International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
National Conference "NCSEEE-2016", 19 March 2016
Available online at www.ijrat.org

Design of Smart Dustbin using Wireless Sensor Network and MQTT

Prasad Ostwal[1]
Department of Electronics and Telecommunication[1]
VIIT, Pune

Swapnil Kashid[1]
Department of Electronics and Telecommunication[1]
VIIT, Pune

Harsh Pathak[2]
Department of Computer Science[2]
VIIT, Pune

Anurag Wagh[2]
Department of Computer Science[2]
VIIT, Pune

Abstract- **There are different challenges like littering, inefficient garbage collection, lack of segregation and ambiguity in the existing garbage collection system of the municipality leading to bad hygiene for the citizens. So, to make the garbage collection and disposal system easy and efficient, we have designed IoT (Internet of Things) based solution. The solution comprises of solar powered smart Wi-Fi based electronic hardware and adequate ultrasonic sensors, connected to the cloud server. A whole set-up capsuled in an environment protected enclosure (IP 55 grade) has been mounted on the top of the dustbin to detect the volume of the garbage. A separate ID has been provided for different types of wastes like E-waste or organic waste to achieve the maximum recycling of waste generated in the city. An appropriate user interface has been developed to monitor the micro details regarding the dustbin as well as the Garbage collector vehicles. User Interface has been developed such that he can access the application through web as well as mobile app. This application provides live analytics using different reports like graphs, charts and maps. This data can be used by authorities to analyze and efficiently monitor the scenario of collection system which helps them to plan efficient logistics avoiding dustbin overflow and making the city smarter, cleaner and healthier.**

Keywords:Smart Dustbin, IoT, Sanitation, MQTT, WSN, CouchDB, ESP8266, M2M, Smart City, Node.js

I. INTRODUCTION

Existing Scenario: Solid waste disposal system in the metropolitan cities seems to be quite unorganized. There are a lot of areas having shabby picture of garbage overflowing. The main reason is that there is no transparency and effective communication between logistics management and the municipal authorities. Garbage bins are partially picked up at and due to lack of appropriate segregation, all kind of garbage is dumped on the waste land leading to degradation of land and various health hazards. Also, whenever garbage vehicle is going to collect garbage, they are checking each and every dustbin. If dustbin is empty, it is a waste of fuel and manpower.

We have designed a pilot module which can convert any existing dustbin into a smarter one. In short we are providing a brain to a dustbin by generating periodic data about the dustbin using our intelligent system and ultrasonic sensor. Whenever the dustbin is 70% full, a notification will be sent to the respective authorities to alert them about the current situation of the dustbin. In addition to the amount of garbage it also sends information like fire or explosive, if any. Proper directions are suggested to the garbage vehicles drivers about the optimal routes and collection points on the Android or Google Map applications. If truck driver lacks in using an Android device, the cloud system sends them the pick-up points via SMS. This will save a lot of fuel consumption and manpower. Whenever dustbin is emptied completely, its status will be set to 0% on the database and the garbage pickers will be paid accordingly. The added advantage of this would be that the authorities in their offices could easily monitor if the garbage is timely picked up or not.
The data generated in the cloud database can be used for further analysis. Analytics can be used to study the pattern in which the garbage is filled in various areas and a municipal corporation can plan strategies according to this formed pattern. This generated statistics will be showed in forms of heat maps, pie charts and bar graphs. Also an algorithm to provide strategies for efficient garbage picking will be generated. This statistical data can predict about the garbage generation in various areas in the upcoming months.

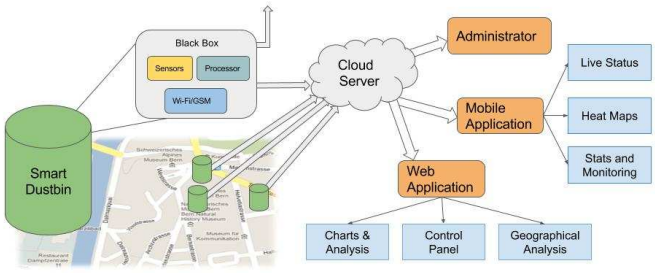

Fig 1. Block Diagram of overall application

II. HARDWARE APPROACH

The Electronics part in the Smart Garbage System comprises of various sensors and processor. Main heart of the circuit is ESP8266 IoT module, which acquires the data from Ultrasonic sensor and smoke sensor. Solar panel is integrated for making the system operative for 24x7 irrespective of absence of AC power..

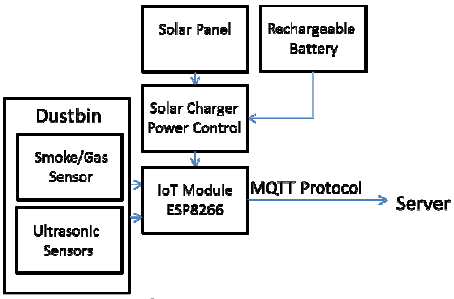

Fig 2. Circuit Outline

A. ESP8266 SoC

ESP8266 is a economical 32 bit Wi-Fi SoC with embedded TCP/IP stack. Some features of this chip include: On chip 4MB ROM, 802.11 b/g/n protocol, GPIO's, ADC, Wi-Fi Direct. This chip has AT command firmware for accessing the terminal. If we use AT firmware it will be required to be commanded by other microcontroller hence to avoid this, we flashed up customized NodeMCU firmware developed for ESP8266. It supports execution of Lua scripts on chip, so whole system is running on ESP's on chip Xtensa processor. This reduces the hardware cost required. NodeMCU firmware comes with lots of API built, one of them is MQTT which we need to communicate to our server.

A. Ultrasonic Sensor

HC-SR04 sensor is used for detecting the level of garbage in the dustbin. Sonar is used for calculating the distance of incident object ranging from 2 to 400 cm.
The sensor is fixed on the top of the dustbin facing downwards. A short triggering pulse of typically 10uS is sent by transmitter. By measuring the pulse width of received signal distance is calculated.

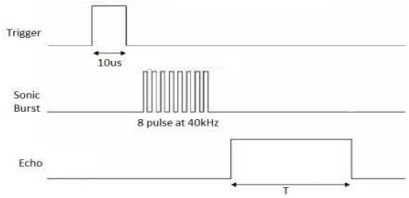

Fig 3. Typical waveforms for HCSR04 Ultrasonic Sensor

B. Solar Power System

As the device is going to be fixed over the dustbins, using AC mains power supply is not adequate; hence solar power has been used. Solar panel provides adequate amount of power, say 12 Volts, 20 Watts for battery charging and overall circuitry. In case of fading of power due to lack of sunlight, higher powered solar cells can be also used to provide uninterrupted services in case of power fading. Still if the power goes off below the desired limits, the server is also informed the same and device gets online once the power is resumed.

III. MESSAGING PROTOCOL

We have used MQTT as our primary protocol to communicate with the server via TCP/IP. MQTT is very simple to use, open source and designed to be simple for implementation. It is ideal for Machine to Machine (M2M) and Internet of Things (IOT). MQTT has small code footprint. The bandwidth required is also low so it can be implemented over GSM network at lowermost cost. MQTT holds a great potential and has lots of features mentioned below. In our project, we have used IBM Bluemix's broker as a cloud computing platform. It has extensive support for MQTT protocol.

A. The publish and subscribe pattern of MQTT:

MQTT is a publish and subscribe protocol where a server is running as a MQTT-Broker. Publisher is the client connected to broker who is originating a message and subscriber is the client which is also connected to broker and is going to receive the message sent by publisher. Publisher and subscriber don't know about the existence of each other. Broker is responsible for filtering of messages and distributing them over subscribers.

B. MQTT Message Filtering:

Broker filters all the messages over network so that a subscriber can subscribe for specific messages only. MQTT basically supports three options:-
1) Subject based filtering

2) Content based filtering
3) Type based filtering

C. MQTT Connection and acknowledgments:

MQTT client can be publisher as well as subscriber concurrently so that it can receive as well as transmit messages.

C.1 Connect:

Client initiates connection to broker by sending a connect message. The message is a MQTT packet containing various fields about client such as client ID, username and password which is sent to broker. Broker opens a connection only if the packet is well constructed. In response to connect message, broker sends out acknowledgement message to the client.

C.2 Publish:

Client acts as a publisher, sends publish message to the broker. Each message should contain topic so that the broker can filter out the message and publish it to the respective clients.

C.3 Subscribe:

Subscribe message is very simple. It contains a unique packed ID and list of topics to be subscribed. Acknowledgement will be sent by broker regarding the subscription.

C.4 Topics:

Topics in MQTT are defined by UTF8 string. Topics are multilevel which are used to structurize the data levels. E.g. in our system we have cities, a city contains localized areas, and areas contain colonies. Now these can be subscribed at various levels for data analysis. A server subscribed to a city gets all the messages from dustbin and can analyze the data with various algorithms.

III. SOFTWARE APPROACH

A. Data Fusion in IoT

IoT devices are connected in clusters, IoT integrates data generated from physical devices and integrate the device into information networks in order to provide advanced intelligence about the clusters of interconnected wireless sensor networks. IoT generates a lot of data from multiple WSN (Wireless Sensors Network) at once. Hence data integration is important since it provides manipulation and management of this data.

The general definition of data fusion [0] is that, it is a formal framework which Contains expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality. The exact definition of greater quality depends on the application. In the IoT environment, data fusion is a framework that comprises theories, methods, and algorithms for inter-operating and integrating multi-source heterogeneous data from sensor measurements or other sources, combining and mining the measurement data from multiple sensors as well as related information obtained from associated databases, and achieving improved accuracy with more specific inferences than that obtained by using only a single sensor.

To achieve Data Fusion, generated data for particular WSN is stored in non-relational database. Engineering database for WSN requires continuous availability and scalability of the data. To achieve availability and partition tolerance, CouchDB is used. CouchDB uses JSON to store data. CouchDB stores data in terms of documents. Each document maintains its own data and self-contained schema. An application may access data from multiple databases such as one's stored on mobile phone and other stored on web server.

B. Node Red

Node-Red is a visual learning tool for wiring IoT devices. It is used to quickly assemble the flow of hardware devices, APIs and online services together. It can be run at the edge of the network or in the cloud. Node-Red is used to map incoming data form MQTT nodes into CouchDB.

Electronics hardware generates the height of the dustbin, transmits the obtained height using MQTT protocol to the cloud. At cloud, Node-Red data is obtained in JSON format and the generated data is mapped into document, and stored into CouchDB.

C. Node.js

Node.js Server has been used to integrate the CouchDB Database to the website on the server. Real time data would get updated in the database after passing the threshold. The updated value is taken as an input in maps present in both Android as well as Nods.js application. API's that were integrated in the web app were Cloudant Express.io., npm(Node Package Manager), body-parser etc. Node.js is intended to operate on a dedicated HTTP server and to function single threaded with only one process at a time. Node.js applications run asynchronously and are events-based.

D. Android

Android devices are popular and widely used. Data generated in CouchDB is replicated at local database on android. This data is represented using pie-chart and histograms.

An android device tracks the garbage vehicles to analyze the frequency of the garbage collection from particular area. According to this analysis, various areas of the cities can be classified based on the amount of waste generated in the city. This classification can be used to focus on the areas which produce garbage in relatively large amount. Tracking of the garbage trucks allows the process of waste disposal in a transparent way.

G. Google Maps

Google developed a specialized web-mapping service called Google Maps. It offers images from satellite, street views, even 360° panoramic views of streets, real-time traffic scenario, and route adaption technique for traveling by foot,

car, or public transportations. Coordinates of every dustbin all over the city are fetched from the cloud. So status of dustbin has been fetched from the cloud and plotted on Heat maps. Color codes used in maps are Red for 80%, green for 40 - 80% and blue for 40% or less filled dustbins. "Get directions" function is used to provide the information about the shortest route for the driver to collect all garbage with great fuel efficiency.

IV. IMPLEMENTATION

We have made a hardware prototype of Smart dustbin. The prototype was tested with few model dustbins. Web application and Android application were also developed and integrated with the hardware and actual dustbin. Figure 4, 5 & 6 show the screenshots of actual application with different types of analytics.

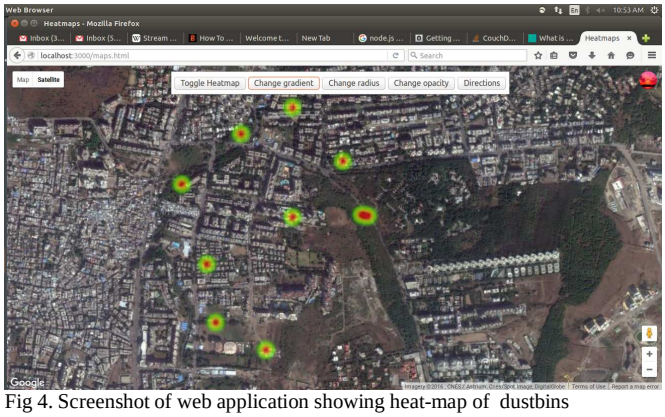

Fig 4. Screenshot of web application showing heat-map of dustbins

Fig 5. Screenshot of web application showing rate of generation of garbage.

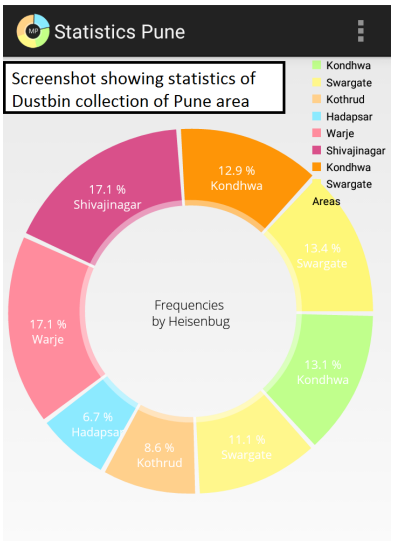

Fig 6. Screenshot of Android application showing statistics of different areas

IV. FUTURE SCOPE

This application can improve the sanitation system of any city. Vast scalability of this application requires assurance of Quality of Service, extensibility and feasibility. The real-time processing of data on platforms like Hadoop, Apache Storm and Apache Kafka can be integrated in future. From hardware point of view, dedicated customized System can be developed which will have all the functionalities of ESP8266, Ultrasonic and smoke sensing etc. which will improve the efficiency, cost and size of hardware. Mesh Network among all Electronic systems can be established to eliminate the need of services like Wi-Max, hence reducing the development cost.

REFERENCES

[1] Y. Xu; S. Helal, "*Scalable Cloud-Sensor Architecture for the Internet of Things*," in IEEE Internet of Things Journal , vol.PP, no.99, pp.1-1

[2] S. Dayu, X. Huaiyu, S. Ruidan and Y. Zhiqiang, "*A GEO-Related IOT Applications Platform Based on Google Map*," e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on, Shanghai, 2010, pp. 380-384.

[3] S. Nastic, S. Sehic, D. H. Le, H. L. Truong and S. Dustdar, "*Provisioning Software-Defined IoT Cloud Systems*," Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, Barcelona, 2014, pp. 288-295.

[4] K. Govindan and A. P. Azad, *"End-to-end service assurance in IoT MQTT-SN*," Consumer Communications and Networking Conference

(CCNC), 2015 12th Annual IEEE, Las Vegas, NV, 2015, pp. 290-296.

[5] A. Narmada and P. S. Rao, "*Zigbee Based WSN with IP Connectivity*," Computational Intelligence, Modelling and Simulation (CIMSiM), 2012 Fourth International Conference on, Kuantan, 2012, pp. 178-181.

[6] S. Nastic, S. Sehic, M. Vogler, H. L. Truong and S. Dustdar, "*PatRICIA -- A Novel Programming Model for IoT Applications on Cloud Platforms,*" Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on, Koloa, HI, 2013, pp. 53-60.

[7] M. S. Aslam, S. Rea and D. Pesch, "*Service Provisioning for the WSN Cloud*," Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, Honolulu, HI, 2012, pp. 962-969.