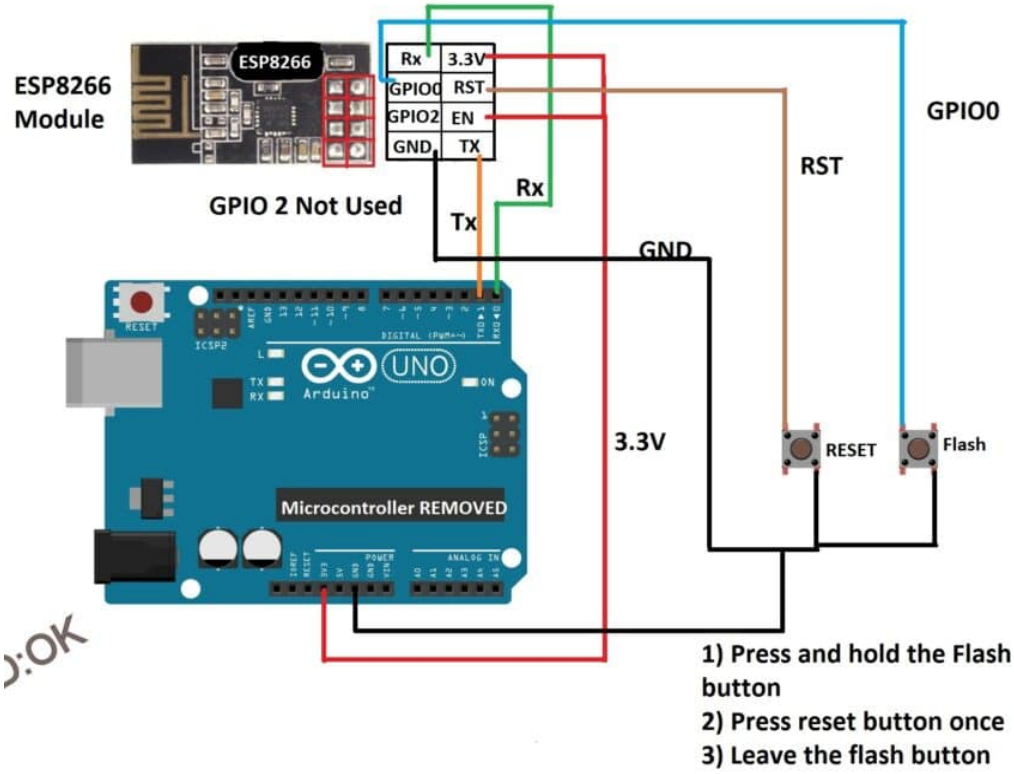
1. Uploading code to ESP8266 wireless transceiver module.

   Code:

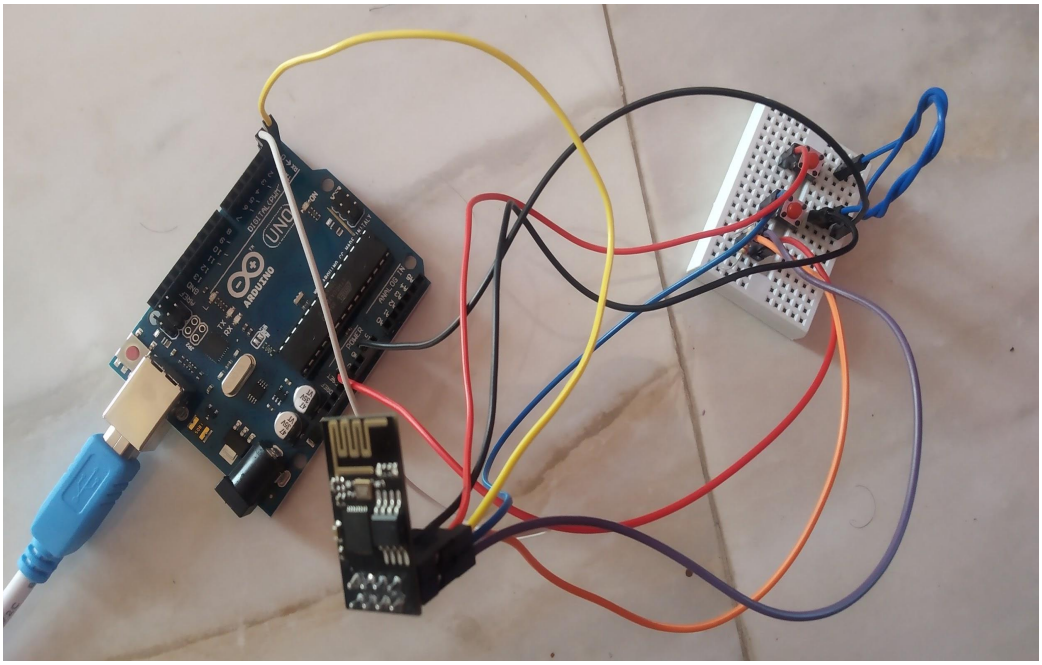
```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>

//-------- 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 = "";
}
```

To upload the above code into the esp8266 module we need to wire the circuit as shown below:



After wiring up the circuit looks like this:



- Press and hold the flash button now press the reset button once and leave the flash button. Pressing buttons in reverse sequence or pressing in some other way will result in failing to upload the program code.
- Then select the board as generic ESP8266 on your IDE.
- Select 9600 as baud rate and select the correct COM port.
- Then click the upload button, it will take a couple minutes to compile and another one minute to upload the code.

Once we successfully upload this code to ESP8266, we need to rewire and upload another code to the Arduino board.

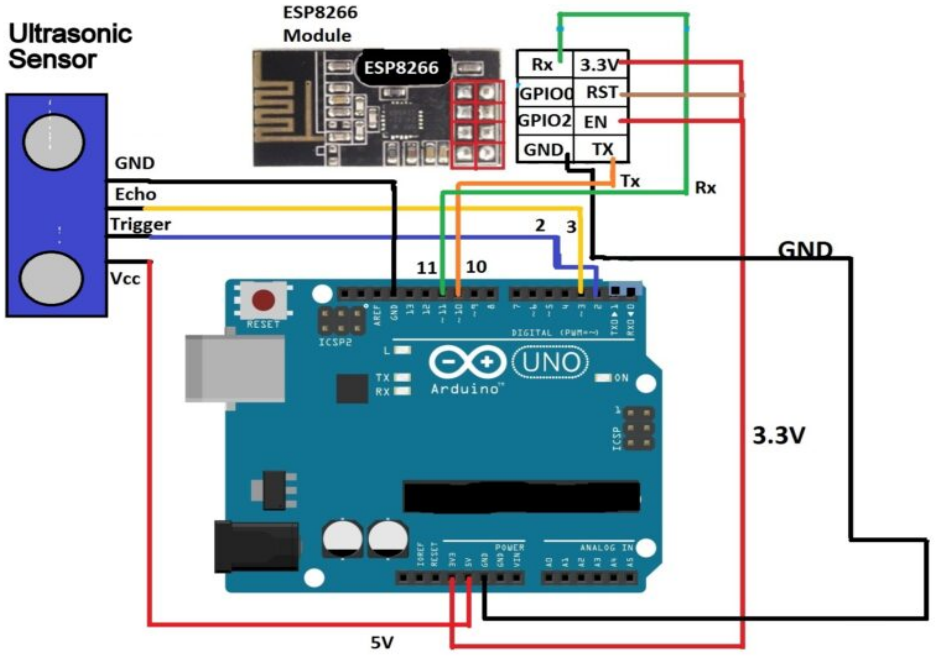
2. Uploading code to Arduino Uno:

Code:

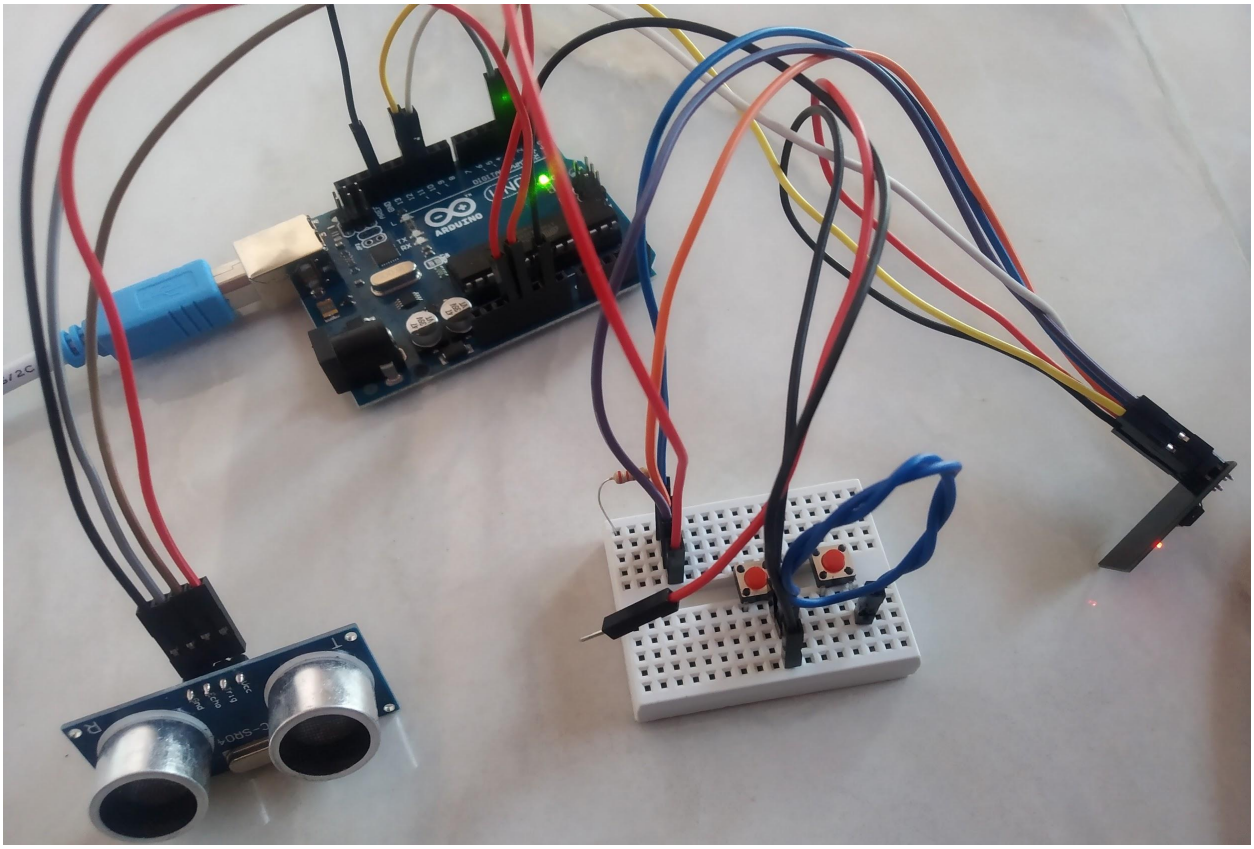
```
#include <SoftwareSerial.h>
SoftwareSerial ESP(10,11);
const int trigger = 2;
const int echo = 3;
```

```
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);
}
```

To upload the above code into the arduino uno we need to wire the circuit as shown below:



After wiring up the circuit looks like this:



After uploading the code to Arduino board, We should open the serial monitor at 9600 baud rate to see the measurement and open the thingspeak channel to see the incoming data on Thingspeak.

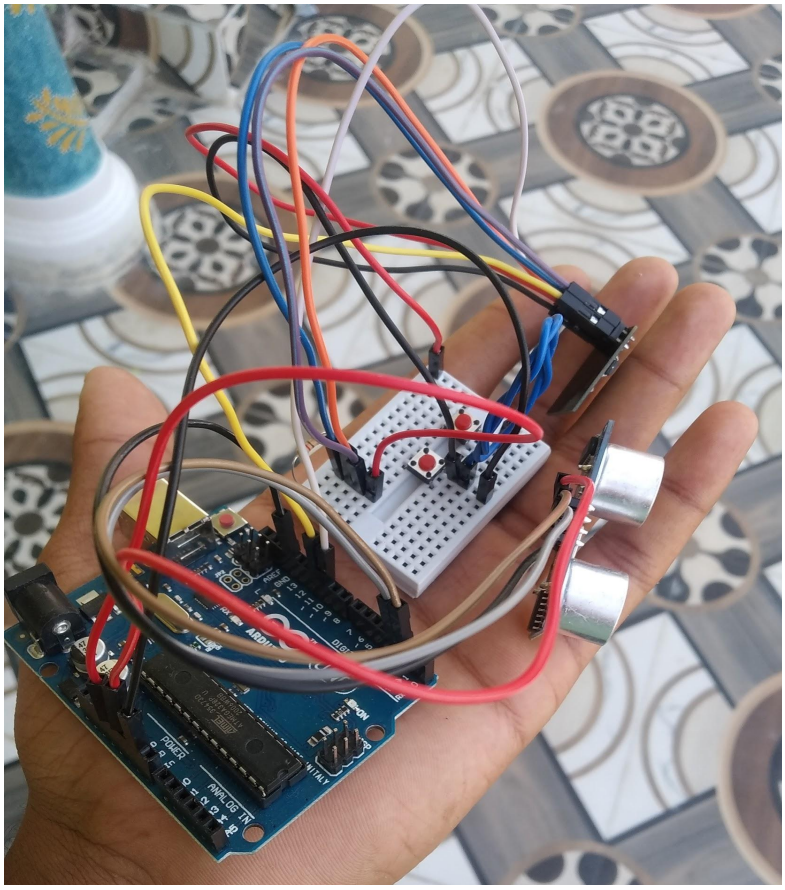
We made a video on the entire process of uploading code , it can be accessed via the following link.https://youtu.be/OqIzo4FaqsA

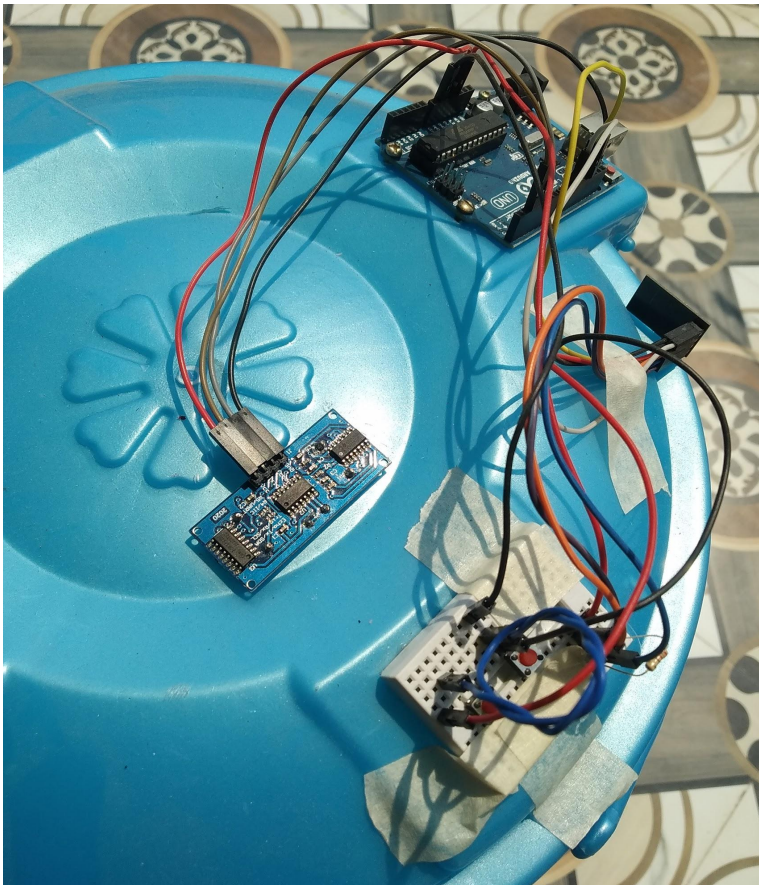
❖ Matlab analysis to read the channel data and send a trigger mail after reaching the threshold value:

Code:

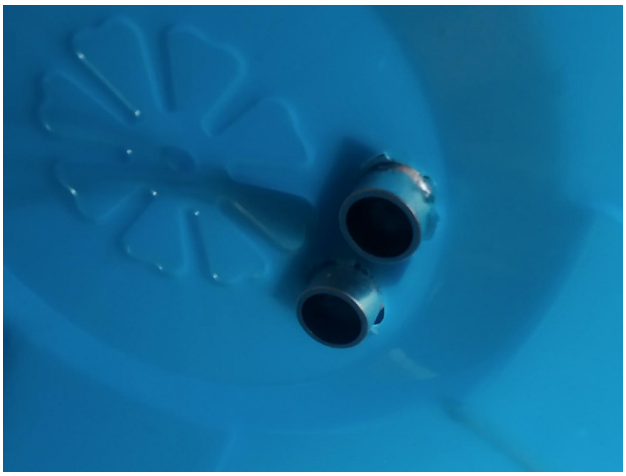
```
channelID = 1339839;

% 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
```

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





Interior view of the dustbin:



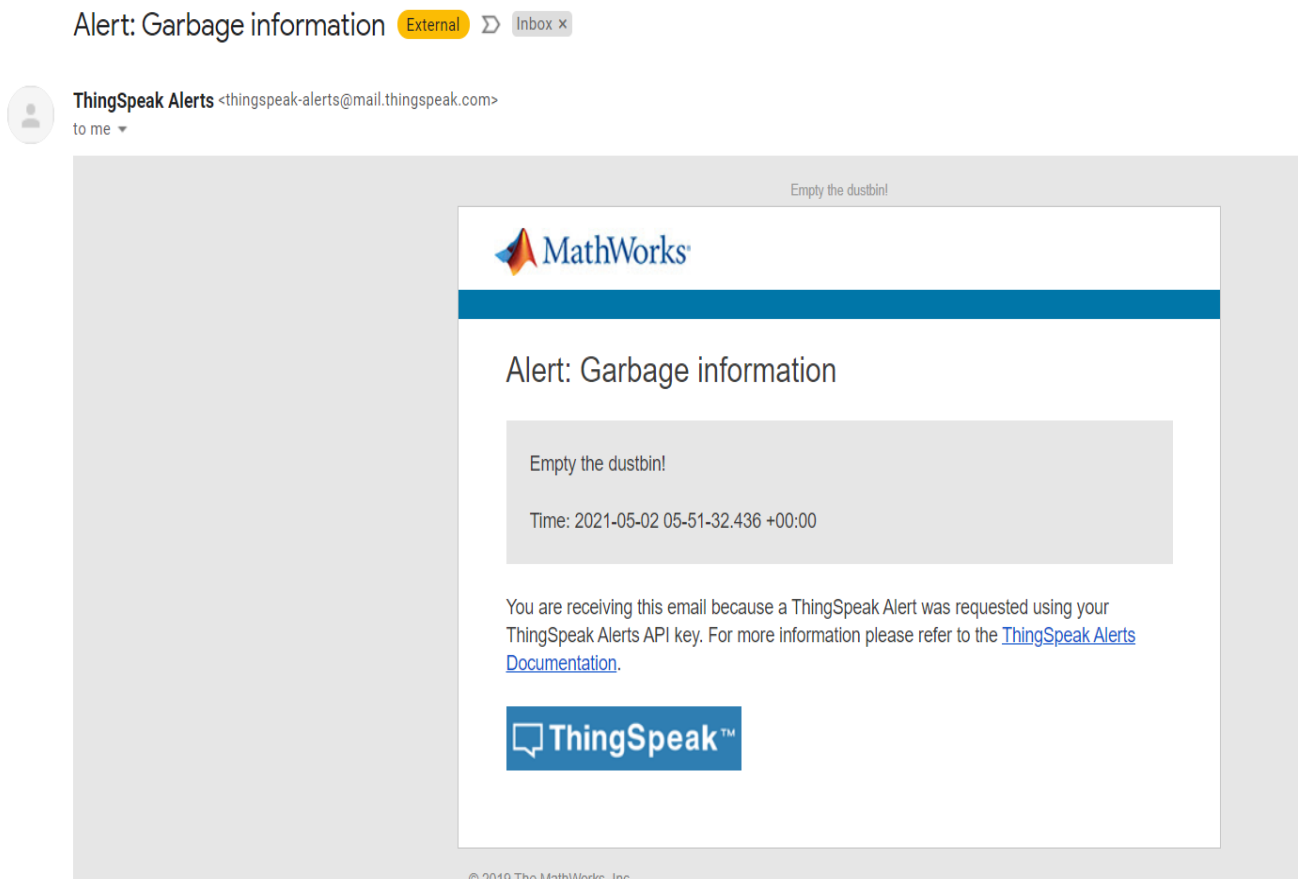
Demonstration video link:https://drive.google.com/file/d/1ZDDyE3r1UvYeT_7pusQywcre3h5chCBE/view?usp=sharing

❖ 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.



Comparing our model with other modules:

In this project, an integrated system of ESP8266 and Ultrasonic Sensor is introduced for efficient and economic garbage collection. “For real time implementation of our project we need to use the GSM module instead of ESP8266 “ .The developed system provides an improved database for garbage collection time and waste amount at each location. By implementing this project we will avoid overflowing of garbage from the container in residential areas which was previously either loaded manually or with the help of loaders in traditional trucks. It can automatically monitor the garbage level & send the information to the municipal corporation. The technologies which are used in the proposed system are good enough to ensure the practical and perfect for solid garbage collection process monitoring and management for a green environment.

We have a **future prediction** for our developed module, it predicts the future based on the previous data.

We have successfully completed the project “Smart Waste Management”, but there's still need for change.’

Disadvantages:

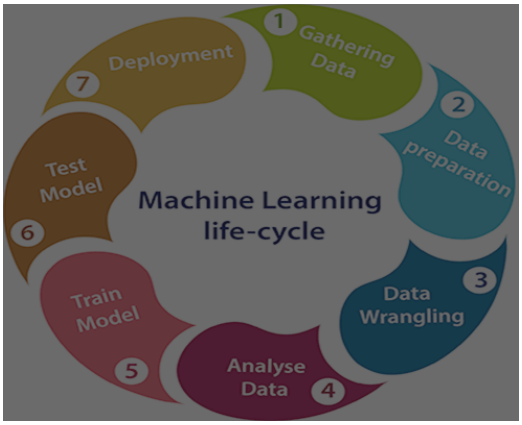
1)Developed models are not waterproof.

2)As we are using an ultrasonic sensor for the development of a module, the ultrasonic sensor produces sound waves which can be absorbed by some of the garbage elements, at which instance our model fails.

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.

References:

- Glossary of Environment Statistics : Series F, No. 67 / Department for Economic and Social Information and Policy Analysis, United Nations. New York: UN, 1997.

- Esp8266 Module-1 Datasheet.

- "Arduino - Introduction". Arduino.cc

- Thingspeak cloud servicehttps://in.mathworks.com/help/thingspeak/collect-data-in-a-new-channel.html

- Machine Learning Decision Tree Classification Algorithm https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

- Mean Square Error and R2 score Calculations.

Thank You!