**INTRODUCTION**

While doing the research work and interviewing the common people and the government authorities associated with the garbage management (Municipal corporations) of various places, few very common things turned up: A nation always possess rules, regulations and technologies but the matter of grave concern is that the linking factor is missing, faithful following of duties by the officers and low grade workers is nowhere to be seen. Here arises a urgent need of developing a system which can handle the situation intelligently before it’s too late

The invention named “Garbage Monitoring and Management System using Sensors, Arduino-Uno and Wi-Fi-module” is very much related to society welfare and health. It is a scientific and faithful approach to achieve a garbage free, disease- free and healthy locality by providing the local government with a system which uses number of basic technologies at reasonable price.

A proper waste management system is required to keep the city clean and hygienic. These dustbins are connected with micro controller, Ultrasonic Sensors and Wi-Fi-modules where the Ultrasonic sensor will detect the level of the dustbin and will send the signals to micro controller. The data received will be analyzed and processed and accordingly the dustbin level can be found out on weekly basis.

Authorized person will have the access of Garbage Monitoring and Management system which will show the current level of dustbin. This will help in regularly monitoring the current status of dustbin and clean the dustbins at right times so unnecessary bad smell will be reduced.

Nowadays, there are tons of flats and apartments which have been built in the rapid urbanization area. This is due to high housing demands which have been drastically risen as a result of migration from villages to cities to find works. In order to accommodate the growing population in the urban area, the government has built flats, apartments or condominiums, to provide shelter for them. Besides, flats become popular mainly in the United State, Europe and other developing countries

The waste disposal can be managed more properly and efficiently by constantly monitoring the bin status and the garbage level. In addition, the municipality can be alerted when the bin is full or almost full, thus promoting dynamic scheduling and routing of the garbage collection.

By comparing to the conventional static scheduling and routing, this dynamic scheduling and routing are said to allow operational cost reduction, by reducing the number of trucks, the manual labor cost and the transport mileage savings.

**2 LITERATURE REVIEW**

**2.1 Existing system**

During research, we found that in India there are so many problems in garbage management system. Public waste bins are filling up faster than ever and inevitably many of the bins end up overflowing before collected, causing not only cluttered streets and bad odors but also negative health and environmental impacts.

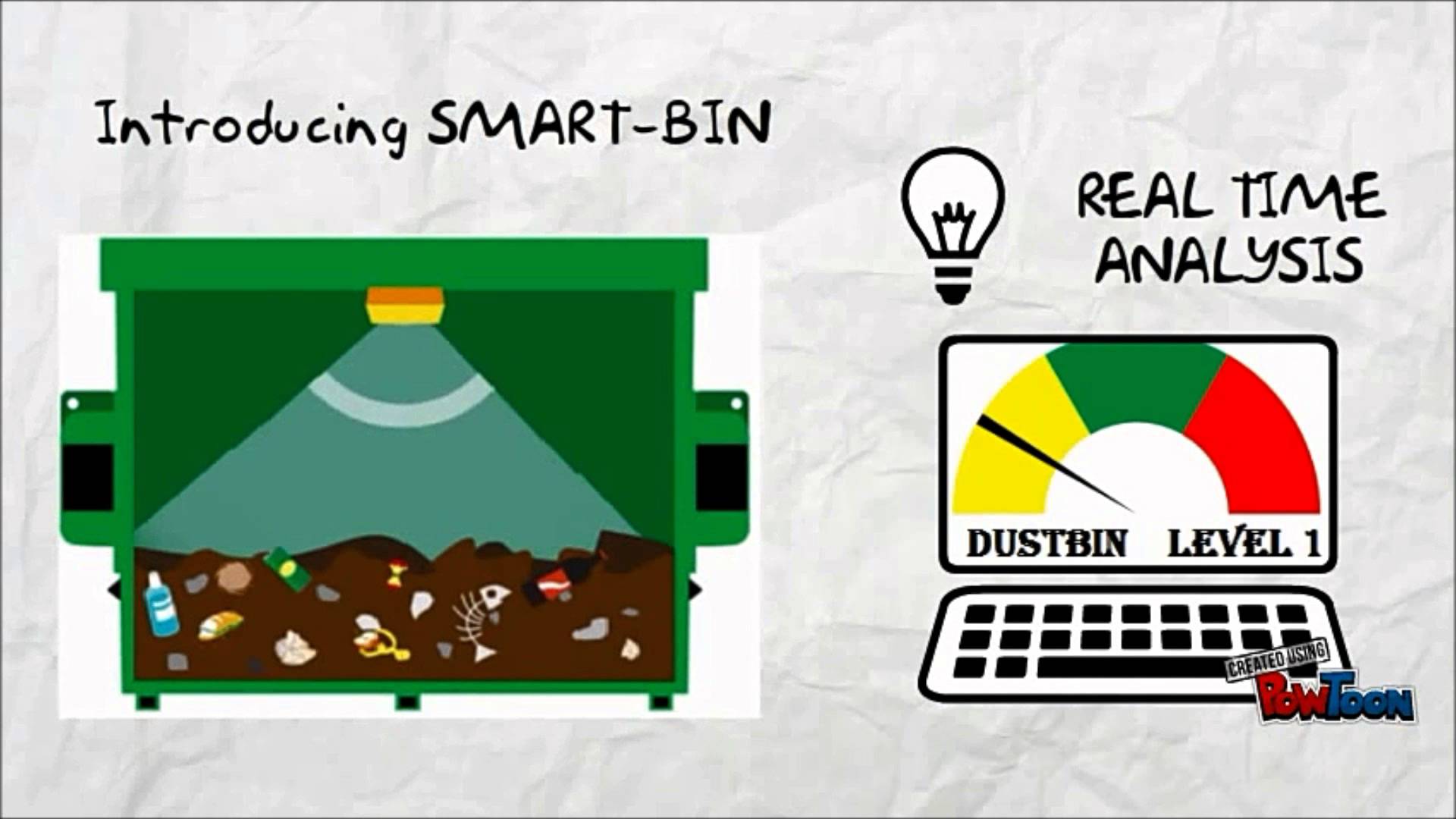
* Storage of waste at source is lacking.
* Domestic waste/trade waste is thrown on streets.
* Segregation of recyclable waste at source is not done.
* Some cities the careless attitudes of local bodies towards maintaining cleanliness have caused deadly results.
* Primary collection of waste not done at place of generation.



We also research on garbage separation on street. In some areas garbage is get separate in two sections. I.e. Wet and Dry**.** It helps to recycle that garbage and again we can use that. But in some rural areas it is not done in proper way. Both Dry and Wet garbage is mixed together which is very difficult to recycle. It is also very unhygienic.

**2.2 Proposed system**

So to overcome this all problems we design a new system. We invent a smart dust bin which is very useful for all cities. This is used to form the integrated system to monitor the waste bins remotely. The sensors are placed in the common garbage bins placed at the public places. When the garbage reaches the level of the sensor, then that indication will be given to Arduino Uno Controller. The controller will glow the LED’s on bins and also display on webpage which garbage bin is completely filled and needs urgent attention.



In the proposed system describes that the level of garbage in the dustbins is detected with the help of Sensor systems, and communicated to the authorized control room. It describes the application of our model of “Smart Bin” in managing the waste collection system of an entire city.

**3.1 Hardware Specifications**

**1. Arduino Uno**

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet).

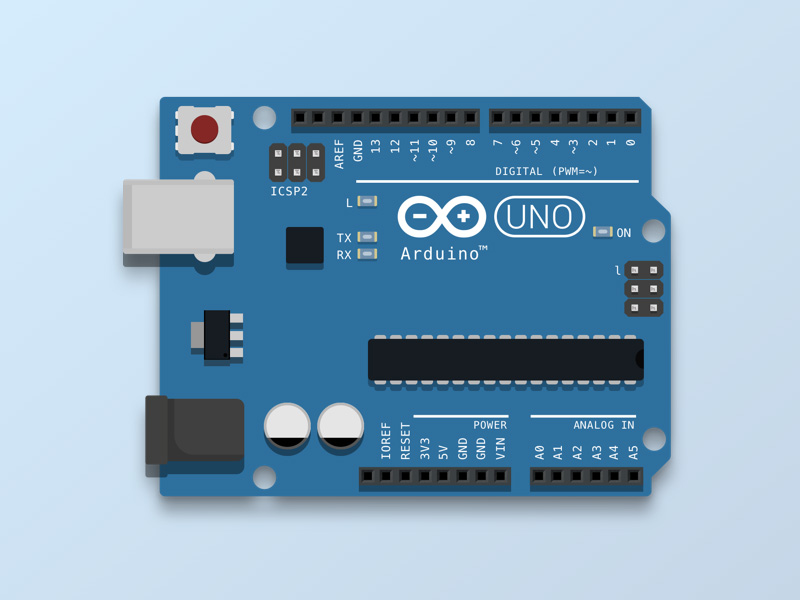


Figure3.1.1: - Arduino Uno

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

**2. Wi-Fi Modem**

This module has a powerful enough onboard processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development upfront and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts. There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support.



Figure3.1.2: - WI-FI Module

In the Documents section below, you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IOT (Internet of Things) solution!

**3. LED’s**

A **light-emitting diode** (**LED**) is a two-[lead](https://en.wikipedia.org/wiki/Lead_(electronics)) [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) [light source](https://en.wikipedia.org/wiki/Light_source). It is a [p–n junction](https://en.wikipedia.org/wiki/P%E2%80%93n_junction) [diode](https://en.wikipedia.org/wiki/Diode) that emits light when activated. When a suitable [current](https://en.wikipedia.org/wiki/Electric_current) is applied to the leads, [electrons](https://en.wikipedia.org/wiki/Electron) are able to recombine with [electron holes](https://en.wikipedia.org/wiki/Electron_hole) within the device, releasing energy in the form of [photons](https://en.wikipedia.org/wiki/Photon). This effect is called [electroluminescence](https://en.wikipedia.org/wiki/Electroluminescence), and the color of the light (corresponding to the energy of the photon) is determined by the energy [band gap](https://en.wikipedia.org/wiki/Band_gap) of the semiconductor. LEDs are typically small (less than 1 mm2) and integrated optical components may be used to shape the [radiation pattern](https://en.wikipedia.org/wiki/Radiation_pattern)

Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity infrared light. Infrared LEDs are still frequently used as transmitting elements in remote-control circuits, such as those in remote controls for a wide variety of consumer electronics. The first visible-light LEDs were of low intensity and limited to red. Modern LEDs are available across the [visible](https://en.wikipedia.org/wiki/Visible_spectrum), [ultraviolet](https://en.wikipedia.org/wiki/Ultraviolet), and [infrared](https://en.wikipedia.org/wiki/Infrared) wavelengths, with very high brightness.

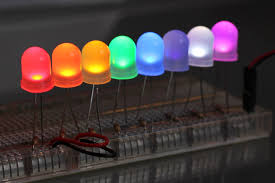


Figure3.1.3: -LED’S

Early LEDs were often used as indicator lamps for electronic devices, replacing small incandescent bulbs. They were soon packaged into numeric readouts in the form of seven-segment displays and were commonly seen in digital clocks. Recent developments have produced LEDs suitable for environmental and task lighting. LEDs have led to new displays and sensors, while their high switching rates are useful in advanced communications technology.

**4. Ultrasonic sensors**

The Ultrasonic Sensor sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect back. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, (like a tiny speaker), the other receives them, (like a tiny microphone). The speed of sound is approximately 341 meters (1100 feet) per second in air.



Figure3.1.4: - Ultrasonic Sensor

The ultrasonic sensor uses this information along with the time difference between sending and receiving the sound pulse to determine the distance to an object.

Technical Specifications: -

• Power Supply − +5V DC

• Quiescent Current

• Working Current − 15mA

• Effectual Angle − <15°

• Ranging Distance − 2cm – 400 cm/1″ – 13ft

• Resolution − 0.3 cm

• Measuring Angle − 30 degree

**5. Moisture Sensor: -**

The soil moisture sensor consists of two probes which are used to measure the Volumetric content of water. The two probes allow the current to pass through the soil and then it gets the resistance value to measure the moisture value.

When there is water, the soil will conduct more electricity which means that there will be less resistance. Therefore, the moisture level will be higher. Dry soil conducts electricity poorly, so when there is less water, then the soil will conduct less electricity which means that there will be more resistance. Therefore, the moisture level will be lower.

This sensor can be connected in two modes; Analog mode and digital mode. First, we will connect it in Analog mode and then we will use it in Digital mode.

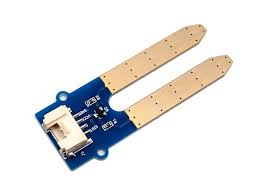


Figure3.1.5: - Moisture Sensor

Specifications

The specifications of the FC-28 soil moisture sensor are as follows:

* Input Voltage: 3.3 – 5V
* Output Voltage: 0 – 4.2V
* Input Current: 35mA
* Output Signal: Both Analog and Digital

**3.2 Software Specifications**

1. Arduino Compiler

A program for Arduino may be written in any [programming language](https://en.wikipedia.org/wiki/Programming_language) with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE), which is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It originated from the IDE for the languages [*Processing*](https://en.wikipedia.org/wiki/Processing_(programming_language)) and [*Wiring*](https://en.wikipedia.org/wiki/Wiring_(development_platform)). It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, [brace matching](https://en.wikipedia.org/wiki/Brace_matching), and [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension Arduino Software (IDE) pre-1.0 saved sketches with the extension

2. MC Programming Language: C

**Embedded C** is a set of language extensions for the [C programming language](https://en.wikipedia.org/wiki/C_(programming_language)) by the [C Standards Committee](https://en.wikipedia.org/wiki/ISO/IEC_JTC_1/SC_22) to address commonality issues that exist between C extensions for different [embedded systems](https://en.wikipedia.org/wiki/Embedded_system). Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as [fixed-point arithmetic](https://en.wikipedia.org/wiki/Fixed-point_arithmetic), multiple distinct [memory banks](https://en.wikipedia.org/wiki/Memory_bank), and basic [I/O](https://en.wikipedia.org/wiki/Input/output) operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.

Embedded C uses most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

**4.1 System Architecture**

The IOT Garbage Monitoring system is a very innovative system which will help to keep the cities clean. This system monitors the garbage bins and informs whether the Dustbins in full or empty via a web page. For this the system uses ultrasonic sensors placed over the bins to detect the garbage level and compare it with the garbage bins depth. The system makes use of Arduino Uno, Wi-Fi modem for sending data and a buzzer. Whereas a web page is built to show the status to the user monitoring it. The system puts on the buzzer when the level of garbage is almost to get full. Thus, this system helps to keep the city clean by informing about the garbage levels of the bins by providing the output on the Webpage.

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes preprogramed with an AT command set firmware. The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

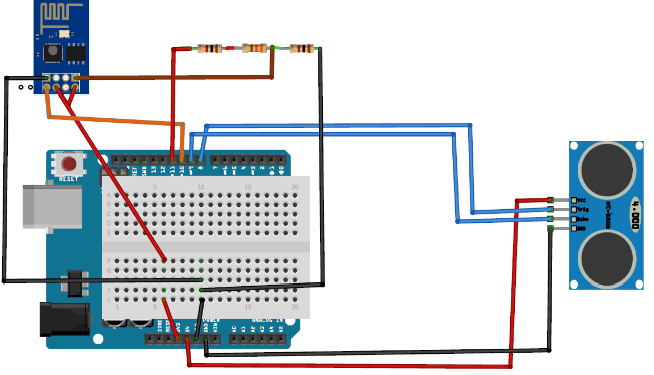


Figure 4.1:- System Architecture

**Explanation: -**

First of all, we will connect the ESP8266 with the Arduino. ESP8266 runs on 3.3V and if you will give it 5V from the Arduino then it won’t work properly and it may get damage. Connect the VCC and the CH\_PD to the 3.3V pin of Arduino. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it. Three 1k resistors connected in series will do the work for us. Connect the RX to the pin 11 of the Arduino through the resistors as shown in the figure below and also the TX of the Arduino to the pin 10 of the Arduino.

Now it’s time to connect the HC-SR04 ultrasonic sensor with the Arduino. Connections of the ultrasonic sensor with the Arduino are very simple. Connect the VCC and the ground of the ultrasonic sensor to the 5V and the ground of the Arduino. Then connect the TRIG and ECHO pin of ultrasonic sensor to the pin 8 and 9 of the Arduino respectively.

**4.2 Flow Chart: -**

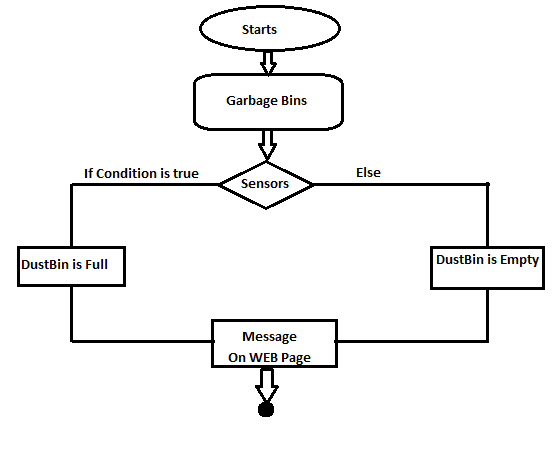


Figure 4.2:- Flow Chart.

**5. Project Management**

**5.1 Development Methodologies**

**What Are Process Models and development methodology?**

A process model guides the order of project activities and represents the life cycle of a project. Historically, some process models are static and others do not allow checkpoints. Two such process models are the waterfall model and the spiral model.

These models provide two different approaches to the project life cycle. The preceding illustration shows the waterfall model’s cascading checkpoints and the spiral model’s circular approach to process.

**5.1.1 Waterfall model.**

This model uses milestones as transition and assessment points. When using the waterfall model, you need to complete each set of tasks in one phase before moving on to the next phase. The waterfall model works best for projects in which the project requirements can be clearly defined and are not liable to modifications in the future. Because this model has fixed transition points between phases, you can easily monitor schedules and assign clear responsibilities and accountability.



Fig5.1.1: Waterfall Model

**5.1.2 Spiral model.**

This model is based on the continual need to refine the requirements and estimates for a project. The spiral model is effective when used for rapid application development of very small projects. This approach can generate great synergy between the development team and the Member because the Member is involved in all stages by providing feedback and approval. However, the spiral model does not incorporate clear checkpoints. Consequently, the development process might become chaotic.

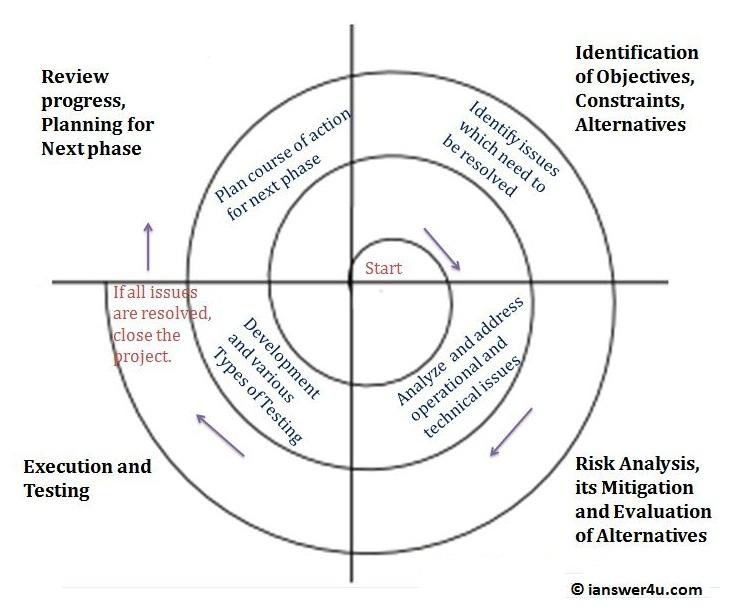


Fig5.1.2: Spiral model

**5.1.3 MSF Process Model:**

The MSF – Microsoft Solution Framework Process Model combines the best principles of the waterfall and spiral models. It combines the waterfall model’s milestone-based planning and resulting predictability with the spiral model’s benefits of feedback and creativity.

MSF process model is divided to repetitive cycles. Each cycle divided to five phases: envisioning, planning, development, stabilizing and deploying.

Fig: 5.1.3: MSF Process Model

In garbage monitoring system process we have used MSF Process Model. This is a combination of spiral and waterfall process which is a very easy to use.

**5.2 Project Development Life Cycle (PDLC)**

The project will completed by this four distinct phases as per MSF Process Model.

**5.2.1 Envisioning**

Envisioning is gathering the requirement of the project from different sources. Some techniques for gathering information are interviewing, shadowing, user instructions, and prototyping. Creating and identify the project scope. The scope of the project specifies what will and will not be included in the project. In this phase we have created the scope document. It includes information about the team and project structure, the problem statement, the vision statement, the scope of the project, the solution concept, user profiles, and project goals.

**5.2.2 Planning**

The planning phase results in the architecture and design of the solution, the plans to accomplish the development and deployment of the solution, and the schedules associated with tasks and resources. There are three design processes in the planning phase: conceptual, logical, and physical design.

**5.2.3 Developing and Designing the System**

Design of any application is not complete without a way for users to interact with the system. User interaction takes place through the application’s presentation layer. The presentation layer is the part of the application that provides a communication mechanism between the user and the business service layer of the system. The simplest presentation layers contain user interface components, such as Windows Forms or ASP.NET Web Forms. For more complex user interactions, you can design user process components to orchestrate the user interface elements and control the user interaction. User interface components display data to users, acquire and validate data from user input, and interpret user gestures that indicate the user wants to perform an operation on the data. Additionally, the user interface should filter the available actions to let users perform only the operations that are appropriate at a certain point in time.

**5.2.4 Stabilizing**

The purpose of the stabilizing phase is to reduce the risks of releasing the solution to production. A successful stabilizing phase requires that the team make the transition from a mindset focused on building features to one focused on getting the solution to a known state of quality. Deliverables of the deploying phase are operations and support information systems, repository of all versions of documentation and code, and project closeout reports.

**5.3 SOFTWARE DEVOLOPMENT LIFE CYCLE**

**5.3.1** **INCREMENTAL MODEL**

The incremental build model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished when it satisfies all of this requirements. In incremental model the whole requirement is divide into various builds. Multiple development cycle take place here, making a life cycle “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Each model passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

****

**5.3.1 Advantages of incremental model**

* Generates working software quickly and early during the software lifecycle.
* This model is more flexible-less costly to change scope and requirement.
* It is easier to test and debug during a smaller in iteration. This model customer can respond to each built. Lower initial delivery cost.
* Easier to manage risk because risky pieces are identified and handled it iteration

**6. DIAGRAMS**

**6.1 DFD DIAGRAMS**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the [visualization](https://en.wikipedia.org/wiki/Data_visualization) of [data processing](https://en.wikipedia.org/wiki/Data_processing) (structured design)

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured [flowchart](https://en.wikipedia.org/wiki/Flowchart) which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model

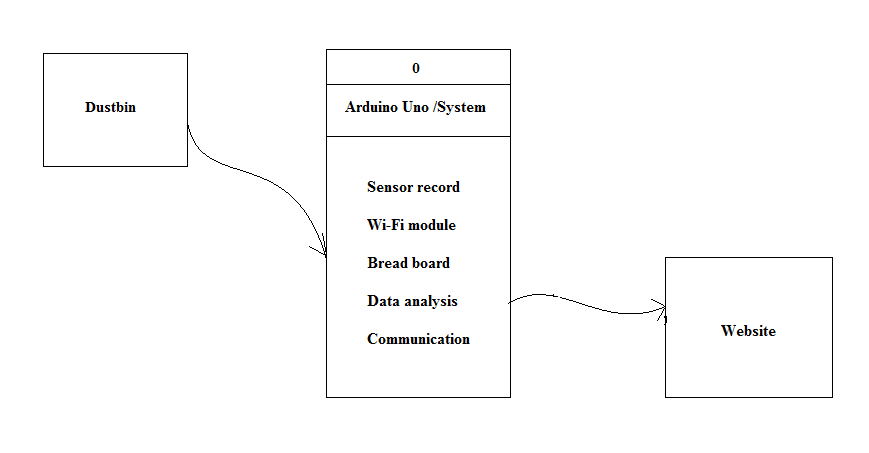
**6.1.1 Level 0 Data Flow Diagram:**

Figure 6.1.1 DFD 1 Diagram

**6.1.2 Level 1 Data Flow Diagram**

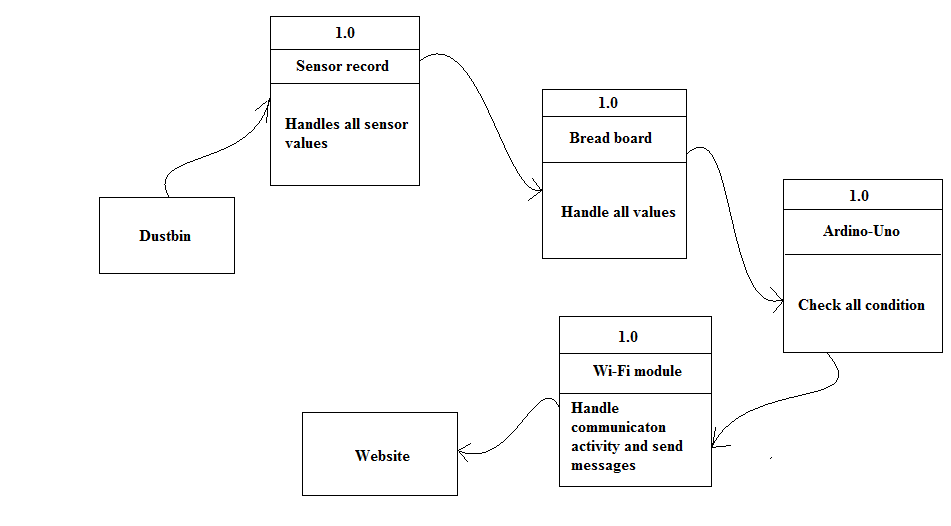
****

Figure 6.1.2 Level 1 Data Flow Diagram

**6.2 UML DIAGRAM**

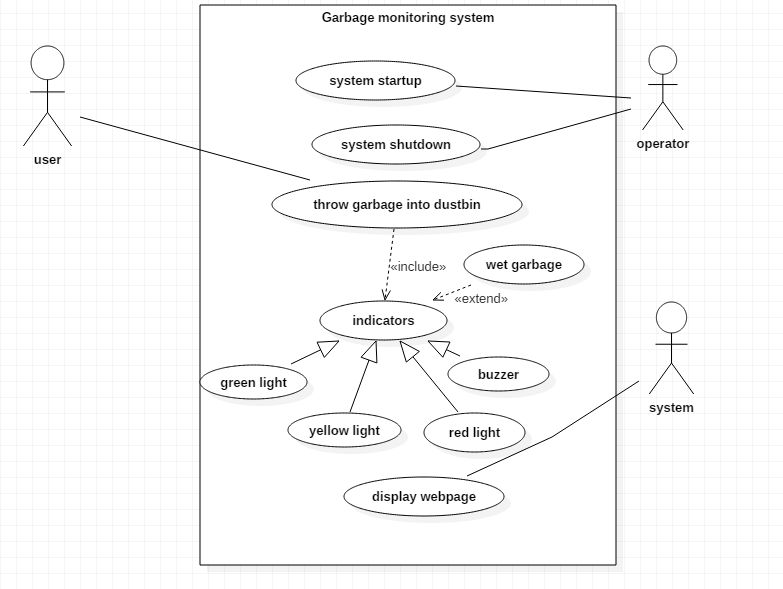
The **Unified Modeling Language** (**UML**) is a general-purpose, developmental, [modeling language](https://en.wikipedia.org/wiki/Modeling_language) in the field of [software engineering](https://en.wikipedia.org/wiki/Software_engineering) that is intended to provide a standard way to visualize the design of a system.

The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design. It was developed by [Grady Booch](https://en.wikipedia.org/wiki/Grady_Booch), [Ivar Jacobson](https://en.wikipedia.org/wiki/Ivar_Jacobson) and [James Rumbaugh](https://en.wikipedia.org/wiki/James_Rumbaugh) at [Rational Software](https://en.wikipedia.org/wiki/Rational_Software) in 1994–1995, with further development led by them through 1996.

In 1997 UML was adopted as a standard by the [Object Management Group](https://en.wikipedia.org/wiki/Object_Management_Group) (OMG), and has been managed by this organization ever since. In 2005 UML was also published by the [International Organization for Standardization](https://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO) as an approved ISO standard. Since then the standard has been periodically revised to cover the latest revision of UML.

**6.2.1 USE CASE DIAGRAM**

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different [use cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved.

****

|  |
| --- |
| Figure 6.2.1 : USE CASE DIAGRAM |

A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

**6.2.2 SEQUENCE DIAGRAM**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

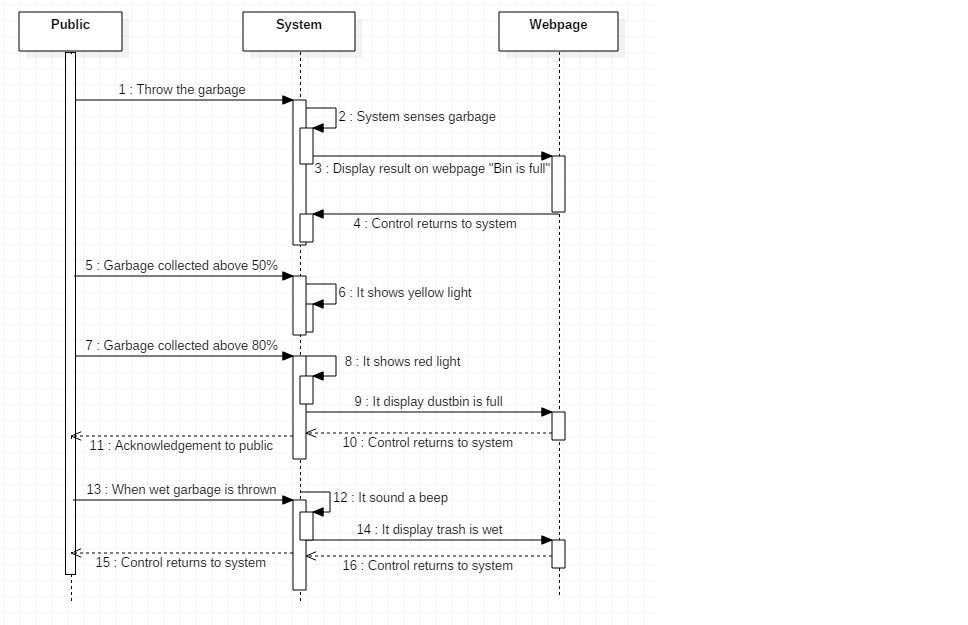


Figure 6.2.2: SEQUENCE DIAGRAM

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

**6.2.3 ACTIVITY DIAGRAM**

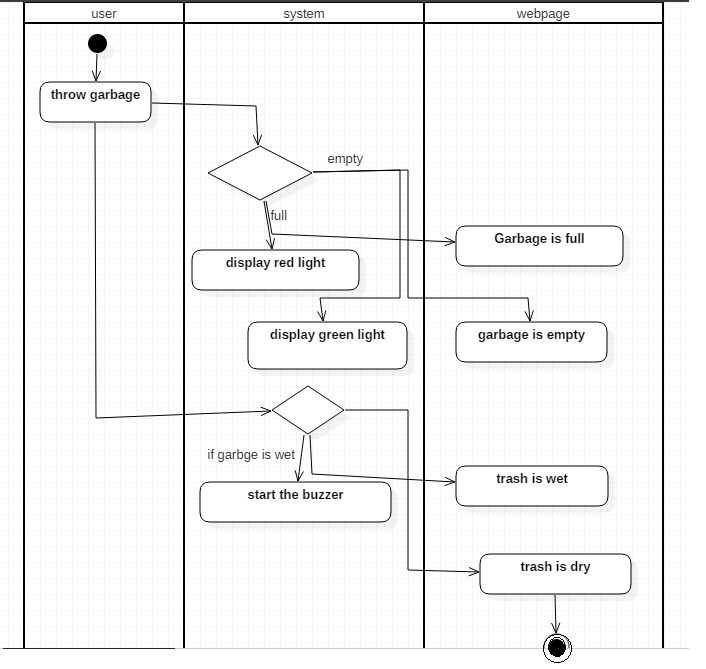
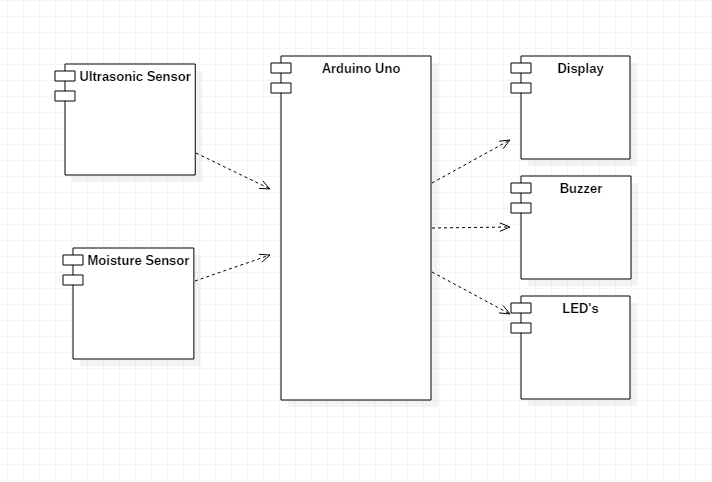
**Activity diagrams** are graphical representations of [workflows](https://en.wikipedia.org/wiki/Workflow) of stepwise activities and actions with support for choice, iteration and concurrency. In the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language), activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities.

Figure 6.2.3: ACTIVITY DIAGRAM

**6.2.4 COMPONENT DIAGRAM**

In the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language), a **component diagram** depicts how [components](https://en.wikipedia.org/wiki/Component_(UML)) are wired together to form larger components or [software systems](https://en.wikipedia.org/wiki/Software_system). They are used to illustrate the structure of arbitrarily complex systems.

|  |
| --- |
| Figure 6.2.4:- Component Diagram |

**6.2.5 DEPLOYMENT DIAGRAM**

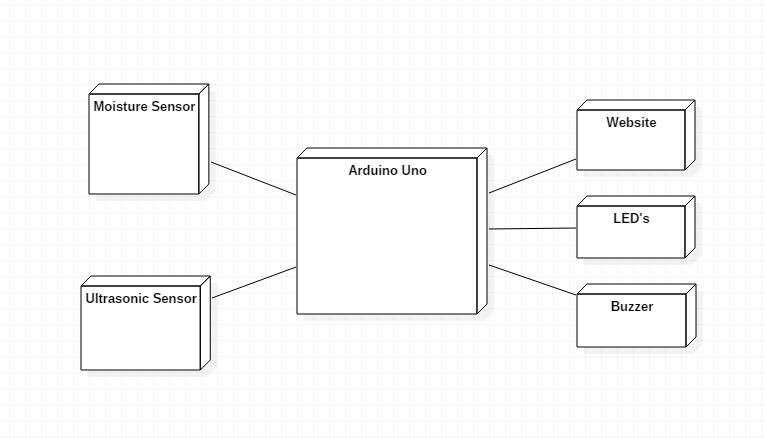
A **deployment diagram** in the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) models the *physical* deployment of [artifacts](https://en.wikipedia.org/wiki/Artifact_(UML)) on [nodes](https://en.wikipedia.org/wiki/Node_(UML)). To describe a [web site](https://en.wikipedia.org/wiki/Web_site), for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a [web server](https://en.wikipedia.org/wiki/Web_server), an [application server](https://en.wikipedia.org/wiki/Application_server), and a [database server](https://en.wikipedia.org/wiki/Database_server)), what software components ("artifacts") run on each node (e.g., [web application](https://en.wikipedia.org/wiki/Web_application), [database](https://en.wikipedia.org/wiki/Database)), and how the different pieces are connected.

Figure 6.2.5:- Deployment Diagram

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

**7. TESTING**

**7.1 INTRODUCTION**

A system should always be tested thoroughly before implementing it, as regards its individual programs. This is because implementing a new system is a major job which a lot of man hours and a lot of other resources, so an error not detected before implementation may cost a lot. Effective testing early in the process translates directly into long term cost saving from reduced number of errors. This is also necessary because in some cases, a small error is not detected and corrected before installation, which may explode into much larger problem.

Programming and testing is followed by the stage of installing the new computer based system. Actual implementation of the system can begin at this point using either a parallel or a direct changeover plan, or some blend of two. Testing and implementation of **“GARBAGE MONITORING AND MANAGEMENT SYSTEM”** is carried out as below.

Software testing is a critical element of Software Quality Assurance and represents the ultimate review of specification, design and coding. The purpose of product testing is to verify and validate the various work products viz. units, integrate unit, final product to ensure that they meet their respective requirements.

**7. 2 Testing Procedure**

Software Testing is the critical element of the Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is the process of checking whether software works according to the specification.

Testing will be performed by running the program using the test data. Testing is vital to the success of the system. It will also test whether the system identifies the problem correctly.

System is tested by following steps:

* **Unit Testing**:

Each program is tested individually using dummy records to see whether that program produce satisfactory reports.

* **Sequential Testing**:

The program, whose output will affect the processing done by another program, will be tested using dummy records.

* **System Testing**:

The system is corrected in such a way that it does not affect the forced system failure. This testing is done with low volumes of data

**7.3: Test Plan**

The testing department will test the newly developed system.

**Objective:**

* Define the activities required to and prepare for conducting Beta, system and user acceptance test.
* Communicate to all responsible parties the test strategy.
* Define deliverable and responsible parties.

**7.4: Test Strategy:**

The test strategy consists of a series of different tests that will fully exercise the system. The primary purpose of the test is to uncover the system limitations.

Following are the several tests that will be conducted:

**7.4.1 Unit/Module Testing**

Testing conducted to verify the implementation of the design for one software element (e.g., unit, module) is called unit testing. The purpose of unit testing is to ensure that the program logic is complete and correct and ensuring that the component works as designed.

In this module, each unit will go through Unit testing after the completion of the module. The bugs in module testing will be reported in Test Log document and will be reported to the developers. After fixing the bug successfully, one more iteration of module testing (Regression Testing) is done. This process is repeated till all critical test cases pass.

**7.4.2 Integration Testing**

Testing conducted in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated. The purpose of integration testing is to ensure that design objectives are met and ensures that the software, as a complete entity, complies with operational requirements. This type of testing will be done after all module test cases are passed through module testing, security testing, performance testing, user interface testing and regression testing.

**7.4.3 User Interface Testing**

Testing done to ensure that the application operates efficiently and effectively on each client machine.

**7.4.4 Security Testing**

The client needs to enter his login id and password. If the user name and password is valid then only the client is allowed to log in

**7.4.5 Performance Testing**

In developing the system, we are going to use Java which will reduce the response time. In Performance Testing, We are going to test Response time for each Screen. It is a type of non-functional testing.

Performance testing is testing that is performed; to determine how fast some aspect of a system performs under a particular workload. It can serve different purposes like it can demonstrate that the system meets performance criteria. It can compare two systems to find which performs better. Or it can measure what part of the system or workload causes the system to perform badly. This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions.

**7.4.6 Regression Testing**

Testing done to ensure that, the changes to the application have not adversely affected previously tested functionality. Here testing will take care of the test cases passed during the first module testing will not be affected in the subsequent rounds of module testing.

**7.4.7 Usability Testing**

Usability testing is an informal process, reviewing your design with a few users, or a formal process in an established usability lab. Either way, the purpose is the same i.e. learning first hand from the users where your design works and where it needs improvement.

The following points would be considered:-

* **Effectiveness:**

Accuracy and completeness with which specified users can achieve specified goals in a particular environment.

* **Efficiency:**

The resources expended in relation to the accuracy and completeness of the goals achieved.

* **Satisfaction:**

The comfort and acceptability of the work system to its users and other people affected by its use.

* **Learn ability:**

How easy it is for users to accomplish basic tasks the first time they encounter the design.

* **Memory ability:**

When the user returns to the design after a period of not using it, how easily can they re-establish proficiency.

**7.5 TESTING OBJECTIVES:**

* Testing is a process of executing a program with the intent of finding an error.
* A good test has a high probability of finding an as yet undiscovered error.
* A successful test is one that uncovers an as yet undiscovered error.

The objective is to design tests that systematically uncover different classes of errors and do so with a minimum amount of time and effort. Testing cannot show the absence of defects, it can only show that software defects are present.

**7.6 UNIT TESTING:**

**Interface**

Number of input parameters should be equal to number of arguments.

Parameters and argument attributes must match.

Parameters passed should be in correct order.

Global variable definitions consistent across module.

If module does I/O.

* File attributes should be correct.
* Open/Close statements must be correct.
* Format specifications should match I/O statements
* Buffer Size should match record size.
* Files should be opened before use.
* End of file condition should be handled.
* I/O errors should be handled.
* Any textual errors in output information must be checked.

**7.6 Testing of Each Component with Arduino Uno**

**7.6.1 Interfacing of Ultrasonic Sensor with Arduino Uno**

An Ultrasonic Sensor is a device that measures distance to an object using **Sound Waves**. It works by sending out a sound wave at ultrasonic frequency and waits for it to bounce back from the object. Then, the time delay between transmission of sound and receiving of the sound is used to calculate the distance.

It is done using the formula **Distance = (Speed of sound \* Time delay) / 2**

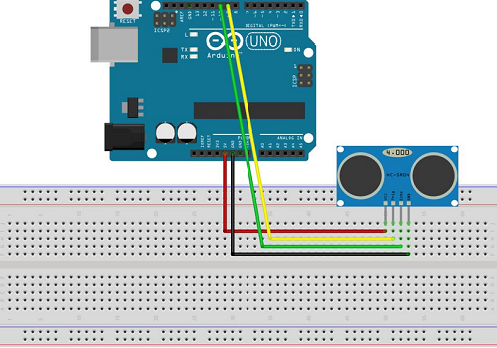
We divide the distance formula by 2 because the sound waves travel a round trip i.e from the sensor and back to the sensor which doubles the actual distance.

The HC-SR04 is a typical ultrasonic sensor which is used in many projects such as obstacle detector and electronic distance measurement tapes. In this Instruct able I'll teach you how to interface the HC-SC04 with an Arduino Uno.

**Component Used: -**

* Arduino Uno
* HC-SR04 Module
* Breadboard
* Jumper wires

**Circuit Diagram:-**



The connections are as follows: -

* Vcc to 5V Pin of the Arduino.
* Gnd to Gnd Pin of the Arduino.
* Trig to Digital Pin 9.
* Echo to Digital Pin 10.

**/\*\* Ultrasonic Sensor HC-SR04 interfacing with Arduino. \* \*/**

// defining the pins

const int trigPin = 9;

const int echoPin = 10;

// defining variables long duration;

int distance;

void setup()

{ pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

Serial.begin(9600); // Starts the serial communication

}

void loop()

{

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds duration = pulseIn(echoPin, HIGH);

// Calculating the distance distance= duration\*0.034/2; // Prints the distance on the Serial Monitor

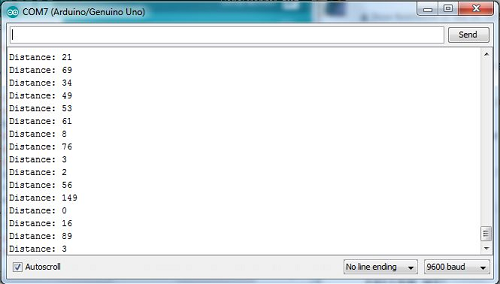
Serial.print("Distance: ");

Serial.println(distance);

}

**Result: -**

Once We upload the code, the board will begin to transmit the data to the computer .At that movement open the Serial Monitor, you will see the distance being displayed.



**7.6.2 Interfacing of Soil Moisture Sensor with Arduino Uno**

When you hear the word **Smart Garden**, one of the things that pop up to your mind is the **automatic measurement of the moisture content of the soil.** If you're building a Smart Garden that waters plants automatically and give you the readings of the wetness of the soil, then you will definitely need a **Soil Moisture Sensor**.

A typical Soil Moisture Sensor consist of two components. A two legged Lead, that goes into the soil or anywhere else where water content has to be measured. This has two header pins which connect to an Amplifier/ A-D circuit which is in turn connected to the Arduino.

The Amplifier has a Vin, Gnd, Analog and Digital Data Pins. This means that you can get the values in both Analog and Digital forms.

**Component Used: -**

* Arduino Uno
* Soil Moisture Sensor
* Breadboard
* Hook Up Wires

**How It Works: -**

Most soil moisture sensors are designed to **estimate soil volumetric water content based on the dielectric constant (soil bulk permittivity) of the soil.** The dielectric constant can be thought of as the soil's ability to transmit electricity. The dielectric constant of soil increases as the water content of the soil increases. This response is due to the fact that the dielectric constant of water is much larger than the other soil components, including air. Thus, **measurement of the dielectric constant gives a predictable estimation of water content.**

**Connections: -**

* Connect the two pins from the Sensor to the two pins on the Amplifier circuit via hook up wires.
* Connect the Vcc from the Amplifier to the 3.3V pin on the Arduino and the Gnd pin to the Gnd pin on the Arduino.
* Now connect the Analog Data Pin to the A0 pin on the Arduino (Since I'm interested in Analog Data).

**Code: -**

void setup() {

// initialize serial communication at 9600 bits per second:

Serial.begin(9600);

}

// the loop routine runs over and over again forever:

void loop() {

// read the input on analog pin 0:

int sensorValue = analogRead(A0);

// print out the value you read:

Serial.println(sensorValue);

delay(1); // delay in between reads for stability

}

**7.7 Test Results**

The listed tests were conducted in the software at the various developments stages. Unit testing was conducted. The errors were debugged and regression testing was performed. The integration testing will be performed once the system is integrated with other related systems like Inventory, Budget etc. Once the design stage was over the Black Box and White Box Testing was performed on the entire application. The results were analyzed and the appropriate alterations were made. The test results proved to be positive and henceforth the application is feasible and test approved.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR.NO. | DESCRIPTON | INPUT DATA | EXPECTED RESULT | ACTUAL RESULT | STATUS |
| 1 | Check modules and Components. | - | Components should work properly. | Components work properly. | PASS |
| 2 | Check Wi-Fi module connection with computer system. | Turn on Wi-Fi module and connect to computer system. | Wi-Fi module should connected with computer. | Wi-Fi module connected with computer. | PASS |
| 3 | Check GUI of Website. | Open the website. | It should easy graphical user interface (GUI). | Easy graphical user interface (GUI). | PASS |
| 4 | Check output of ultrasonic sensor. | Empty bucket. | It should display 0% of garbage level in bucket. | It display 0% of garbage level in bucket. | PASS |
| 5 | Check output of ultrasonic sensor. | Empty bucket. | It should display 0% of garbage level in bucket and green LED is on. | It display 0% of garbage level in bucket and green LED is on. | PASS |
| 6 | Check output of ultrasonic sensor. | 50% of garbage in bucket. | It should display 50% of garbage level in bucket and Yellow LED is on. | It display 50% of garbage level in bucket and Yellow LED is on. | PASS |
| 7 | Check output of ultrasonic sensor. | 85% of garbage in bucket. | It should display 85% of garbage level in bucket and display warning massage on system and Red LED is on. | It display 85% of garbage level in bucket and display warning massage on system and Red LED is on. | PASS |
| 8 | Check warning message. | 85% of garbage in bucket. | It should display warning message. | It does not display warning message. | FAIL |
| 9 | Check working of moisture sensor. | Throw wet garbage on bucket. | Moisture sensor should work properly. | Moisture sensor work properly. | PASS |
| 10 | Check warning message of moisture sensor. | Throw wet garbage on bucket. | Busser should on. | Busser is on. | PASS |

**8.1 ADVANTAGES:**

* Real time information on the fill level of the dustbin. Deployment of dustbin based on the actual needs.
* Cost Reduction and resource optimization.
* Fewer smells
* Cleaner cities
* Intelligent management of the services in the city. Effective usage of dustbins.

**9.1 EXPERIMANTAL RESULTS**

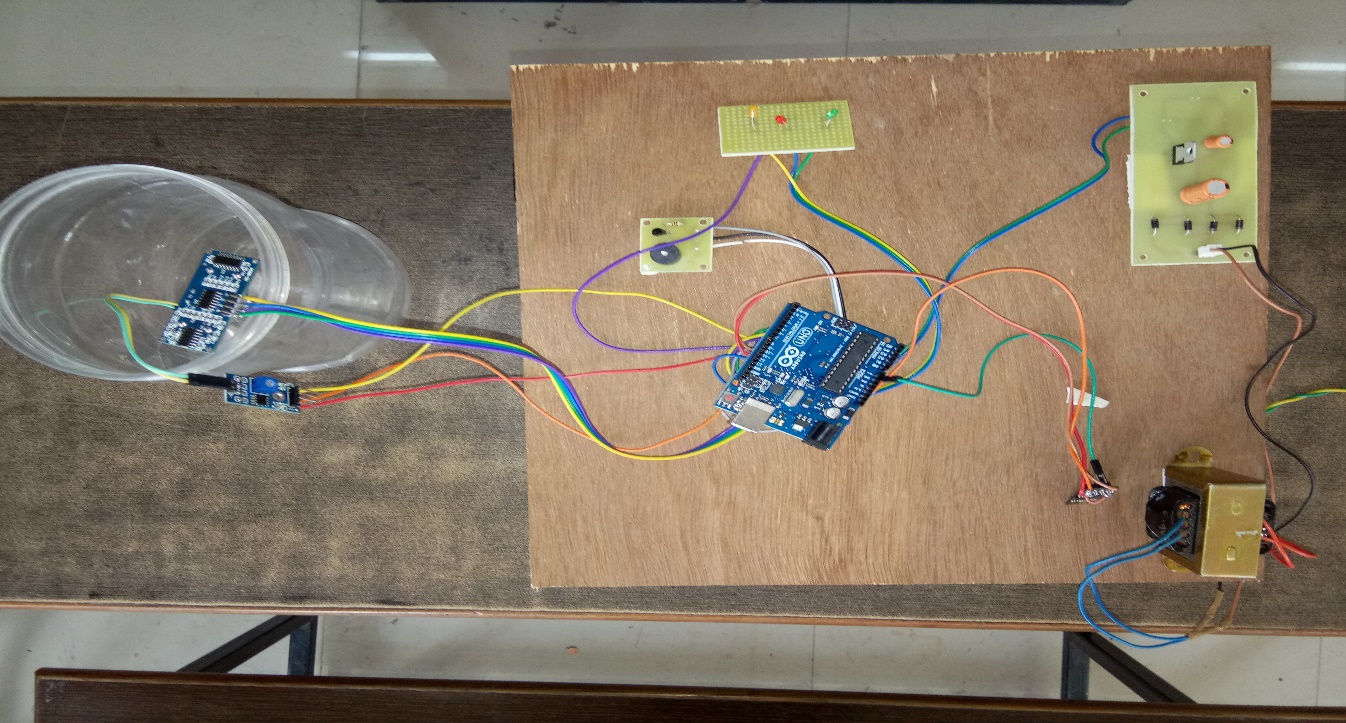
****

Fig. Overall System

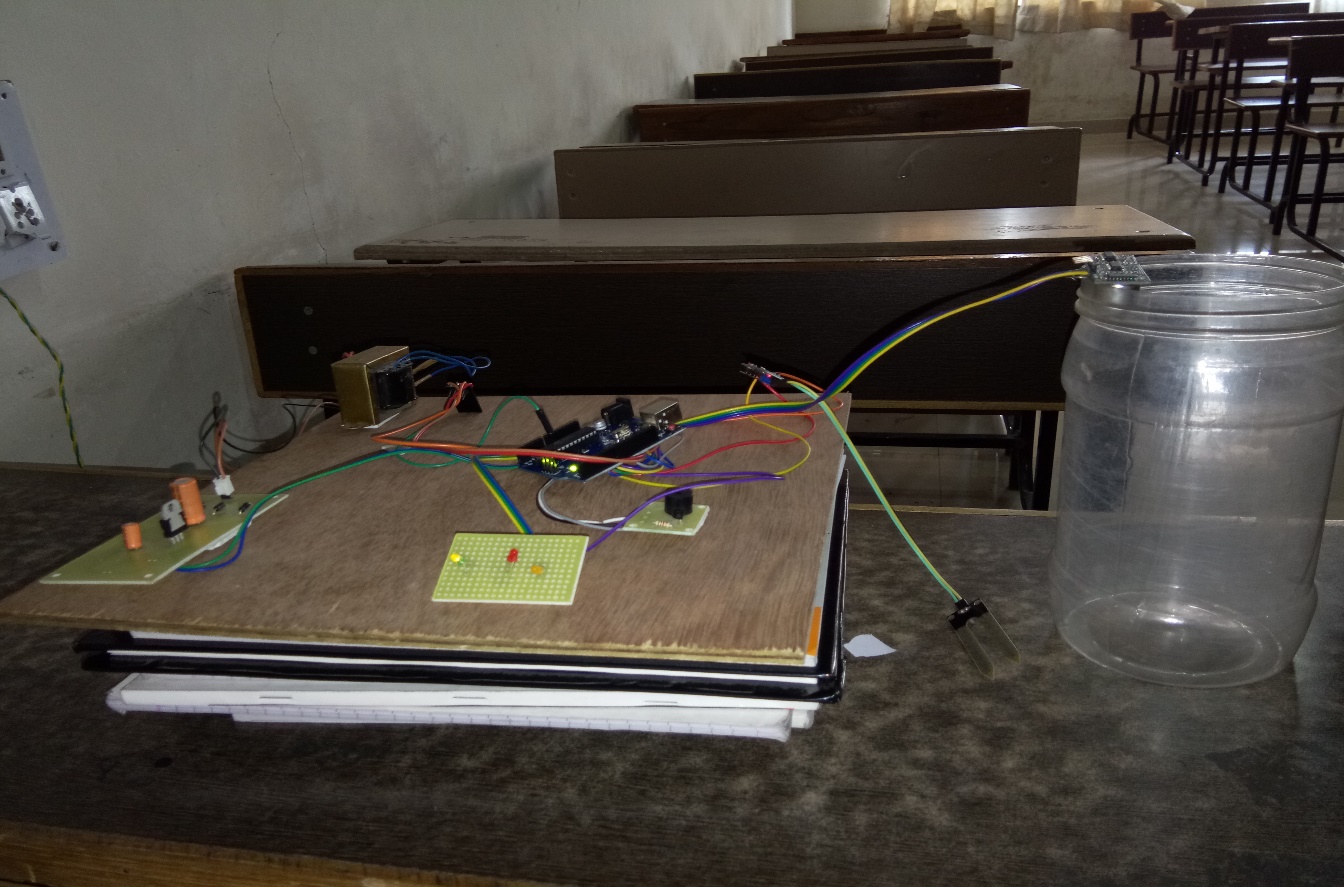
****

Fig. Empty State

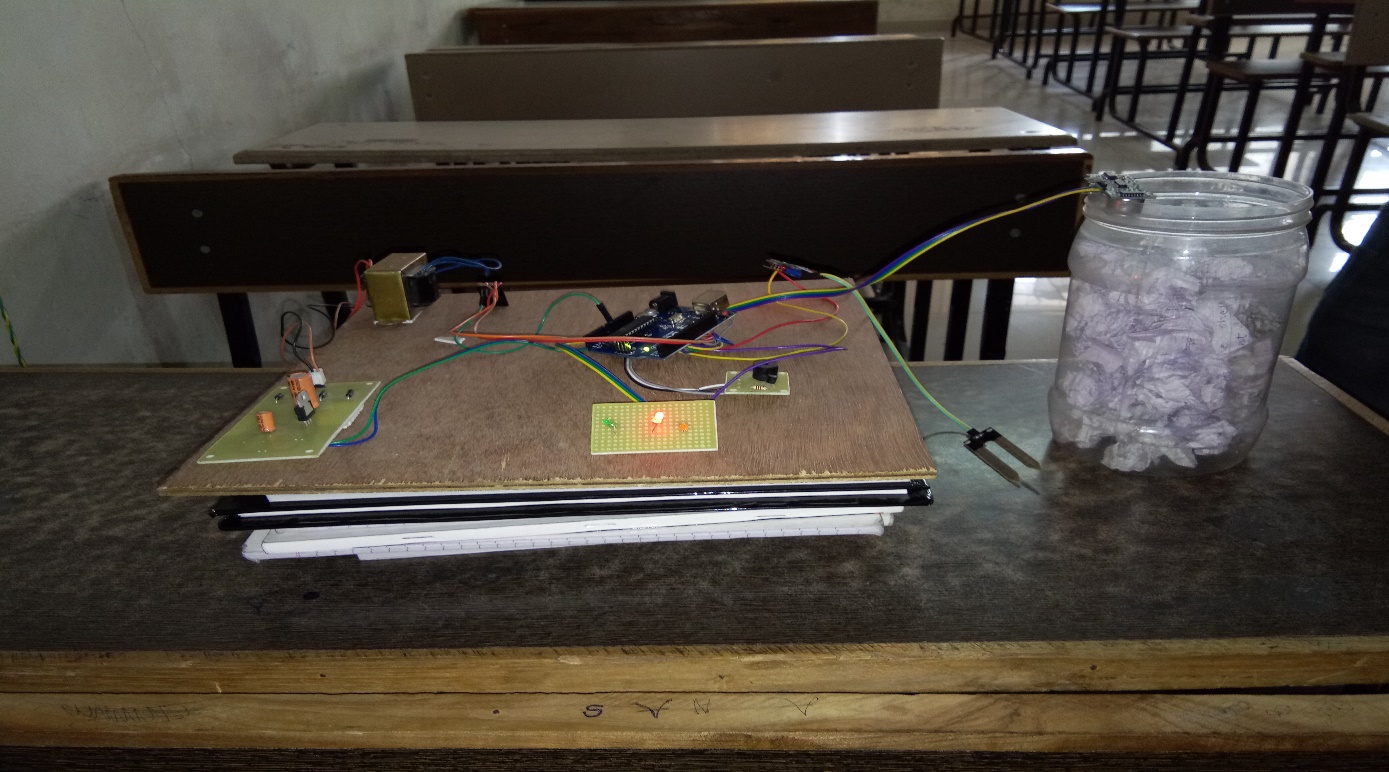
****

Fig. Middle State

****

Fig. Full State

**10.1 FUTURE SCOPE**

* The system provides us with the real-time information and status of garbage bins located in different areas.
* The real-time information we can monitor the bins and once the bins are full the workers can collect the garbage and set them to empty again.
* This system is cost effective and can be accessed from anywhere.
* Traffic can be controlled as the workers collect the garbage only when the bin is full whereas in traditional way workers collect the garbage daily whether the bin is filled or not.
* This system has a future scope where this system can be used with time stamp where real-time clock will be made available to the authority stating at what time Garbage bins was full and at what time did the garbage is collected from the smart Garbage Bins

**11.1 CONCLUSION**

In this project, an integrated system of Wi-Fi modem, IOT, GSM, Ultrasonic Sensor is introduced for efficient and economic garbage collection. The developed system provides improved database for garbage collection time and waste amount at each location. We analyzed the solutions currently available for the implementation of IOT. By implementing this project, we will avoid over flowing of garbage from the container in residential. It can automatically monitor the garbage level and through the Led indication we can get to know whether the dustbin will full, half-filled or empty. The technologies which are used in the proposed system are good enough to ensure the practical and perfect for solid garbage collection process monitoring and management for green environment.

**12.1 REFERENCES**

* www.wikipedia.com
* www.google.com.
* www.infoprojects.com.
* www.projects4students.com.
* www.pmc.gov.in.
* www.bioenabletech.com.
* www.hindawi.com.