# IOT BASED SMART CITY WASTE MANAGEMENT WITH CONNECTED TRASH CANS

## 1. INTRODUCTION:

### 1.1    Project Overview:

**Waste management system or waste disposal is a streamlined process that organizations use to dispose of, reduce, reuse, and prevent waste. It is also an approach where companies implement comprehensive strategies to efficiently manage wastes from their origin until their final disposal.**

**With the help of smart sensors that are located in smart waste bins you can easily track your waste containers. Smart sensors that are empowered with IOT technology will provide you the real-time information of your assets.**

### 1.2 purpose:

➢ Built-in sensors alert garbage collectors when it's time to pick up, eliminate offensive odors, and spruce up unsightly public trash cans, all thanks to data analytics and cutting-edge technology.

➢ Smart bin is the new high technology that integrated waste containers with smart sensors, which allows you to track through the waste management processes. For example, with the smart waste bin system you can control the occupancy ratio of your smart waste bins.

## 2. IDEATION & PROPOSED SOLUTION:

### 2.1  Problem statement definition.

   **Design a smart waste collection system that allows citizens to segregate the various types of solid waste they want to dispose and the municipal authorities to efficiently collect the same. The system should be mobile app (Android) based.**

 **It is seen in the market most of the dustbins are manually operated, so the user will have to move towards the dustbin to throw the waste. So, the existing dustbins are not user-friendly system because it only can be used by normal people and not for the person with disabilities.**

### 2.2 Empathy Map Canvas:

- ➢ The fill level of solid waste in each of the containers, which are strategically situated across the communities, is detected using ultrasonic sensors.
- ➢ A Wireless Fidelity (Wi-Fi) communication link is used to transmit the sensor data to an IOT cloud platform
- ➢ Depending on the fill level, the system sends appropriate notification message (in form of tweet) to alert relevant authorities and concerned citizen(s) for necessary action.
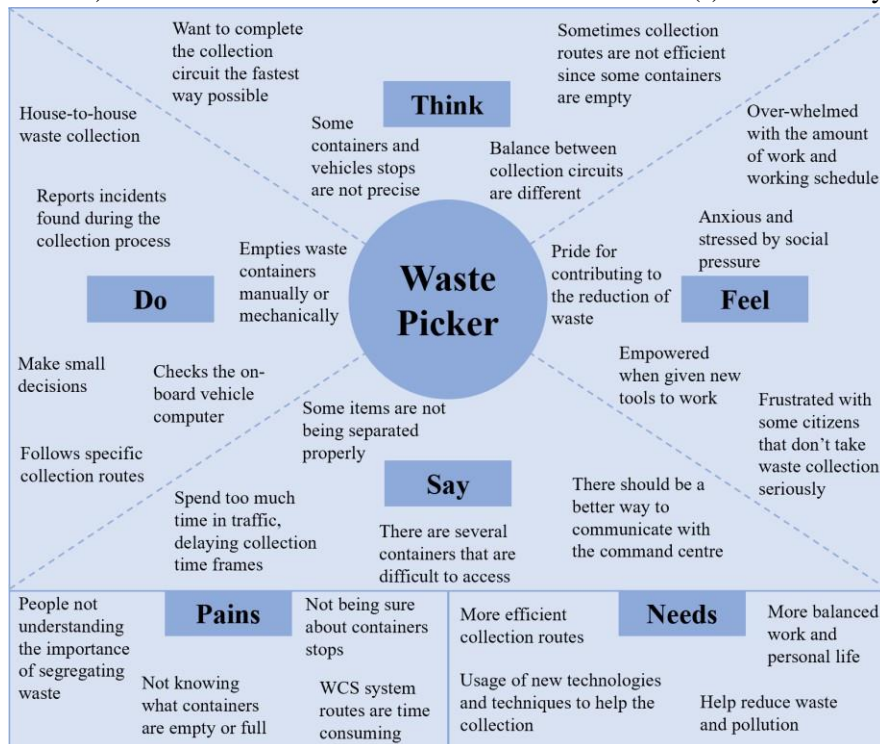


- • The empathy map canvas for a smart city waste management reveals the diverse needs, emotions, and concerns of the key stakeholders involved and emphasizing the need for accuracy and efficiency through automated meter readings.

## 2.3 Ideation & brainstorming:

- At the same time, smart waste management reduces the amount of overflowing waste and illegal dumping, which contributes to a safer and more sanitary working environment for waste collectors.
- Implement a system that automatically garbage level readings, eliminating the need for manual readings and reducing human errors.

**2.4 Proposed solution:**

| S.no. | Parameter | Description |
|-------|-----------|-------------|
| 1. | Problem Statement (Problem to be solved) | Major problem almost the cities are facing is to manage the waste that are generated daily. Currently emptying of the bin is done by corporation vehicles they need to visit the bin every day in a specified route, sometime bin will be overflowing by the garbage which attracts dogs and cattle's these animals will make the garbage to spill on the road. Secondly people won't go near to bin to throw their waste due to overfilled bins, Currently there is no bin is monitored remotely so complete automation is needed to monitor the bins remotely by the concerned persons and can inform the collecting vehicle to empty the bin when its full, by doing in these method fuel can be saved and spill over of waste can also be avoided |

| 2. | Idea / Solution description | The smart waste management with connected trashcans to automate the waste management process. It includes features such as automated meter reading, real-time data processing and accurate level of trash level measurement. The system provides insights into usage patterns and opportunities for the smart waste management |
|----|----|----|

| 3. | Novelty / Uniqueness | The smart waste management system stands out due to its integration of advanced technologies like IOT (Internet of Things) for automated meter reading Data analytics for Insights, and user-friendly interfaces for public hygiene needs. The system's ability to provide accurate dustbin level of the trashes and sustainability features makes it unique inthe market. |
|---|---|---|
| 4. | Social Impact / Customer Satisfaction | From a citizen's perspective, the social benefits of "smart bins" – besides their economic and environmental advantages – are interesting. They help to raise public awareness of utilizing renewable energy. improve street sanitation |
| 5. | Scalability of the Solution | The smart billing system is designed to be scalable, capable of handling a garbage level system of society. The system's architecture andinfrastructure are  Designed to support scalability, ensuring its long-term viability and growth. |

## 3. REQUIREMENT ANALYSYS:

### 3.1 Functional requirement:

- Garbage management
- Trash collection
- Notification
- E Commerce
- complaint
- Reporting and Analytics
- Integration and Interoperability
- Usability and User Experience

### 3.2 Non Functional requirement:

- Performance
- Reliability
- Security
- Usability
- Compliance
- Integration
- Maintainability

# 4. PROJECT DESIGN:

## 4.1 Data flow diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data Enters and leaves the system, what changes the information, and where data is stored

```
                    ┌──────────────┐
                    │     1.0      │
                    ├──────────────┤
                    │ Trash Reached│
                    │    Level     │
                    └──────┬───────┘
                           │
                    Sensor Send Signal
                           │
                           ▼
                    ┌──────────────┐
                    │     2.0      │
                    ├──────────────┤
                    │ Sensor Send  │
                    │  Signal to   │
                    │   Arduino    │
                    └──────┬───────┘
                           │
                    Connect to WIFI
                           │
                           ▼
                    ┌──────────────┐
                    │     3.0      │
                    ├──────────────┤
                    │ WIFI Shield  │
                    │ Connect to   │
                    │   Server     │
                    └──────┬───────┘
                           │
                   Connect to Server
                           │
                           ▼
                    ┌──────────────┐                ┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐
                    │     4.0      │                  DS1 │ Notification Table
                    ├──────────────┤──────────────▶ │                       │
                    │ Server Send  │                └ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘
                    │ Notification │
                    │ to all users │
                    └──────┬───────┘
                           │
                 Start Sending Notification
                           │
                           ▼
                    ┌──────────────┐
                    │     5.0      │
                    ├──────────────┤
                    │ Users Received│
                    │ Notification │
                    └──────────────┘
```
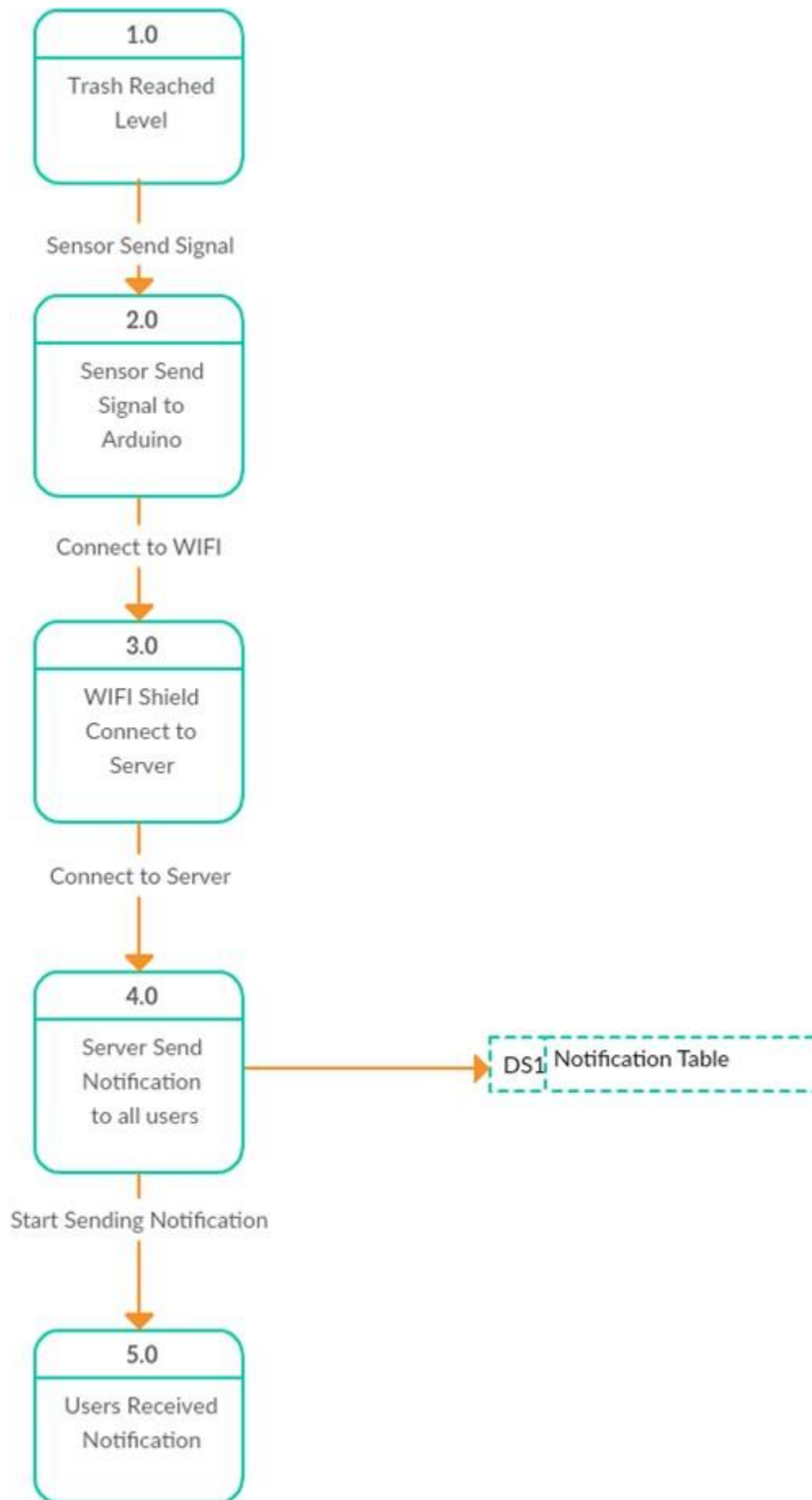
## 4.2 Solution & Technical Architecture

The system starts with smart level trash cans installed at public needs. These meters use IOT technology to collect real-time garbage usage data and transmit it securely to the central system. A reliable and secure communication network, such as a wireless or cellular network, is used to transmit data for the level of trash filled in the garbage notified to the user. This network ensures the continuous flow of meter data for trash level analysis
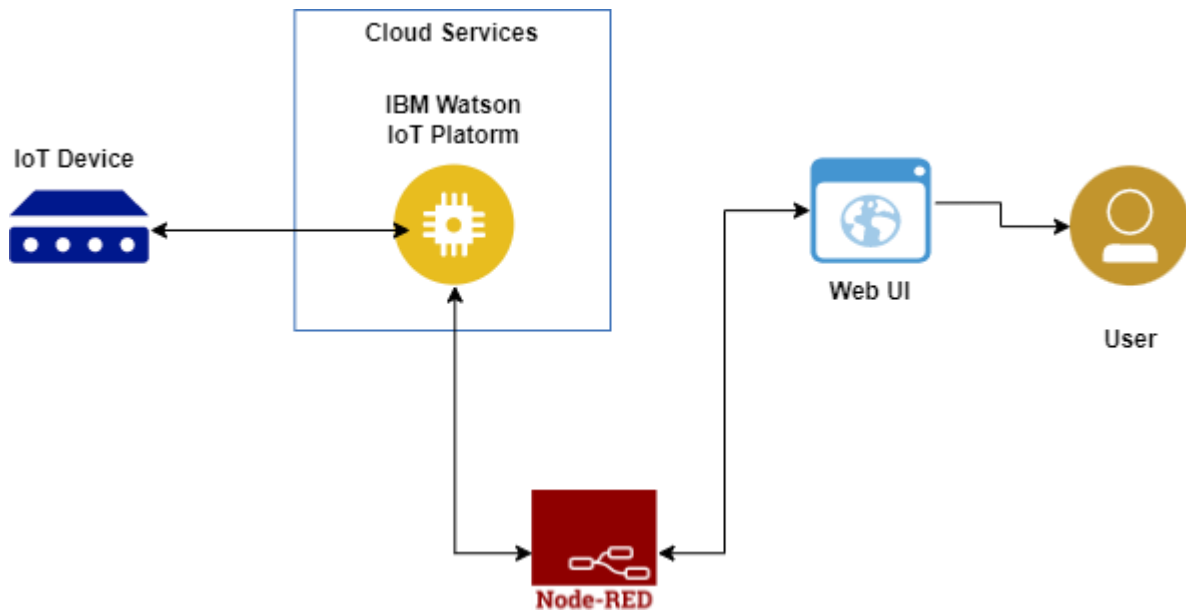
### 4.3 User stories:

- As a garbage collector, I want a smart waste management system that can accurately calculate the trash level of the garbage system

- As a garbage collectors, we want the notification level of how trashes are At full the message has been received to the municipality corporation

- As a garbage collector, we can receive the compliance regarding any issues against the waste processing due to ease of transmission of smart waste management

## 5. CODING & SOLUTIONING:

**5.1 Feature 1:** Dustbin level management

- The Dustbin level management feature allows level of dustbin. Here's an example solution using Python and a POST GRES SQL database:

**Python code:**

```
INT led= 13;
// the
pin that
the LED
is
attached
to

          INT sensor= 2;                // the pin that the sensor is attached to
          INT state = LOW;              // by default, no obstacle detected
          INT VAL=0;                    // variable to store the sensor status (value)
          INT pin no = 7;
          void setup() {
            Pin Mode(led, OUTPUT);
            Pin Mode(pin no, OUTPUT); // initialize LED as an output
            Pin Mode(sensor, INPUT);   // initialize sensor as an input
            Serial. Begin(9600);       // initialize serial
          }
          void loop(){
            Val = digital Read(sensor);  // read sensor value
            if (VAL == HIGH) {           // check if the sensor is HIGH
              Digital Write(led, HIGH);  // turn LED ON
              delay(100);                // delay 100 milliseconds

              if (state == LOW) {
                Serial. Print Ln("Dustbin is full!");
                state = HIGH;        // update variable state to HIGH
              }

            }
            else if(Serial. available()){
                char in Byte = Serial .read();

                Digital Write(pin no , HIGH);
                delay(1000);
                //Serial. Print Ln(in Byte);
              }
            else {
                Digital Write(led, LOW);
                Digital Write(pin no , LOW);// turn LED OFF
                delay(200);              // delay 200 milliseconds
```

```
            if (state == HIGH){
               Serial. print Ln("Dustbin is empty!");
               state = LOW;        // update variable state to LOW
        }
    }
}
```

**5.2 Database schema:**

**Table:** Containers

- id (Primary Key)
- address (String)
- latitude VARCHAR
- max Capacity  INT

**Table:** Tasks

- id Container
- vehicle
- order

i.  In this schema, the "Containers" table stores information, including their id, address, latitude and maximum capacity

ii. The "Tasks" table contains details about the id Container, trash vehicle and order which passed by waste collection

iii. These tables can be connected using foreign keys and relationships as required. This schema provides a basic structure to store trash can details and tasks which forges for waste management

# 6. RESULTS:

## 6.1 Performance metrics:

- **Waste level detection:** waste level inside the dustbin has been detected and level of wastes has been measured and sensed message to garbage system

- **Data access: The data can be accessed anytime and from anywhere in case of waste management system**

- **System Availability:** Track the uptime and availability of the smart waste management This metric measures the system's reliability and ensures that it isaccessible to people when needed.

- **Customer Satisfaction:** Gather feedback from customers to assess their satisfaction with the waste management for their concerned ecology need

## 7. ADVANTAGES & DISADVANTAGES:

**Advantages:**

- It saves time and money by using smart waste collection bins and systems equipped with fill level sensors. As smart transport vehicles go only to the filled containers or bins. It reduces infrastructure, operating and maintenance costs by upto 30%.
- It decreases traffic flow and consecutively noise due to less air pollution as result of less waste collection vehicles on the roads.
- This has become possible due to two way communication between smart dustbins and service operators.
- It further reduces manpower requirements to handle the garbage collection process.

**Disadvantage:**

- System requires more number of waste bins for separate waste collection as per population in the city. This resultsinto high initial cost due to expensive smart dustbins compare to other methods.
- It reduces man power requirements which results into increase in unemployment for skilled people.
- The training has to be provided to the people involved in the smart waste management system.

### 8. CONCLUSION:

- Due to the absence of sustainable waste management technology, the current waste disposal situation is likely to worsen. This work presents an enhanced solution to the problem of waste management by the littering of the garbage bins once they are full. Littering of the environment and the health hazards are minimized as timely disposal of the wastes isensured as the system automatically sends a message alert to the garbage collector or the management authority once the bin is full thereby ensuring that the bin is made empty to avoid dumping of refuse on the floor

### 9. FUTURE SCOPE:

- We can add GPS to this project. This will help to track the position in case there are more dustbins and also we can make separate dustbins for dry waste and wet waste . There are many birds and animals like Dog and Cat roaming around so we can add a cage to protect the dustbin from them.
- This management will create a cleanable environment and generation of waste level measurement

### 10. APPENDIX:

### Source code:

```
#include <WIFI .h>//library for WIFI

#include <PUB SUB CLIENT .h>//library for MQTT
#include "Ultrasonic .h"
Ultrasonic ultrasonic(2, 4);
float distance;
void callback(char* subscribe topic, byte* payload, unsigned
INT Pay Load Length);
//-------credentials of IBM Accounts------
#define ORG "6yocvj"//IBM ORGANITION ID
```

```
#define DEVICE_TYPE "Smart Dustbin"//Device type mentioned in
IBM Watson IOT Platform
#define DEVICE_ID "12345" //Device ID mentioned in IBM WATSON
IOT Platform
#define TOKEN "12345678"       //Token
String data3;
//float h, t;
//-------- CUSTOMIZE the above values --------
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";// Server Name
char publish Topic[] = "iot-2/EVS/Data/FMT/JSON";// topic name
and type of event perform and format in which data to be send
char Subscribe Topic[] = "iot-2/CMD/test/FMT/String";// CMD
REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char AUTH Method[] = "use-token-AUTH";// authentication method
char token[] = TOKEN;
char client Id[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID;//client id
//-------------------------------------------------------------------
WIFI Client WIFI Client; // creating the instance for WIFI client
PUBSUBCLIENT client(server, 1883, callback ,WIFI Client);
//calling the predefined client id by passing parameter like
server id ,port and WIFI credential
void setup()// configuring the ESP32
{
  Serial. begin(115200);


  delay(10);
 Serial. Print Ln();
 WIFI connect();
 MQTT connect();
}
void loop()// Recursive Function
{
  Distance = ultrasonic. read(CM);
  Serial. print("Distance in CM: ");
 Serial. Print Ln(distance);
  delay(1000);
 Publish Data(distance);
  delay(1000);
  if (!client. loop()) {
    MQTT connect();
  }
}
/*..................................retrieving to
Cloud .........................*/
void Publish Data(float distance) {
  MQTT connect();//function call for connecting to ibm
  /*
```

```
      creating the String in in form JSon to update the data to
IBM cloud
  */
  String payload = "{\"distance\":";
  payload += distance;

  payload += "}";

  Serial. print("Sending payload: ");
 Serial.  Print Ln(payload);

  if (client. publish(publish Topic, (char*) payload. C_STR())) {
    Serial. Print Ln("Publish ok");// if it successfully
uploaddata on the cloud then it will print publish ok in
Serial
monitor or else it will print publish failed
  } else {
    Serial. Print Ln("Publish failed");
  }

}
void MQTT connect() {
  if (!client. connected())
              {
    Serial. print("Reconnecting client to ");
   Serial. print Ln(server);
    While (!!! Client. connect(client Id, AUTH Method, token)) {
     Serial. Print (".");
      delay(500);
    }

    In it Managed Device();
    Serial. Print Ln();
  }
}
void WIFI connect() //function definition for WIFI connect
{
 Serial. Print Ln();
  Serial. Print("Connecting to ");
  WIFI. begin("WOKWI-GUEST", "", 6);//passing the WIFI
credentials to establish the connection
  while (WIFI.STATUS() != WL_CONNECTED) {
    delay(500);
   Serial. Print (".");
  }
 Serial. Print Ln("");
  Serial. Print Ln("WIFI connected");
  Serial. Print Ln("IP address: ");
  Serial. Print Ln (WIFI. LOCALIP());
}
void in it Managed Device() {
```

```
if (client. subscribe(subscribe topic)) {
```

```
      Serial. Print Ln((subscribe topic));
      Serial. Print Ln("subscribe to CMD OK");
  } else {
      Serial. Print Ln("subscribe to CMD FAILED");
  }
}
void callback(char* subscribe topic, byte* payload, unsigned
INT payload Length)
{

  Serial. print("callback invoked for topic: ");
  Serial. Print Ln(subscribe topic);

  for (INT i= 0; i < payload Length; i++) {
    //Serial. print((char)payload[i]);
    data3 += (char)payload[i];
  }

  Serial. print Ln("data: "+ data3);

data3="";

}
```