IoT based Smart Efficient Waste Management System using Embedded IoT devices: NodeMCU & Rasberry Pi

[1]Jignesh J. Patoliya & [2]Miral M. Desai
[1]Assistant Professor, [2]Assistant Professor
[1]Department of EC Engineering,
[1]CSPIT, CHARUSAT, Changa, Anand, Gujarat, INDIA

Received: April 02, 2019 Accepted: May 03, 2019

ABSTRACT: : *Population of the country as well as the world is increasing day by day. As population is increased, it will increase the waste of basic requirement of life like water and food. People are throwing the waste and splitting anywhere. Government is keeping waste baskets in different zones of the city. But maintenance of those dustbins is not up to the mark. Sweepers hired by the Government are not doing their job honestly. Sometimes people are throwing the waste in the full dustbin. Ultimately all this things invite the dieses and illness. To support "Swatch Bharat Abhiyan", it requires designing Smart waste management system which supports smart city. Smart waste management system can be design using the IoT (Internet of things). In the proposed system, smart dustbins are arranging in the various area of the city which gives the information regarding dustbins is full or empty. Using Sensors interfaced with the cloud, the depth of all bins is displayed on to the server. It will provide the information regarding status of the bins of respective area of the city to the Municipal Corporation. Municipal corporation is managing bins of the various area of the city according to the information of the main server. System can be extended by sensing wet type waste and dry type waste and update the filtered status on to the main server using IoT.*

Key Words: *IoT, MQTT, PyQt, Waste Management.*

I. Introduction

The idea smart bins or smart waste management system have been in discussion since the idea of smart city is coming into the picture. All the idea of smart bins is reached to the same objective but the implementation method is different. Implementation method will be reliable, if the flavor of IoT is added into the method. The proposed system is working as the real time indicator of the garbage level in the dustbin at any given time. As we know that garbage bins around the town is collected waste twice in a day and there is a chance of inefficiency to collect the waste. Because its not necessary that all the garbage bins are filling up at the maximum level at the same time. For example let's say street A is a most busy street. So that garbage fills in the bins are very fast whereas in street B with low population even after two days the bin is not even half full. The proposed system gives a real time indicator of the garbage level in dustbin at any given time. By optimizing this data of waste collection on to the server, we can take necessary action. The proposed system is providing better and more convenient services in the following ways [1]:

- Stop overflowing of dustbins along roadside as smart bins are managed real time.
- Operational efficiency and waste reduction
- Smart city will be greener and more people friendly
- Dynamic and Innovative economy.

The functional flow diagram of the proposed system is as follows:

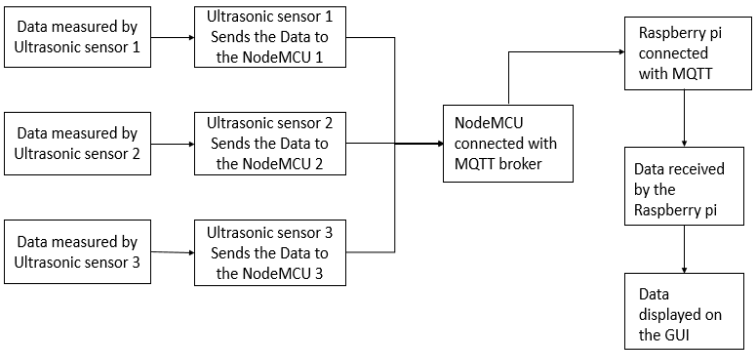

Fig. 1 Functional Flow Diagram of Proposed System

As shown in Functional Flow diagram of proposed system (Fig.1), waste is measured in individual dustbins and the measured data of waste quantity is send to the controlling device. Controlling device (Node MCU) is already connected with the cloud. So measured data sent to the cloud via controlling device. On other side, another controlling device (Raspberry Pi) which is also connected with the cloud, will fetch the data and send the status of the data onto the GUI.

The proposed paper is organized in the given form. Paper starts with the Introduction as Section I; the objective of the system is described in Section II. Proposed Block Diagram is described in Section III. System Requirements are described in Section IV. Literature Review is described in Section V. Section VI describes Software Environment. Experimental Setup describe in Section VII. Paper is concluded with the references.

II. Objective

To design smart waste management system or smart dustbin based on IoT Platform. It will be used to clean city or urban area very quickly as the all dustbins of respective zone are continuously under monitor. The proposed system will work as the fundamental block for designing the system as "Smart City".

III. Proposed Block Diagram

The proposed block diagram of suggested system is shown in Fig.2

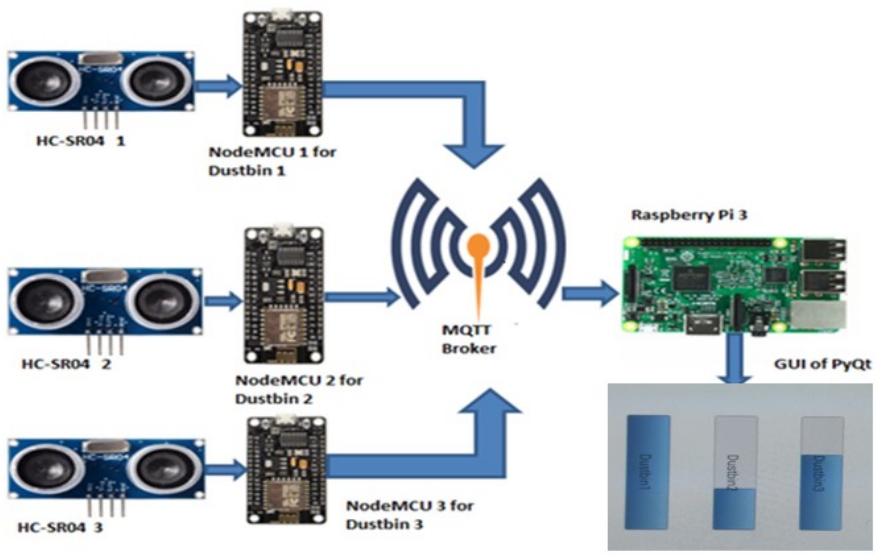

Fig. 2 Proposed Block Diagram

As shown in proposed block diagram (Fig.2), Individual Ultrasonic sensors are connected on the top of the individual dustbins. Ultrasonic sensors are used to measure the depth of the waste quantity. Ultrasonic sensors will give the data of the waste quantity to the IoT module (Node MCU). Node MCU is connected with the cloud via MQTT Protocol. So measure quantity of the waste data will be send to the cloud. At other side, another IoT module (Raspberry Pi) is also connected with the cloud via MQTT broker will subscribe the data of waste quantity measured in individual bins. Raspberry Pi will display measured data on the GUI using PyQt.

IV. System Requirements

To design IoT based smart waste management system, it must fulfill the following requirements.

4.1Hardware Requirements

For designing the proposed system we require following hardware.

- Node MCU (Individual for individual bins)
- Ultrasonic sensors (3 Nos.)
- Raspberry pi 3

3.2 Software Requirements

Following software requirement are require designing proposed system.

- Arduino C
- MQTT Broker

V. Literature Review

5.1 IoT Module - Node MCU

As shown in Fig. 4, NodeMCU is An IoT module comes with a built in Wi-Fi chip as ESP8266, USB connector and rich assortment of general purpose Input-output Pins.

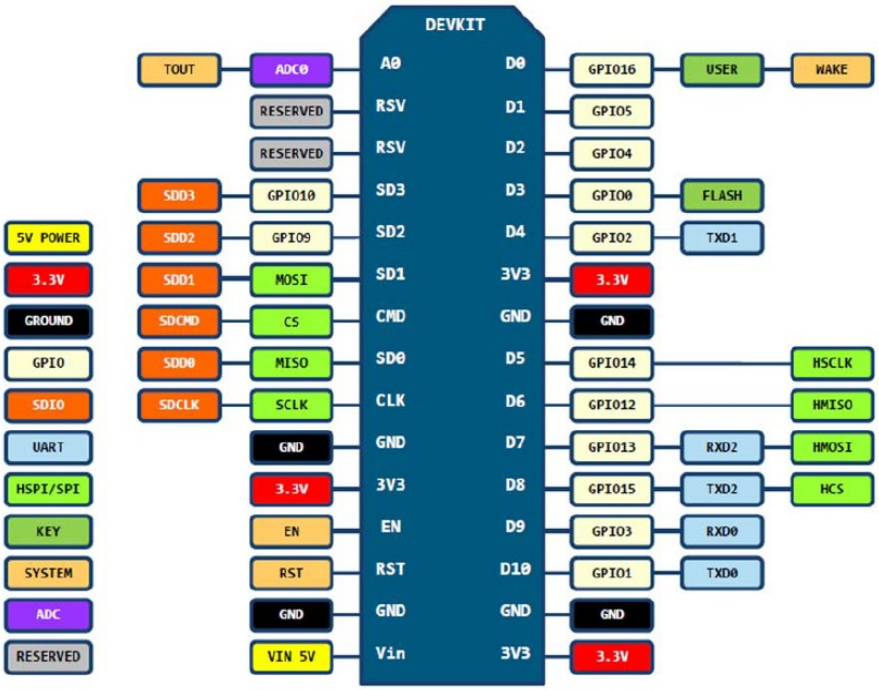

Fig. 3 NodeMCU Pin-Out Diagram

The Technical Specification of NodeMCU is as follows [2]:

- Voltage: 3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Ten silica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

5.2 Ultrasonic Sensor

To measure the level of waste in dustbin, Ultrasonic sensor is used in suggested system. The HC-SR04 is an ultrasonic ranging module. This economical sensor provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm. Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit. Technical Specification of Ultrasonic sensor is as follows [3]:

- Operating Voltage: 5V DC
- Operating Current: 15mA
- Measure Angle: 15°

Fig. 4 Ultrasonic Sensor

5.3 Raspberry Pi 3

Raspberry Pi 3 is a single borad computer which is as small as debit card/credit card. The Raspberry Pi 3 board is shown in Fig. 4. Raspberry Pi 3 board contains ARM processor additional with inbuilt Bluetooth and inbuilt wifi. Raspberry Pi boards does not contain on chip ADC. Raspberry pi board has on chip camera slot and picamera is also available in the market. Because of on chip camera Raspberry pi board is also used for designing Image and Video processing based applications.

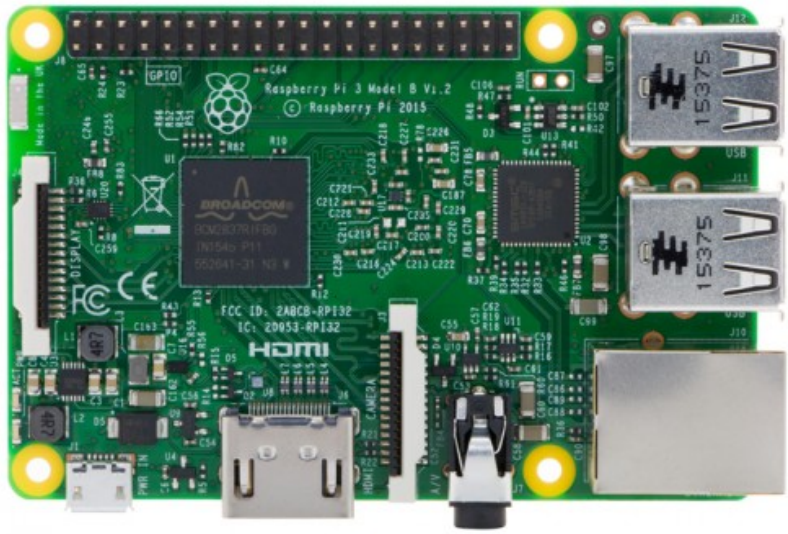

Fig. 5 Raspberry Pi 3 board

The Technical Specification of Raspberry Pi 3 board is as follows [4]:

- **SoC:** Broadcom BCM2837
- **CPU:** 4× ARM Cortex-A53, 1.2GHz
- **GPU:** Broadcom VideoCore IV
- **RAM:** 1GB LPDDR2 (900 MHz)
- **Networking:** 10/100 Ethernet, 2.4GHz 802.11n
- **Bluetooth:** Bluetooth 4.1 Classic, BLE
- **Storage:** microSD
- **GPIO:** 40-pin header, populated
- **Ports:** HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial, Interface (CSI), Display Serial Interface (DSI)

5.4 Arduino C

Arduino C is the derived programming language from C/C++ which contains in built hardware based libraries.

5.5 MQTT Protocol [5]

MQTT stands for Message Queue Telemetry Transport. It is a lightweight message queuing and transport protocol. It's a broker based publish/ subscribe messaging protocol. MQTT, as its name implies, is suited for the transport of telemetry data (sensor and actor data). MQTT has a client/server model, where every sensor is a client and connects to a server, known as a broker, over TCP. MQTT is message oriented. Every message has a discrete chunk of data, opaque to the broker. Every message is published to an address, known as a topic. Clients may subscribe to multiple topics. Every client subscribed to a topic receives every message published to the topic. For example, imagine a simple network with three clients and a central broker (Fig.5). All three clients open TCP connections with the broker. Clients B and C subscribe to the topic temperature. At a later time, Client A publishes a value of 22.5 for topic temperature. The broker forwards the message to all subscribed clients (Fig 6).

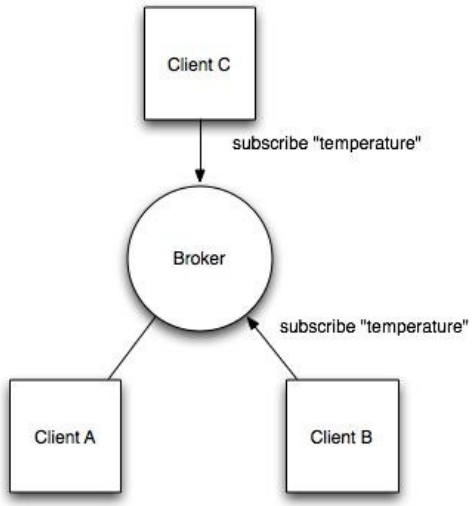

Fig. 6 MQTT Architecture-1

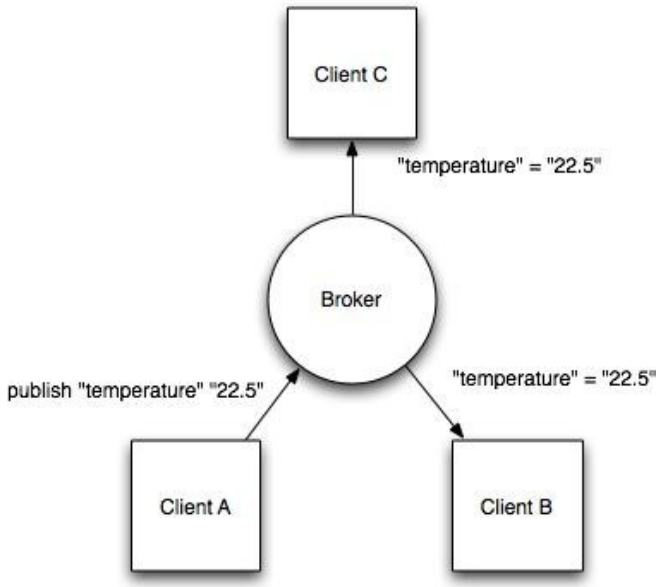

Fig. 7 MQTT Architecture-2

VI. SOFTWARE ENVIRONMENT

The whole software environment is created using Arduino C language additionally with wifi based API and MQTT library. GUI is created in the Raspberry Pi using PyQt Library. The detail software flow diagram for suggested system is shown in Fig. 8.

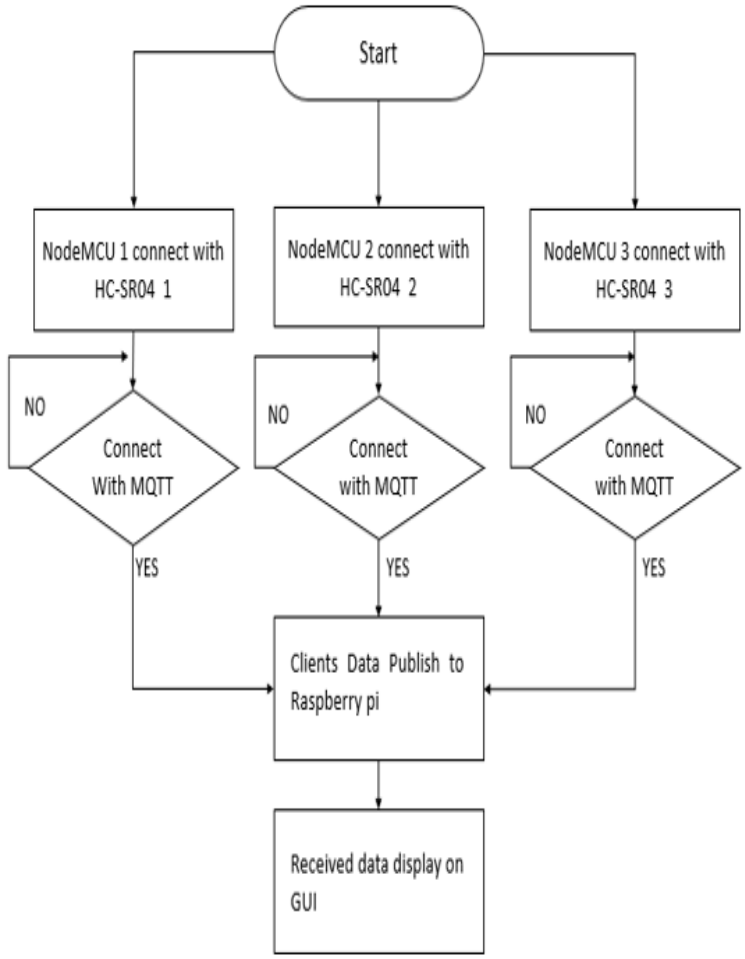

Fig. 8 Software Flow Diagram

VII. EXPERIMENTAL SETUP

Experimental Setup of the suggested system is shown in Fig. 9, 10 & 11 respectively. Fig. 9 shows the side view of Experimental setup where level of waste quantity is shown in all three smart bins. It shows one level as high, second level as medium and third level as low. Fig. 10 shows the arrangement of all three IoT modules along with separately interfaced ultrasonic sensors. Here ultrasonic sensor measured the depth of waste level and it will publish the measured data on to the server via Node MCU. Fig 11 shows the Raspberry Pi as a part of Experimental Setup where Node MCU and Raspberry Pi are communicated using MQTT protocol. The measured data of waste quantity subscribed by the Raspberry Pi and displayed as server of GUI using PyQT library [6].

Fig. 9 Experimental Setup Side View

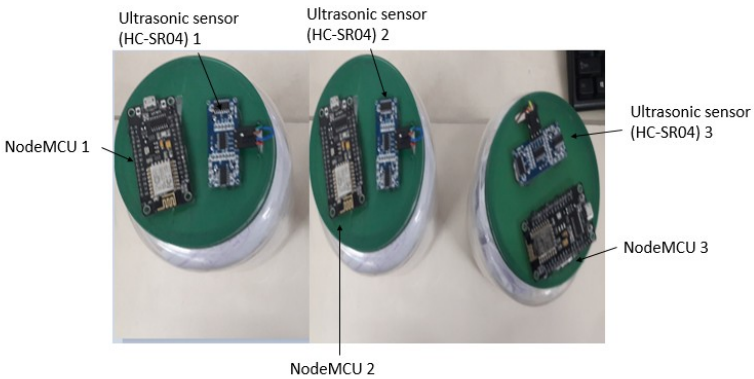

Fig. 10 Experimental Setup Top View

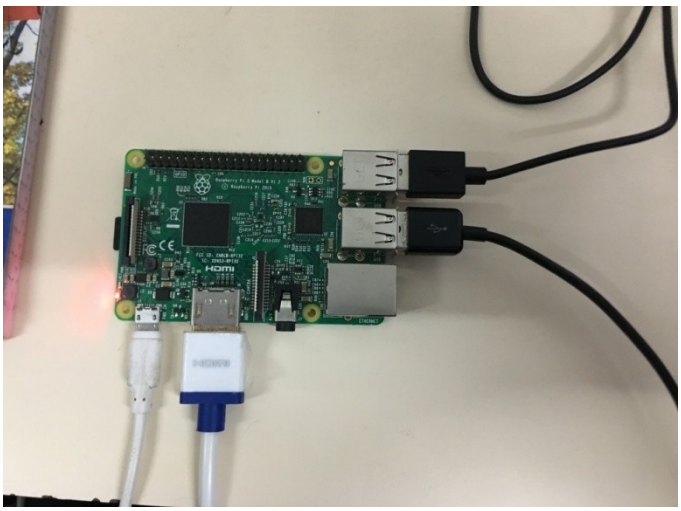

Fig. 11 Experimental Setup – Raspberry Pi as IoT module

VIII. Test Results & Discussion

Test results of the proposed system are discussed in Fig. 12, Fig. 13 and Fig. 14 respectively. Fig. 12 shows that Node MCU first providing its IP address and then successfully connected with the cloud via MQTT Protocol. Here NodeMCU is working in Publisher Mode. Fig. 13 shows the data of waste level of individual dustbins publish on cloud and also in parallel same information display on serial monitor. Fig. 14 shows the GUI which is displayed using Raspberry Pi through PyQt library. As shown in Fig. 14, the waste level of Individual dustbins is updated run time as the event occurring at the NodeMCU section. Here Raspberry Pi is working as Subscriber Mode which subscribe the data of waste level from cloud and display on GUI.

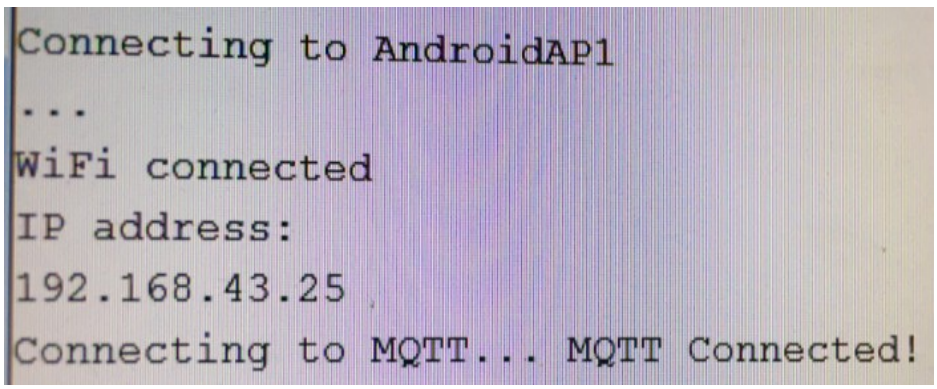

Fig. 12 NodeMCU connected with Broker of MQTT

```
Connecting to MQTT... MQTT Connected!
 Dustbin level at 100%
 Dustbin level at 100%
 Dustbin level at 100%
 Dustbin level at 100%
 Dustbin level at 75%
 Dustbin level at 75%
 Dustbin level at 75%
 Dustbin level at 75%
 Dustbin level at 75%
 Dustbin level at 75%
 Dustbin level at 50%
 Dustbin level at 50%
```

Fig. 13 NodeMCU publishing data on cloud

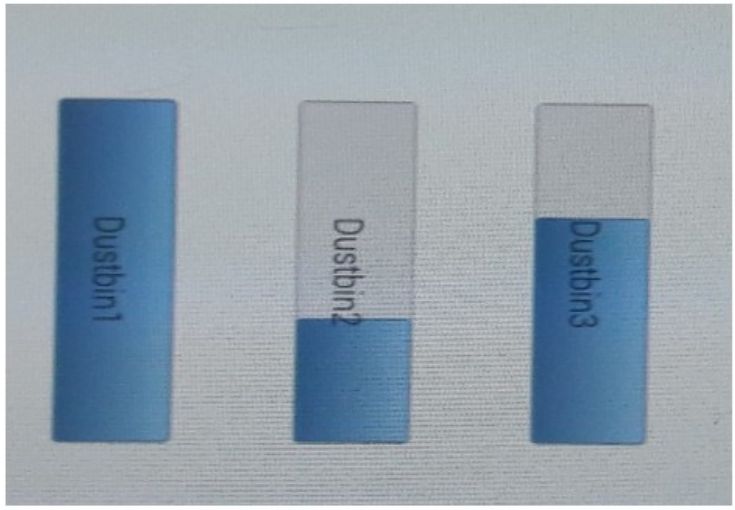

Fig. 14 Raspberry Pi subscribing data from cloud and display on GUI

IX. CONCLUSION

IoT based waste management system using NodeMCU and Raspberry Pi is presented in this paper. Here NodeMCU is work as Publisher mode and Raspberry Pi is working as subscriber mode. Ultrasonic sensors interfaced with NodeMCU measure the depth of the waste level of the individual dustbins. Both IoT modules can communicate with each other via MQTT broker. Raspberry pi subscribe the data from cloud and display it on GUI using PyQt utility. The suggested system is highly responsive as it measure the depth level of waste and display it on GUI runtime. Suggested design can be redesign by replacing NodeMCU with ESP32 (IoT Module) as ESP32 has more no. of analog channels. The suggested system can be extended by updating the depth of waste level in dustbins via email to the specific user.

References

1. G.K.Shyam, S.S.Manvi, P.Bharti "Smart waste management using Internet-of-Things", ICCCT-2017, Feb-2017
2. NodeMcu Datasheet - [online], Available at http://www.handsontec.com/pdf_learn/esp8266-V10.pdf
3. Homepage on Ultrasonic Sensor Available http://centmesh.csc.ncsu.edu/ff_drone_f14_finals/Sensor1/files/hcsr04.pdf
4. Homepage on Raspberry Pi [Online] Available https://www.raspberrypi.org/
5. Homepage on MQTT [online] Available. http://mqtt.org/
6. Homepage on PyQt based GUI using Raspberry Pi Available https://www.baldengineer.com/raspberry-pi-gui-tutorial.html