

PNT2022TMID31883

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

DOMAIN : IOT

TEAM ID : PNT2022TMID31883

TEAM LEADER : SANJAI.R

TEAM MEMBER 1 : SIVADHANUSH.T

TEAM MEMBER 2 : SUBHASRI.R

TEAM MEMBER 3 : YOUKHAN HARI.S.S

PROJECT OVERVIEW :

The main drive of solid waste management is the reduction and elimination of adverse effect of waste materials on human health and environment leading to improvement in quality of life. In this work, an intelligent solid waste monitoring system is developed using Internet of Things (IoT) and cloud computing technologies. This is a recent innovation as cloud computing has been applied, Ultrasonic sensors are employed to detect the fill level of solid waste in each of the containers. The data obtained by the sensor is then transmitted to an IoT cloud platform using a Wi-Fi communication link. For each designated fill level, the system sends appropriate notification message to alert relevant authorities for necessary action. Also, the fill level is monitored on web application in real-time.

ADVANTAGES:

- ➔ Resulting in less manpower.
- ➔ Keeps the environment clean and fresh.
- ➔ Saves the Earth and conserves energy.
- ➔ Reduces environmental pollution.

DISADVANTAGES:

- ➔ The resultant product has a short life.
- ➔ Needs More Global Buy-In.
- ➔ Significant Installation Costs.

BIN 1 :

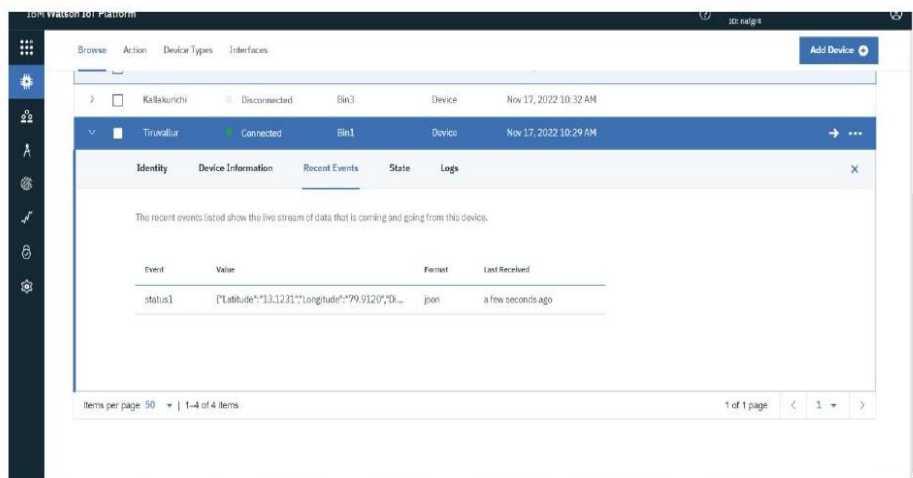
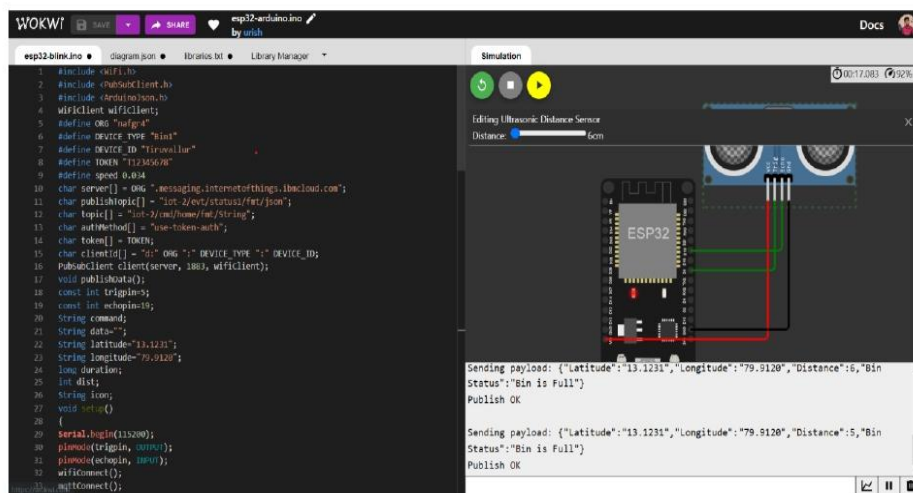
```
#include<WiFi.h>
#include<PubSubClient.h>
#include<ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "nafgr4"
#define DEVICE_TYPE "Bin1"
#define DEVICE_ID "Tiruvallur"
#define TOKEN "T12345678"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server,1883, wifiClient);
void publishData();
constint trigpin=5;
constint echopin=19;
String command;
String data="";
String latitude="13.1231";
String longitude="79.9120";
long duration;
int dist;
String icon;
void setup()
{
  Serial.begin(115200);
  pinMode(trigpin,OUTPUT);
  pinMode(echopin,INPUT);
  wifiConnect();
  mqttConnect();
}
void loop(){
  publishData();
  delay(500);
  if(!client.loop()){
    mqttConnect();
  }
}
void wifiConnect(){
  Serial.print("Connecting to ");
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST","",6);
  while(WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
```

```

}
Serial.print("WiFi connected, IP address: ");Serial.println(WiFi.localIP());
}
void mqttConnect(){
if(!client.connected()){
Serial.print("Reconnecting MQTT client to ");Serial.println(server);
while(!client.connect(clientId, authMethod, token)){
Serial.print(".");
Serial.print("*");
delay(1000);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice(){
if(client.subscribe(topic)){
Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
}
else{
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<20){
icon="Bin is Full";
}
else{
icon="Bin is not Full";
}
DynamicJsonDocument doc(1024);
String payload;
doc["Latitude"]=latitude;
doc["Longitude"]=longitude;
doc["Distance"]=dist;
doc["Bin Status"]=icon;
serializeJson(doc, payload);
delay(3000);
Serial.print("\n");
Serial.print("Sending payload: ");

```

```
Serial.println(payload);
if(client.publish(publishTopic, (char*) payload.c_str())){
Serial.println("Publish OK");
}
else{
Serial.println("Publish FAILED");
}
}
```



BIN 2 :

```
#include<WiFi.h>
#include<PubSubClient.h>
#include<ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "nafgr4"
#define DEVICE_TYPE "Bin2"
#define DEVICE_ID "Chennai"
#define TOKEN "C12345678"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server,1883, wifiClient);
void publishData();
constint trigpin=5;
constint echopin=19;
String command;
String data="";
String latitude="13.0827";
String longitude="80.2707";
long duration;
int dist;
String icon;
void setup()
{
  Serial.begin(115200);
  pinMode(trigpin,OUTPUT);
  pinMode(echopin,INPUT);
  wifiConnect();
  mqttConnect();
}
void loop(){
  publishData();
  delay(500);
  if(!client.loop()){
    mqttConnect();
  }
}
void wifiConnect(){
  Serial.print("Connecting to ");
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST","",6);
  while(WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
```

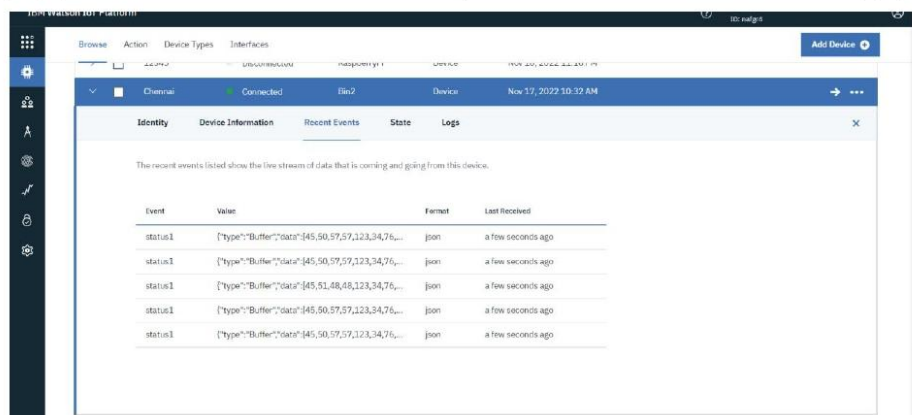
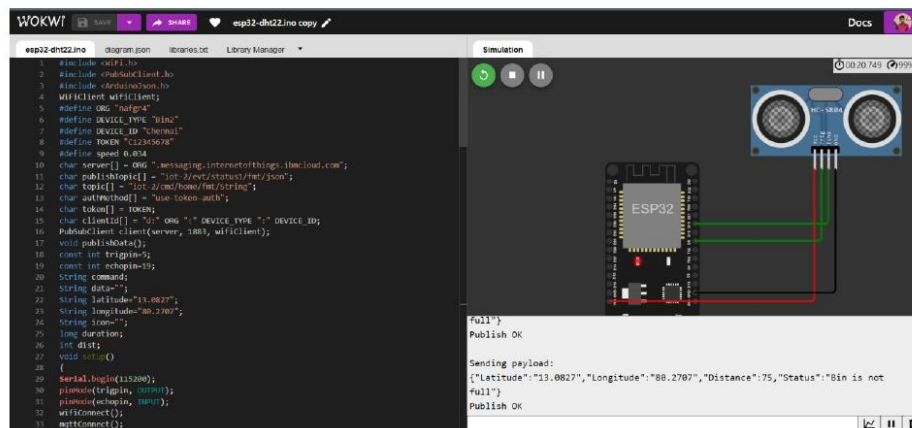
```

}
Serial.print("WiFi connected, IP address: ");Serial.println(WiFi.localIP());
}
void mqttConnect(){
if(!client.connected()){
Serial.print("Reconnecting MQTT client to ");Serial.println(server);
while(!client.connect(clientId, authMethod, token)){
Serial.print(".");
Serial.print("*");
delay(1000);
}
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice(){
if(client.subscribe(topic)){
Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
}
else{
Serial.println("subscribe to cmd FAILED");
}
}
}
void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<20){
icon="Bin is Full";
}
else{
icon="Bin is not Full";
}
}
DynamicJsonDocument doc(1024);
String payload;
doc["Latitude"]=latitude;
doc["Longitude"]=longitude;
doc["Distance"]=dist;
doc["Bin Status"]=icon;
serializeJson(doc, payload);
delay(3000);
Serial.print("\n");
Serial.print("Sending payload: ");

```

PNT2022TMID31883

```
Serial.println(payload);
if(client.publish(publishTopic,(char*) payload.c_str())){
Serial.println("Publish OK");
}
else{
Serial.println("Publish FAILED");
}
}
```



Bin3

```
#include<WiFi.h>
#include<PubSubClient.h>
#include<ArduinoJson.h>
WiFiClient wifiClient;
```


BIN 3 :

```
#define ORG "nafgr4"
#define DEVICE_TYPE "Bin3"
#define DEVICE_ID "Kallakurichi"
#define TOKEN "K12345678"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin = 5;
const int echopin = 19;
String command;
String data = "";
String latitude = "11.7348";
String longitude = "78.9639";
String icon = "";
long duration;
int dist;
void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}
void loop() {
  publishData();
  delay(500);
  if (!client.loop()) {
    mqttConnect();
  }
}
void wifiConnect() {
  Serial.print("Connecting to ");
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}
void mqttConnect() {
```

```

if(!client.connected()){
Serial.print("Reconnecting MQTT client to ");Serial.println(server);
while(!client.connect(clientId, authMethod, token)){
Serial.print(".");
Serial.print("*");
delay(1000);
}
initManagedDevice();
Serial.println();
}
}

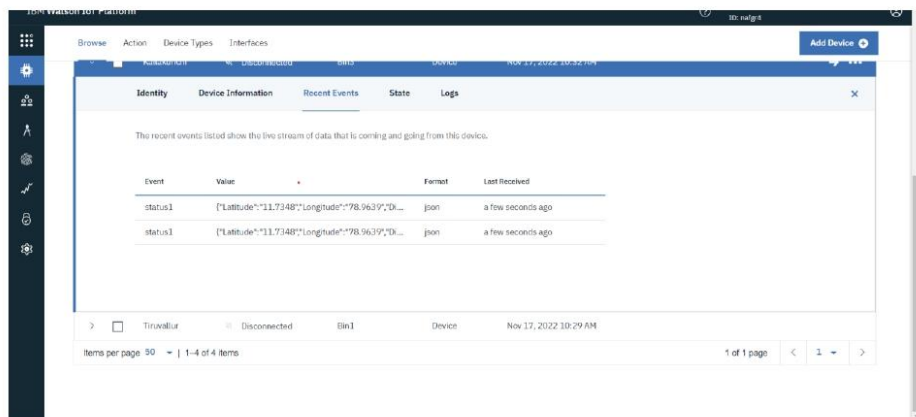
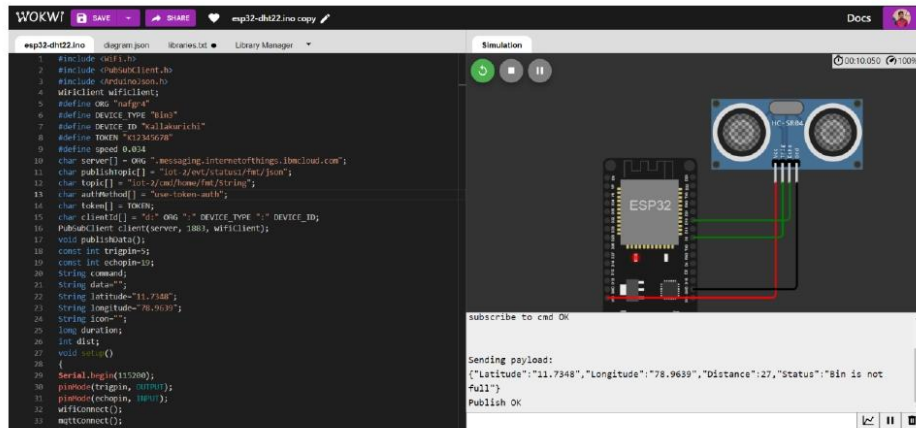
void initManagedDevice(){
if(client.subscribe(topic)){
Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
}
else{
Serial.println("subscribe to cmd FAILED");
}
}

void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<20){
icon="Bin is full";
}
else{
icon="Bin is not full";
}
DynamicJsonDocument doc(1024);
String payload;
doc["Latitude"]=latitude;
doc["Longitude"]=longitude;
doc["Distance"]=dist;
doc["Status"]=icon;
serializeJson(doc, payload);
delay(3000);
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if(client.publish(publishTopic,(char*) payload.c_str())){
Serial.println("Publish OK");
}
}

```

PNT2022TMID31883

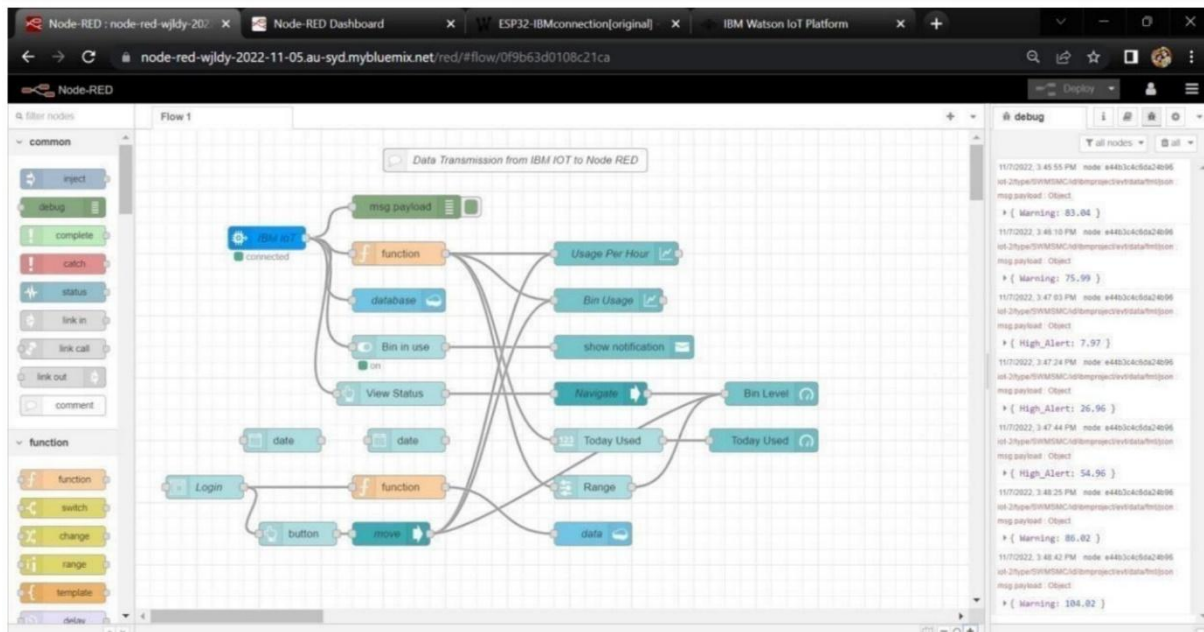
```
else{
  Serial.println("Publish FAILED");
}
}
```



Final Output



Data Transfer from IBM Watson Platform to Node Red :



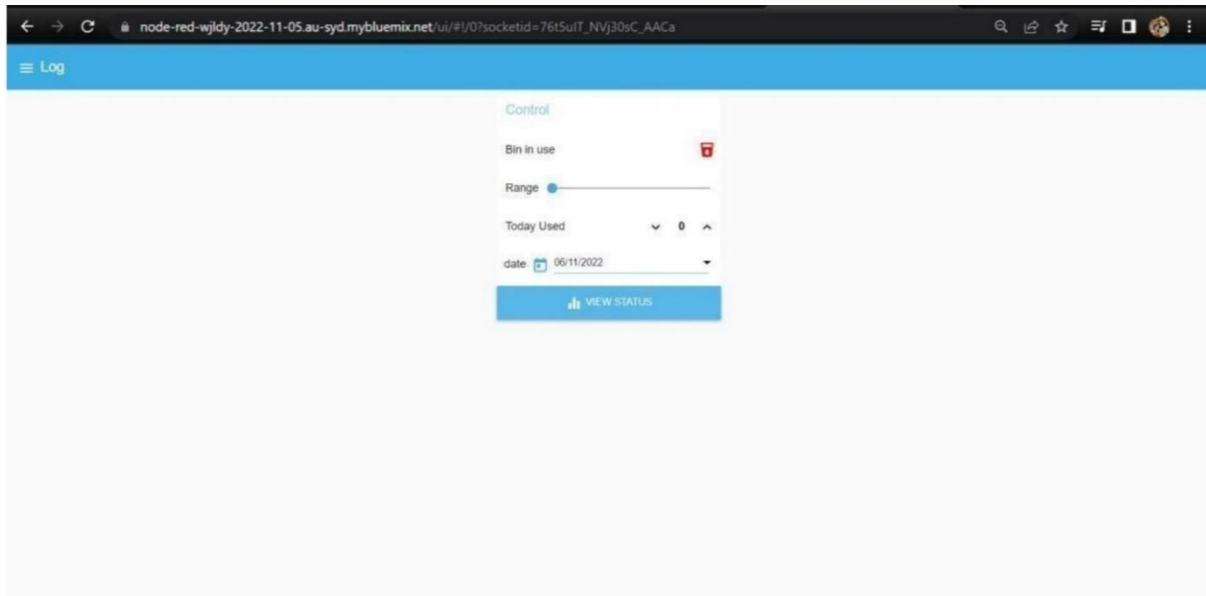
Storing Database in IBM Cloudant DB :

The screenshot shows the IBM Cloudant Databases dashboard. The "Your Databases" section contains a table with the following data:

Name	Size	# of Docs	Partitioned	Actions
login_credentials	13.7 KB	111	No	[Icons for edit, delete, and other actions]
noderedwjldy20221105	37.4 KB	4	No	[Icons for edit, delete, and other actions]
sample	59.4 KB	351	No	[Icons for edit, delete, and other actions]
sensor_data	15.7 KB	90	No	[Icons for edit, delete, and other actions]

PNT2022TMID31883

Web UI :



CONCLUSION :

The behaviour of generating garbage is too dangerous not only for today's generation, but also for future generations. It is critical to educate people and encourage them to practice Recycle, Reuse, and Reduce instead of producing waste. Waste disposal should be a priority for municipalities and governments.

PNT2022TMID31883

PNT2022TMID31883

PNT2022TMID31883

PNT2022TMID31883

PNT2022TMID31883

PNT2022TMID31883