International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

# EFFECTIVE IOT TECHNIQUES TO MONITOR THE LEVELS OF GARBAGE IN SMART DUSTBINS

**Santhosh Voruganti[1], Karnati Ramyakrishna[2]**

[1]*Assistant Professor, Department of IT, CBIT, Hyderabad, Telangana, India*
[2]*Department of MCA, Osmania University, Hyderabad, Telangana, India*

---

**Abstract -***Today main issue for pollution is Garbage Overflow. It creates unhygienic condition for the people and creates bad smell around the surroundings this leads in spreading some deadly diseases and human illness. To avoid all such situations we are going to implement IoT Based waste management using smart dustbin. Implementation is done with the help of IoT concept. The Internet of Things (IoT) is a concept in which surrounding objects are connected through wireless networks without user intervention. Objects communicate and exchange information. In this system multiple dustbins are located throughout the city or the Campus, these dustbins are provided with a sensor which helps in tracking the level and weight of the garbage bins and a unique location will be located using gps for every dustbin in the city so that it is easy to identify which garbage bin is full. When the level and weight of the bin reaches the threshold limit, the device will transmit the reading along with the unique location provided by GPS. The status of the bin is accessed by the concerned authorities from their place with the help of Internet and an immediate action will be taken to replace overflowing bins with the empty bins and an SMS will be send to Authority While this is exceeding the limit with GSM.*

***KeyWords***: Gsm, Gps, Modem, IOT, WeightS ensor, Humidity Sensor, Level Sensor.

## 1. INTRODUCTION

As the world's population grows at a fast pace, more and more waste is produced daily and waste management becomes a more crucial matter. Of particular importance is the collection of solid waste from city garbage bins. Research has shown that solid waste collection and transfer provided by a city to its residents takes up over 70% of the city waste management budget in developing countries, and up to 60% in developed countries. This not only depletes the council of its budget in a single area, but also reduces the resources that can be spent in other aspects of waste management, such as recycling plants and the like. In addition to the resources used, it has also been shown that inadequate or inefficient collection processes also lead to undesirable and in some cases unsanitary conditions that pose a risk to the surrounding communities. Such risks are presented in the form of overfilled garbage bins and foul odour. It was identified that the large number of resources used is generally due to the lack of planning, data on the collection, and poor infrastructure. With the current advancement of technology, smart cities are on the rise. Smart cities represent a conceptual urban development model on the basis of the utilization of human, collective, and technological capital for the development of urban agglomerations. With the popularity of the Internet of things (IoT) growing, and the availability of low cost actuators and sensors, the benefits of these technologies can be used to solve the problems faced in the current methods of waste management in cities. To this effect, this paper proposes a system that will address the issues described above to provide a more efficient solution.

### 1.1 Overview

An embedded system is a special-purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls. Unlike a general purpose computer, such as a personal computer, an embedded system performs one or a few predefined tasks, usually with very specific requirements. Since the system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product. Embedded systems are often mass-produced, benefiting from economies of scale. Personal digital assistants (PDAs) or hand-held computers are generally considered embedded devices because of the nature of their hardware design, even though they are more expandable in software terms. This line of definition continues to blur as devices expand. With the introduction of the OQO Model with the Windows XP operating system and ports such as a USB port both features usually belong to "general purpose computers", the line of nomenclature blurs even more Embedded systems plays major role in electronics varies from portable devices to large stationary installations like digital watches and MP3 players, traffic lights, factory controllers, or the systems controlling nuclear power plants. In terms of complexity embedded systems can range from very simple with a single micro-controller chip, to very complex with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

### 1.2 Significance

Due to their compact dimension, low rate and easy design features made embedded programs very general and

encroached into human lives and have grow to be integral. They are discovered every the place from kitchen ware to house craft. To stress this inspiration here are some illustrations.

### 1.2 Inside of an Embedded System

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU).This hardware additionally comprises reminiscence chips onto which the software is loaded. The software residing on the reminiscence chip is also called the 'firmware'. The operating process runs above the hardware, and the application software runs above the operating system. The equal structure is relevant to any pc including a computer. However, there are massive differences. It isn't obligatory to have an operating process in every embedded for small home equipment such as remote manipulate units, air-conditioners, toys and many others., there is not any need for an working process and we are able to write best the program specific to that utility. For applications involving tricky processing, that you must have an operating process. In any such case, you need to combine the appliance software with the working method after which transfer the whole software on to the memory chip. As soon as the application is transferred to the memory chip, the program will proceed to run for a long time and you don't have to reload new application.

### 1.3 Features of an Embedded System

Embedded systems do a very specific task, they cannot be programmed to do different things. Embedded systems have very restricted assets, certainly the reminiscence. In general, they don't have secondary storage contraptions such because the CDROM or the floppy disk.

Embedded systems must work in opposition to some time limits. A distinctive job needs to be accomplished inside a distinct time. In some embedded systems, called real-time techniques, the points in time are stringent. Missing a useless line may just reason a disaster – loss of existence or injury to property. Embedded systems are constrained for power, As many embedded systems operate through a battery, the power consumption has to be very low. Embedded systems need to be highly reliable. Once in a while, pressing ALT-CTRL-DEL is OK on your desktop, but you cannot afford to reset your embedded system. Some embedded techniques must operate in severe environmental stipulations such as very excessive temperatures and humidity. Embedded programs do a very detailed task they cannot be programmed to do exclusive matters. Embedded programs that handle the patron market (for illustration digital toys) are very rate-potent. Even a discount of Rs.10 is lot of price saving, considering that countless numbers or thousands systems could also be sold. Not like computing device desktops where the hardware platform is dominated through Intel and the working method is dominated by Microsoft, there's a large kind of processors and operating programs for the embedded programs. So, making a choice on the correct platform is essentially the most intricate task.

## 2. LITERATURE SURVEY

In our city many times we see that the garbage bins or dustbins placed at public places are overflowing. It creates unhygienic conditions for people. Also it creates ugliness to that place. At the same time bad smell is also spread. To avoid all such situations we are going to implement a project called Garbage collection bin overflow indicator using IOT (Internet of Things) technology. In this project we are going to place a sensor (Infrared sensor / proximity sensor) under the dustbin. When the sensor signal reaches to the threshold value, a mail notification (like email, twitter, whatsapp message) will be sent to the respective Municipal / Government authority person. We can also see the density of the Dustbin through internet on a Dashboard, this is a GUI (Graphical User Interface) dashboard so any of the authenticate person will easy check the present condition of the dustbin. So then that person can send the collection vehicle to collect the full garbage bins or dustbins.

This implementation of Smart Garbage collection bin using IoT, IR sensor, micro-controller and GUI . This system assures the cleaning of dustbins soon when the garbage level reaches its maximum. If the dustbin is not cleaned in specific time, then the record is sent to the higher authority who can take appropriate action against the concerned contractor. This system also helps to monitor the fake reports and hence can reduce the corruption in the overall management system. This reduces the total number of trips of garbage collection vehicle and hence reduces the overall expenditure associated with the garbage collection. It ultimate helps to keep cleanness in the society. Therefore, the smart garbage management system makes the garbage collection more efficient the use of solar panels in such systems may reduce the energy consumption.

With increase in population, the scenario of cleanliness with respect to garbage management is degrading tremendously. The overflow of garbage in public areas creates the unhygienic condition in the nearby surrounding. It may provoke several serious diseases amongst the nearby people. It also degrades the valuation of the area. To avoid this and to enhance the cleaning, 'smart garbage management system' is proposed in this paper. In the proposed system, the level of garbage in the dustbins is detected with the help of Sensor systems, and communicated to the authorized control room through GSM system. Micro-controller is used to interface the sensor system with GSM system. A GUI is also developed

to monitor the desired information related to the garbage for different selected locations. This will help to manage the garbage collection efficiently.

Implementation of smart garbage management system using IR sensor, micro-controller and GSM module. This system assures the cleaning of dustbins soon when the garbage level reaches its maximum. If the dustbin is not cleaned in specific time, then the record is sent to the higher authority who can take appropriate action against the concerned contractor. This system also helps to monitor the fake reports and hence can reduce the corruption in the overall management system. This reduce the total number of trips of garbage collection vehicle and hence reduce the overall expenditure associated with the garbage collection. It ultimate helps to keep cleanness in the society. Therefore, the smart garbage management system makes the garbage collection more efficient The use of solar panels in such systems may reduce the energy consumption. Such systems are vulnerable to plundering of components in the system in different ways which needs to be worked on.

Smart garbage management system monitors the garbage overflow in garbage bin. The level of garbage in the dustbin is detected with the help of sensor systems, and communicated to the authorized person through GSM system. Infrared sensor (IR sensor) is used to detect the level of garbage. Micro-controller is used to interface the sensor system with the GSM system. An RFID is used to monitor the desired information related to the garbage for different selected locations. It also monitors the attendance of the authority person. With the use of conveyor belt and PH sensor the degradable and non-degradable waste can be separated. Thus the collection and separation of garbage is monitored efficiently.

## 3. SYSTEM DESIGN

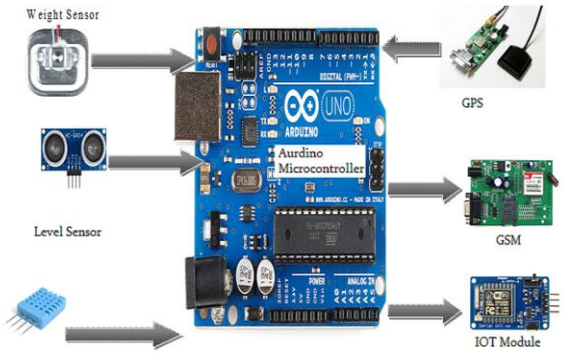


**Fig -1**: System Architecture

### 3.1 Modules

- Weight Sensor
- Humidity Sensor
- Level Sensor
- GPS
- GSM
- IOT Module

## 4. IMPLEMENTATION

**Fig -2**: Arduino Uno R3 Front

The Arduino Uno is a micro-controller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the micro-controller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to serial converter.

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

### 4.1 Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the micro-controller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a boot-loader that allows you to upload new code to it without the use of an external hardware programmer. It

communicates using the original STK500 protocol (reference, C header files).

You can also bypass the boot-loader and program the micro-controller through the ICSP (In-circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot-loader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

### 4.2 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot-loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following halfsecond or so, the boot-loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

### 4.3 Usb Overcurrent Protections

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

### 4.4 Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

### 4.5 Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other micro-controllers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

### 4.6 Memory

The ATmega328 has 32 KB (with 0.5 KB used for the boot-loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

## 5. RESULTS

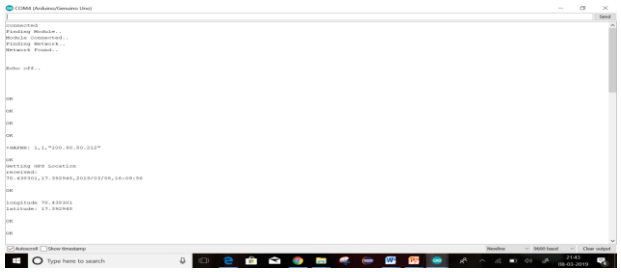

**Fig -3:** Output screen which shows location of the dustbin

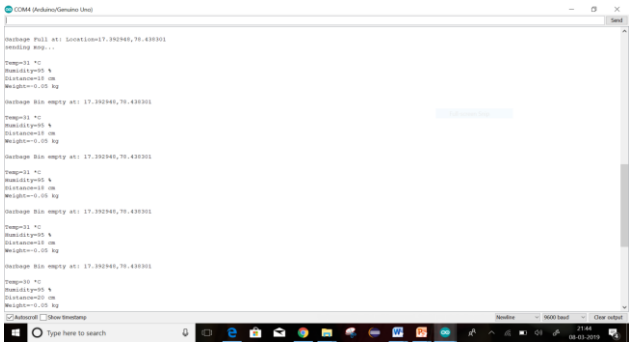

**Fig -4:** Output screen which shows the status of the bin


**Fig -5:** Snapshot of the web page

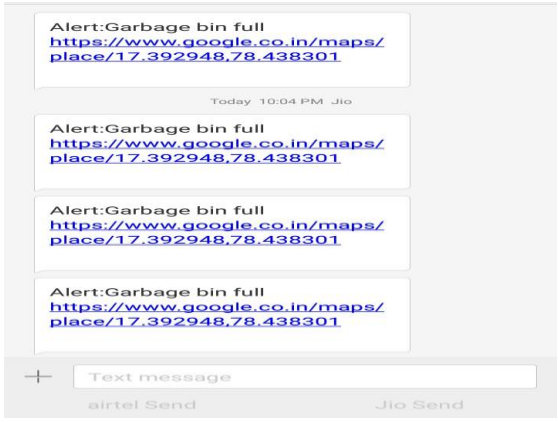


**Fig -6:** Alert message of garbage overflow

## 5.1 Applications versus Embedded Testing

Embedded systems software testing shares much in common with application software testing. Thus, much of this two part article is a summary of basic testing concepts and terminology. However, some important differences exist between application testing and embedded systems testing. Embedded developers often have access to hardware-based test tools that are generally not used in application development. Also, embedded systems often have unique characteristics that should be reflected in the test plan. These differences tend to give embedded systems testing its own distinctive flavor. This article covers the basics of testing and test case development and points out details unique to embedded systems work along the way.

## 5.2 Why Test?

Before you begin designing tests, it's important to have a clear understanding of why you are testing. This understanding influences which tests you stress and (more importantly) how early you begin testing. In general, you test for four reasons:

- To find bugs in software (testing is the only way to do this)
- To reduce risk to both users and the company
- To reduce development and maintenance costs
- To improve performance

***To Find the Bugs.*** One of the earliest important results from theoretical computer science is a proof (known as the Halting Theorem) that it's impossible to prove that an arbitrary program is correct.

***To Reduce Costs.*** The classic argument for testing comes from *Quality Wars*by Jeremy Main. In 1990, HP sampled the cost of errors in software development during the year. The answer, $400 million, shocked HP into a completely new effort to eliminate mistakes in writing software.
The $400M waste, half of it spent in the labs on rework and half in the field to fix the mistakes that escaped from the labs, amounted to one-third of the company's total R&D budget and could have increased earnings by almost 67%.

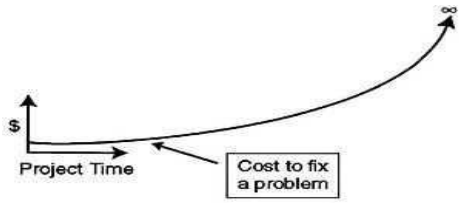


**Fig -7:** Unit Testing

The Cost to Fix a Problem. Simplified graph showing the cost to fix a problem as a function of the time in the product life cycle when the defect is found. The costs associated with finding and fixing the Y2K problem in embedded systems is a close approximation to an infinite cost model.

## 6. CONCLUSION AND FUTURE SCOPE

This Paper shows the implementation of smart garbage management system using Weight, level sensor, micro-controller and GSM, GPS module. This system assures the cleaning of dustbins soon when the garbage level reaches its maximum. If the dustbin is not cleaned in specific time,

then the record is sent to the higher authority who can take appropriate action against the concerned contractor. We can keep it in real time for GHMC using for "swachh bharath".

If the sensors get damaged then accurate and exact results can't be achieved which becomes a problem .So, sensors must be taken care. During the implementation of weight sensor, if the sensor is placed in the opposite direction then negative values are obtained.

In future it can be used as a benchmark by the people who are willing to take one step further for increasing the cleanliness in their respected areas. Various other types of sensors can be used with the ultrasonic sensor to germ or precise output and to take this system to another level.

## 7. REFERENCES

1. V. N. Bhat, "A model for the optimal allocation of trucks for solid waste management", vol. 14, pp. 87-96 , 1996.
2. L. L. Abarca-Guerrero et al, "Solid waste management challenges for cities in developing countries", vol. 33, pp. 220-232, 2013.
3. Microtronics Technologies, "GSM based garbage and waste collection bins overflow indicator", September 2013.
4. Hindustan Embedded System, "City Garbage collection indicator using RF (ZigBee) and GSM technology".
5. Z embedded, "GSM modem interfacing with 8051 for SMS" August 2012.
6. Marian Look, "Trash Plant: India", earth911B.
7. P. H. Brunner and J. Fellner, "Setting priorities for waste management strategies in developing countries", vol. 25, pp. 234-240, 2007.
8. M. Sharholy et al, "Municipal solid waste management in Indian cities – A review", vol. 28, pp. 459-467, 2008.
9. Basic Feature, "Solid waste Management Project by MCGM".
10. M. Angelidou, "Smart city policies: A spatial approach", vol. 41, pp. S3-S11, 2014.
11. K. Ravindra, K. Kaur and S. Mor, "System analysis of municipal solid waste management in Chandigarh and minimization practices for cleaner emissions", vol. 89, pp. 251-256, 2015.
12. S. K. Amponsah and S. Salhi, "The investigation of a class of capacitated arc routing problems: the collection of garbage in developing countries", vol. 24, pp. 711-721, 2004.
13. Clark, R. M. & Gillean, J. I, "Analysis of solid waste management operation in Cleveland", vol.6, pp. 32-42, 2005.
14. Kasliwal Manasi H, "A Novel approach to Garbage Management using IOT for smart cities", vol.2, pp. 348-353, 5,May 2016.
15. Wilson, E.J, "Life cycle inventory for municipal solid waste management", vol.20, pp. 16-22, 2002.
16. Muhammed ali Mazdi & Janice Gillispie "8051 Micro-controller & embedded system" 3rd edition pearson.
17. Jayavardhana Gubbi,a Rajkumar Buyya,b Slaven Marusic,a Marimuthu Palaniswamia, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions,", Elvisier, 2013.
18. L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787 – 2805, 2010.
19. D. Giusto, A. Iera, G. Morabito, and L. Atzori, "The Internet of Things. Springer-Verlag," 2010.
20. Dave Evans, Cisco, "The Internet of Things, How the Next Evolution of the Internet Is Changing Everything," IBSG, 2011.
21. Santhosh Voruganti,Map Reduce A Programming Model For Cloud Computing Based On Hadoop Ecosystem,International Journal of Computer Science and Information Technologies, Vol. 5 (3),2014,3794-3799.
22. Santhosh Voruganti, Survey on Data intensive Applications, Tools and Techniques for Mining Unstructured Data International Journal of Computer Applications (0975 – 8887) Volume 146 – No.12, July 2016.
23. Santhosh Voruganti Local Security Enhancement and Intrusion Prevention in Android Devices International Research Journal of Engineering and Technology (IRJET)Volume: 07 Issue: 01 | Jan 2020e-ISSN: 2395-0056 p-ISSN: 2395-0072.
24. Santhosh Voruganti, Comparative Analysis of Dimensionality Reduction Techniques for Machine Learning", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 4 Issue 8, pp. 364-369, May-June 2018.
25. Santhosh Voruganti, ENHANCED RATING PREDICTION BASED ON LOCATION AND FRIEND SET", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.6, Issue 5, page no.384-388, May-2019.