Table of Content

Introduction and Objective

This paper presents a cost-effective and efficient design of Smart Dustbin which aims to reduce the risk of contamination to the Hospital workers. This system is based on two major operations. The first being it shows dustbin garbage capacity. For that, we have LCD (16x2) equipped I2C and display garbage level. And second is that it opens and closes its lid sensing the presence or absence of the user respectively.

The working prototype is a powered 5 V DC power supply. The input provided by the operator is a parallel input that is converted to series data with the help of an encoder which is transmitted wire through the USB cable.

For opening and closing, we are using a Servo motor equipped with the limiting switch. The Ultrasonic sensor has a trigger and echo which emits and receives ultrasonic sound respectively. They sense the presence of the user through the process of reflection of ultrasonic waves and show the change in electrical parameter which is then compared with the known value with the help of the operational amplifier. If there is a user, the motor operates causing the lid to open, and if the user leaves the dustbin, the lid closes itself.

The main goal of this IOT-based project is to help overall human beings by reducing the risk of contamination of the hospital cleanliness staff by a contactless collection of the waste. We observed that during the COVID-19 pandemic the risk factor for the cleaning staff of the hospital was high as they had to be in contact with the confirmed contaminated patients for cleanliness also, they get in touch with other non-contaminated people. So, we can use a smart dustbin instead of a human for such works.

SYSTEM ANALYSIS

Identification of Need:

Hospital staffs are among the frontline workers who fight against contagious diseases. They are constantly in contact with an infected person in the hospital and with a healthy person during their off-hours. So, they can serve as a disease carriers. This was among the major problem which made COVID-19 difficult to control. In this era, we must be prepared for more dangerous outbreaks than COVID-19. The exponential growth of population and increase in population density of the world possesses difficulty in controlling contagious diseases. So, to minimize this problem to some extent our smart dustbin, which is a remotely operated waste bin can be some help.

Preliminary Investigation:

India has continuously been facing the COVID issues however, currently to feature on therefore spreading of the virus is additionally become a problem. Most of the staff members of the hospitals use their hands to put garbage in the dustbin.

Some issues faced are given below-

- The problem of spreading diseases in hospitals, residential flats, etc.
- The problem with checking the dustbin from time to time is its garbage level.
- Takes effort to put garbage in the dustbin.

To agitate this drawback, we've got come back up with IoT device that is integrated with sensors and motors i.e., “Smart dustbin” and that I am certain this can extremely assist you to alter your issues and problems that area can face for Garbage. Its varied advantages such as you will simply notice it saves our time, it maintains its lid to close after putting garbage and it helps in not to spread diseases.

Feasibility Study:

The feasibility of software and hardware can be tested like: Is the project technically useful? As in our case, smart Dustbin we have some existing examples of the same, therefore no technical impracticability is there.

Finance - Is it financially feasible? Is it too expensive to develop?

Time- Is it going to take a lot of time? We planned each phase, and it seems to be monitored and in time so no extra time costs will be added

Resources- Do we have enough resources to be able to do it?

There are four categories of feasibility tests: Operational feasibility, Technical feasibility, Schedule feasibility, and economic feasibility.

1. **Technical Feasibility**- The dustbin we have developed is very easy to use. It was built in a way that suits the user's comfort zone. There are sufficient physical components used in the bin to provide a robust service to the user.

2. **Operational Feasibility-** A major problem in everyday life is the waste and diseases it propagates. And this problem is indirectly falling behind. Always checking the level of the bins is a major concern. With this project, the problem is solved all at once. By using, an LCD user can view the waste level and it will also save a lot of time.

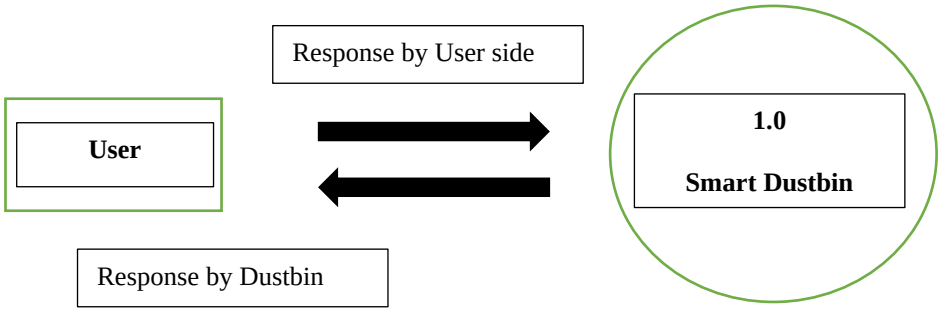
3. **Economic feasibility**: As with the software part, we had to make zero investment for now, which makes it economical but for the model, we have to invest some money in the hardware part, and within the long-term with a lot of options additional like easy handling, still a more robust economic practicableness will be achieved.

DFD

Data Flow Diagram:

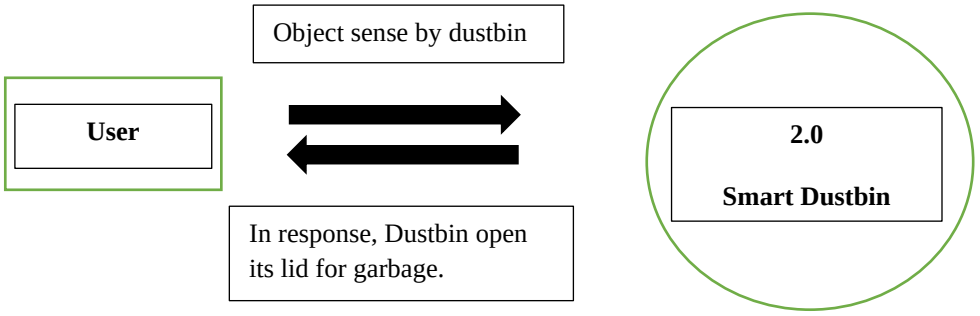
Level 1 for smart dustbin:

The data flow of the Smart dustbin can be seen in the below Data Flow Diagram. Smart Dustbin is an IOT-based application that is going to be accessed by the User.

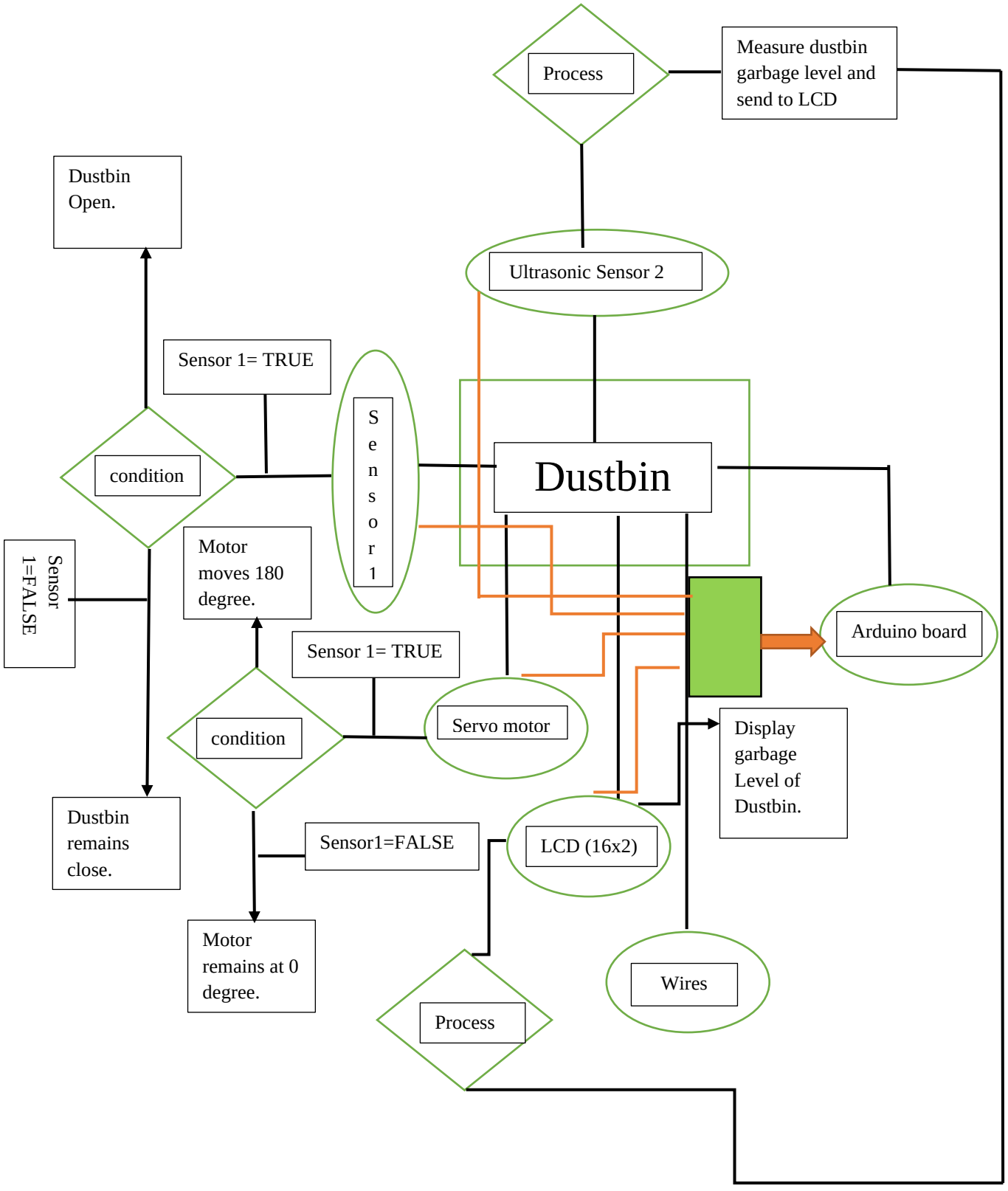


Level 2 DFD

In DFD level 2 of working of the dustbin for the user is shown. Steps after visiting near dustbin the sensor senses the presence of an object or user and lifts the dustbin lid for three seconds and then it will close.

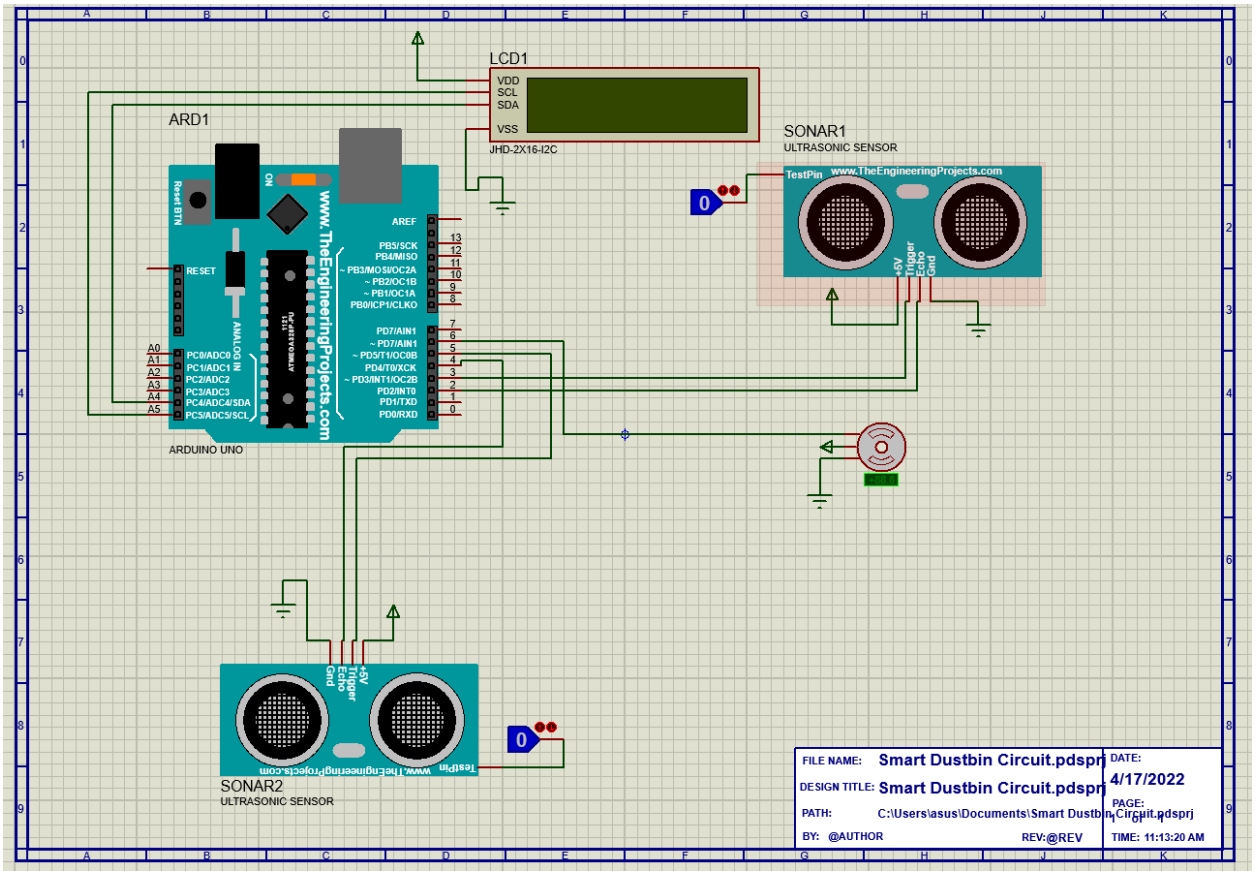


ER Diagram: Below Image of ER diagram that represents the flow of data for the Smart dustbin.



ER diagram for Smart Dustbin

Circuit Diagram:



Explanation of circuit diagram:

- Firstly we have implemented an Arduino UNO board it acts as the CPU of the Smart Dustbin.

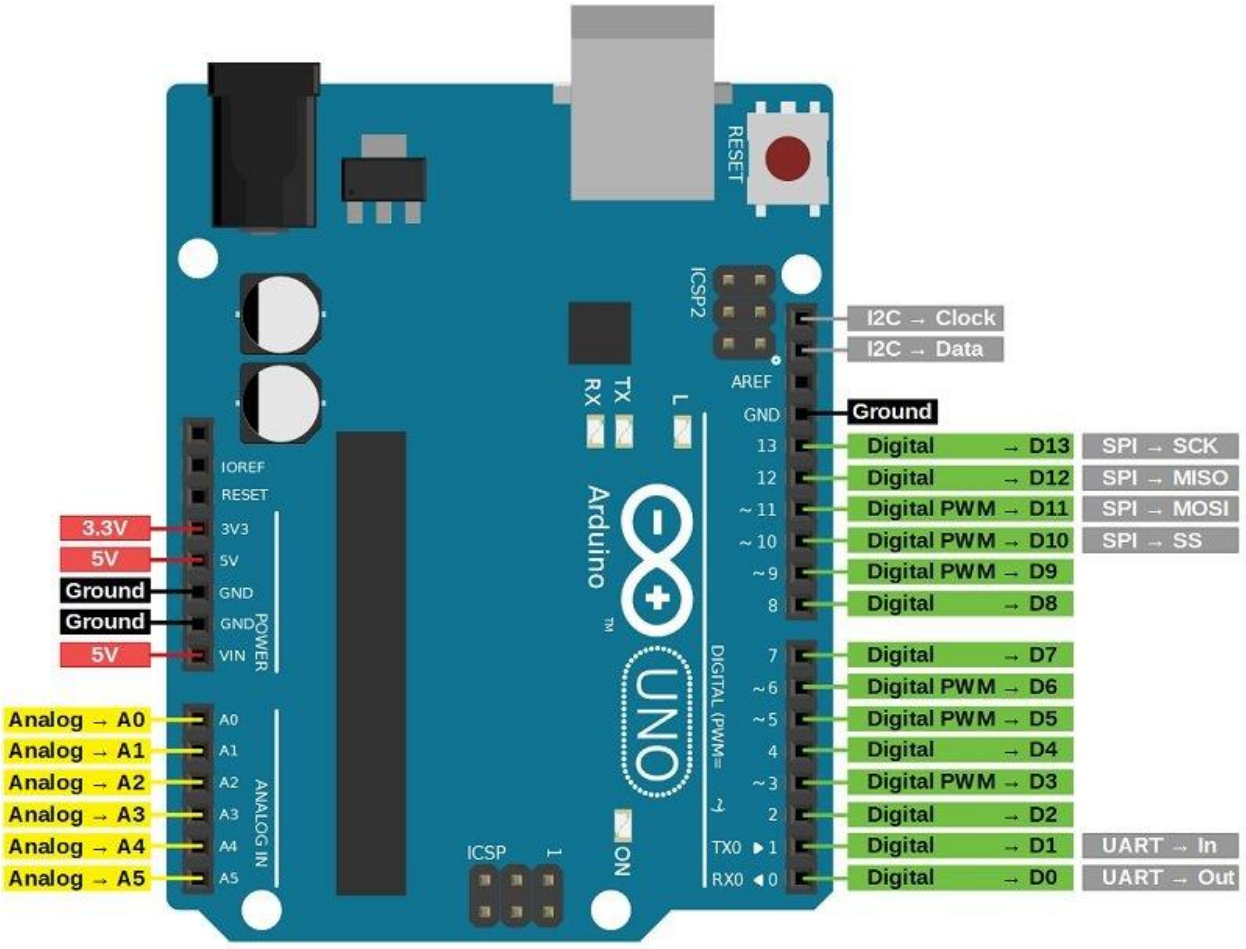
Sensor	Pin number
Ultrasonic sensor1 (VCC, GND, Trigger, Echo)	Power pin, ground pin, 3,2
Ultrasonic sensor2 (VCC, GND, Trigger, Echo)	Power pin, ground pin, 5,4
Servo Motor (VCC, GND, Data pin)	Power pin, ground pin, 6
LCD (VSS, VDD, SDA, SCL)	Power pin, ground pin, A4,A5

H/W & S/W Requirement Specification

Components required for Smart Dustbin:

1. Arduino UNO Board:

Arduino Uno is a microcontroller board primarily based totally at the ATmega328P. It has 14 digital input/output pins (of which 6 of them used as PWM outputs), 6 analog inputs, a sixteen MHz quartz crystal, a USB connection, a strength jack, an ICSP header, and a reset button. It consists of the whole thing had to help the microcontroller; genuinely join it to a pc with a USB cable or strength it with an AC-to-DC adapter or battery to get started.



Technical specification:

Input Voltage	7-12V
Analog Input Pins	6
EPROM	1kb
SRAM	2kb
Operating voltage	5volts
PWM Digital Pins	6
Led	13 pin
Clock speed	16MHz
DC Current per Pin	20 mA
Microcontroller	At mega 328p
Digital Pins	14

Power USB:

The Arduino Power USB port can be powered by our PC with the help of USB cable. Simply connecting the USB cable to the USB port.

Power (Barrel Jack):

This board can also powered directly from the AC power supply by plugging into the barrel outlet.

Voltage Regulator:

The basic work of voltage regulator is to provide the necessary amount of voltage to the board.

Crystal Oscillator:

The crystal oscillator helps Arduino with timing problems. How Arduino calculates time? The answer is to use the Crystal Oscillator. The number printed on the top of the Arduino crystal is 16,000H9H. It tells us that the frequency is 16,000,000 Hertz or 16MHz.

Reset:

You can reset your Arduino board, i.e. start your program from the beginning. You can reset the UNO board in two ways. First, use the reset button on the dashboard. Second, you can connect an external reset button to the Arduino pin labeled RESET.

Pins (3,3, 5, GND, Vin)

3.3V (6) − Supplying 3.3 volts output

5V (7) − Supplying 5 volts output

Most components used with the Arduino board work well with 3.3 volts and 5 volts

GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

Vin (9) − This pin can also be used to power the Arduino board from an external power source, such as a USB cable. B. an AC power supply to supply power.

Analog pins

The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor such as humidity sensor or temperature sensor and convert it to a digital value that can be read by the microprocessor.

Main microcontroller

Each Arduino board has its own microcontroller. You can assume it's your board's brains. The main IC (integrated circuit) on the Arduino differs slightly from board to board. The microcontrollers usually come from ATMEL. You need to know what IC your board has before uploading a new program from the Arduino IDE. This information is available at the top of the IC.For more details on the structure and functions of the IC, see the data sheet.

ICSP pin:

Mostly, ICSP (12) is an AVR, a small programming head for Arduino consisting of MOSI, MISO, SCK, RESET, VCC and GND. It's often referred to as SPI (Serial Peripheral Interface), which you might think of as an "extension" of the output. They subordinate the output device to the SPI bus master.

Power LED :

When we connect our Arduino UNO board to a power pin, the power Led starts blinking, if the led is not blinking then there must be something missing with the connection

Tx and Rx LED:

There are two designations on your circuit board: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, on the digital pins 0 and 1, indicate the pins responsible for serial communication. Second, the TX and RX LED. The TX LED flashes at different speeds while the serial data is being sent. The blink rate depends on the baud rate used by the card. RX flashes during reception.

Digital I/O:

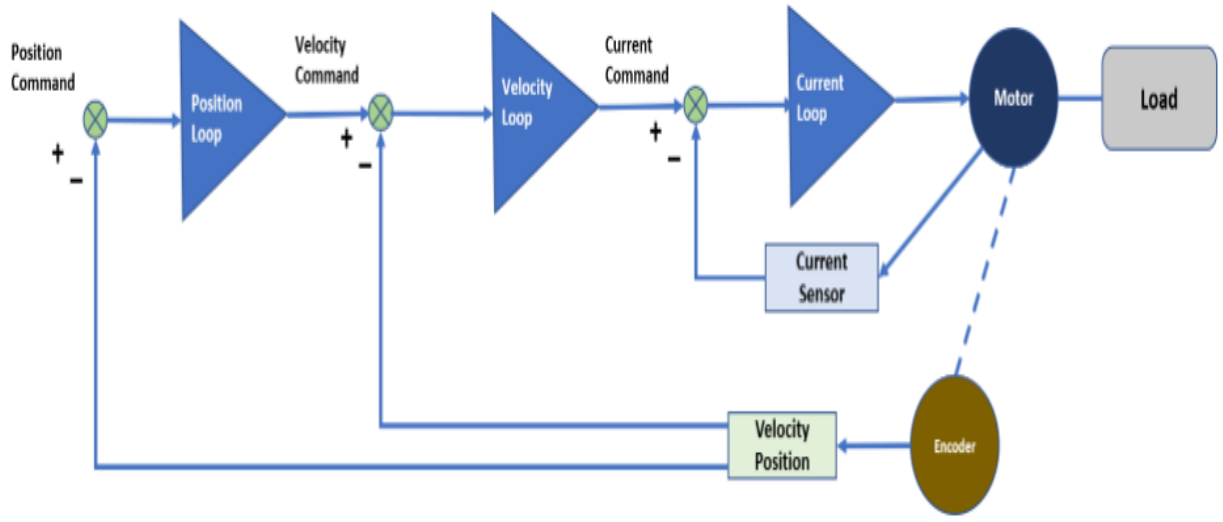
The UNO board has 14 digital input/ output pins in which six pins are PWM pins. These pins having logical values either 0 or 1. "~" this is the symbol to represent those pins.

AREF:

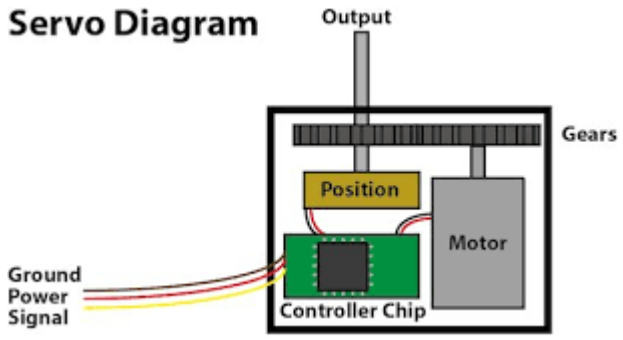
AREF stands for analog reference. It may be used to set an external reference voltage (0-5 volts) as the upper limit for analog input pins.

2. Servo motor

Servo motors are electromechanical devices that generate torque and velocity based on the supplied current and voltage. The servomotor operates as part of a closed-loop system and uses a feedback device to provide the torque and speed commanded by the servo controller to close the loop.



This is a linear or angular position, acceleration, And speed. It consists of a motor coupled to a position feedback sensor. You also need a relatively sophisticated controller. In many cases, you need a dedicated module designed specifically for servomotors.



Servo motor are frequently used in remote control toys, vehicles to control the direction of movement, and also as a motor to move the tray of a CD or DVD player. In addition, there are hundreds of uses for servomotors that we see in our daily lives.

These motors are part of a servo mechanism. It consists of three major parts: feedback, motors, devices, and control electronics. It can be AC or DC, rotary or linear, brushed or brushless, and of any size.

GND Pin: It is connected to the ground.

Power Pin: It is connected to the VCC for power.

Signal pin: Connect to the data pin on the Arduino board to accept commands.

3. LCD Display

It stands for liquid crystal display. It is a type of electronic display which is used in a variety of applications, like: calculators, mobile phones, and televisions. It primarily suitable for multi-segment and 7-segment light emitting diodes. The main goal of using this module is economical. Easy-to-program animations with unlimited display of custom characters, specials, and animations.

Features:

The operating voltage of this LCD monitor is 4.7 V 5.3 V.
It contains two rows, each with the ability to produce 16 characters.
Current consumption is 1mA with no backlight
Each character can be constructed with a 5 × 8-pixel array
Alphabets and alphanumeric numbers on LCD screens
The display can function in two modes, 4-bit, and 8-bit.
These can be available in blue and green with backlighting It
displays some customized characters generated.

LCD pin configuration:

Pin Number	Use
Pin1	It is a ground pin used for ground connection.
Pin2	It is a power pin used for voltage supply to Lcd
Pin3	It is used to adjust the brightness.
Pin4	It is used for switching the control between data registers.
Pin5	It is used for switching read and write mode.
Pin6	It should be high to perform read and write operation.
Pin7-14	These are data pins to transmit data to the display.
Pin15	It is for backlighting the lcd
Pin16	It is for ground connection.

I2C:

I2C is short for Inter-IC. And it's a kind of BUS. This was developed by Philips Semiconductors.
I2C is one-end, multi-washer, multi-master, packet-switched, synchronous serial bus namely. Several chips can be connected to the same bus. I2C uses only two two two-way open manifold or open drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL) that are drawn up with resistors. Typical voltages used are +5V or +3.3V, even if alternative voltage systems are acceptable.

A total of 16 pins are connected to i2c; the module is shown in the figure to reduce the pins for better connection.

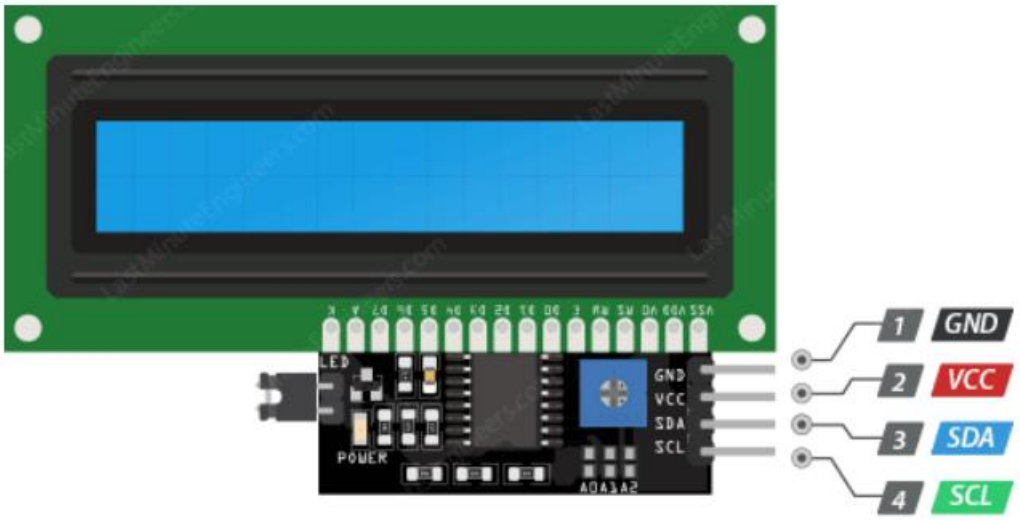


Fig: LCD 16x2 With I2C module

4. Ultrasonic Sensor

It is an electronic device which is used to measures distance of an object by emitting ultrasonic waves and converting the sound reflected through the electrical signal. Ultrasound-waves move faster than audible sounds (sounds that humans are unable to hear). Ultrasonic transducers are composed of two major components: the transmitter and the receiver.

Ultrasonic sensors are also used as level sensors to detect, control liquid levels and levels monitor levels within enclosed containers. Particularly, ultrasound technology has allowed the medical industry to create images of internal organs, to identify tumors.

Computation of the distance between the object and the sensor, the sensor measures the time between the transmitter making the sound and the contact with the receiver. The formula in this calculation is Distance= speed of sound x time taken/2; velocity of sound 343m/s.

VCC Pin: It is for the connect of the power pin on the Arduino UNO board.

Trigger Pin: It is an input pin, used to for the measurement by transmitting ultrasound waves while keeping that pin high for 10us.

Echo Pin: This pin generally receives that sound produced by the trigger pin and calculate time.

GND pin: It is a ground pin ;used to connect to system ground.

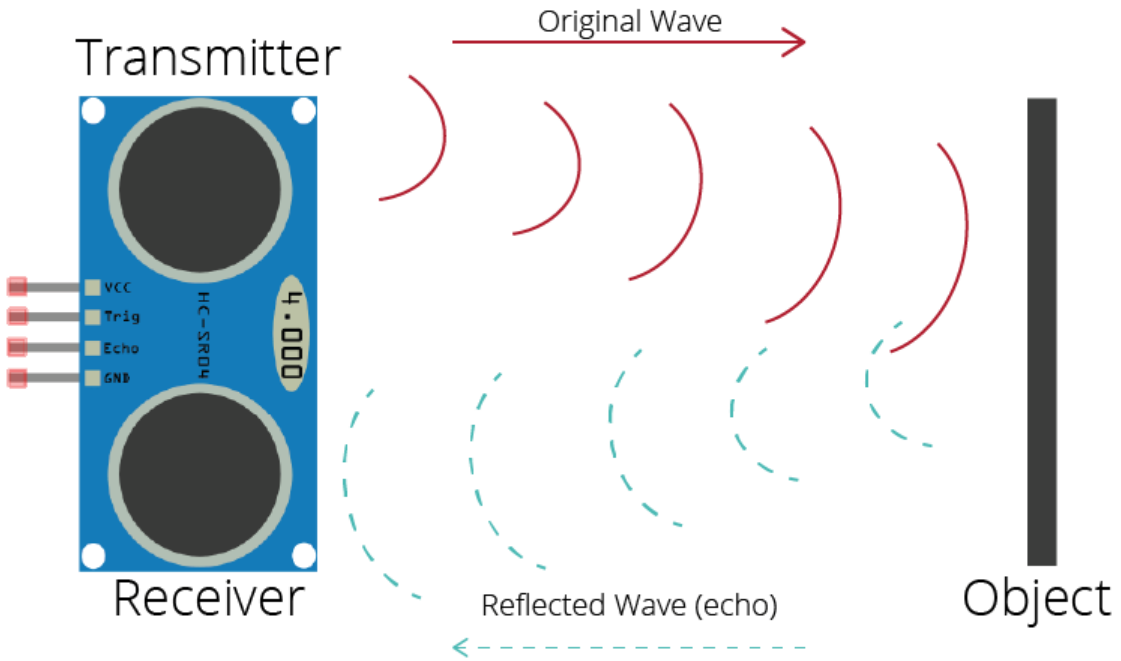


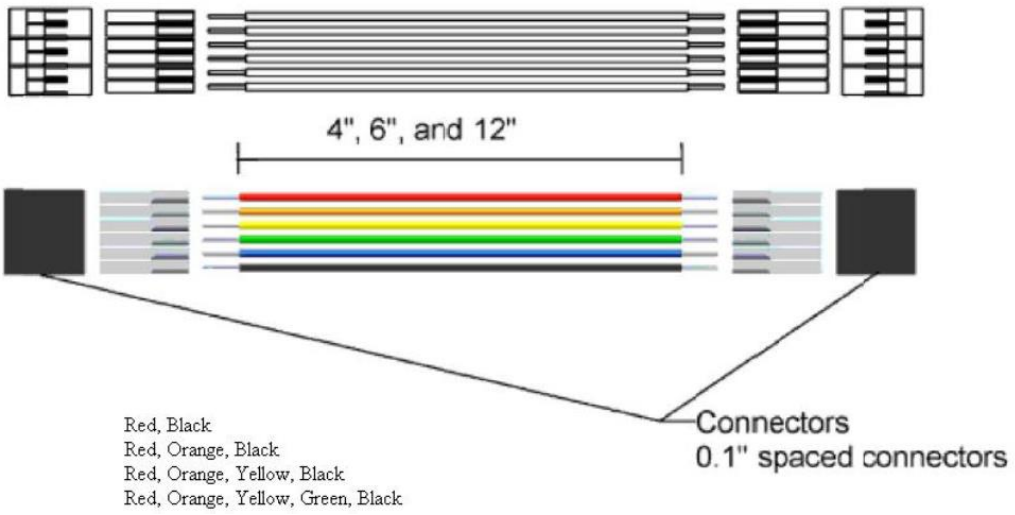
Fig: Ultrasonic sensor working.

5. Jumper Wires:

A jumper-wire is an electric wire, or a group of them in a cable, with a connector or pins at either end, that is normally used to interconnect components of a breadboard or other prototype or test circuit, within it. or with other equipment or components, weld-free.

Types of Jumper Wires.

- M to F: Male to Female.
- M to M: Male to Male.
- F to F: Female to Female.



Software Requirement

- Arduino IDE
- C++

Screen Shots with coding

```
/*#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include<Wire.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo smotor;

#define ePin 2
#define tPin 3
#define ePin1 4
#define tPin1 5

long duration;
int distance;
int set_cm =20;

void setup() {
 lcd.init();
 lcd.backlight();
 pinMode(tPin, OUTPUT);
 pinMode(ePin, INPUT);
 pinMode(tPin1, OUTPUT);
 pinMode(ePin1, INPUT);
 smotor.attach(6);
 smotor.write(0);
 Serial.begin(9600);
 void loop() {

 digitalWrite(tPin, LOW);
 delayMicroseconds(2);

 digitalWrite(tPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(tPin, LOW);

 duration = pulseIn(ePin, HIGH);
 distance = duration * 0.034 / 2;
 Serial.print("Distance: ");
 Serial.print(distance);
```

```
Serial.println(" cm");
 detect();
 if(distance<set_cm){
  smotor.write(0);
  delay(20);
  smotor.write(180);
 delay(3000);
  }
if(distance>set_cm)
{
smotor.write(0);
}
delay(100);
}
void detect(){
long duration1;
int distance1;
 digitalWrite(tPin1, LOW);
 delayMicroseconds(2);
 digitalWrite(tPin1, HIGH);
 delayMicroseconds(10);
 digitalWrite(tPin1, LOW);
 duration1 = pulseIn(ePin1, HIGH);
 distance1 = duration1 * 0.034 / 2;
 Serial.print("Distance1: ");
 Serial.print(distance1);
 Serial.println(" cm");
 if(distance1>=28)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN IS EMPTY");
 }
 if(distance1<=25 && distance1>22)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN 10% FULL");
 }
 if(distance1<=22 && distance1>20)
 {
```

```
lcd.setCursor(0, 0);
 lcd.print("DUSTBIN 20% FULL");
 }
 if(distance1<=20 && distance1>17)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN 30% FULL");
 }
 if(distance1<=17 && distance1>14)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN 40% FULL");
 }
 if(distance1<=14 && distance1>11)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN 50% FULL");
 }
 if(distance1<=11 && distance1>8)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN 60% FULL");
 }
 if(distance1<=8 && distance1>6)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN 80% FULL");
 }
 if(distance1<=6 && distance1>3)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN 90% FULL");
 }
if(distance1<=3 && distance1>1)
 {
 lcd.setCursor(0, 0);
 lcd.print("DUSTBIN IS FULL"); } }*/
```

Fig: IOT-based Smart Dustbin Model.

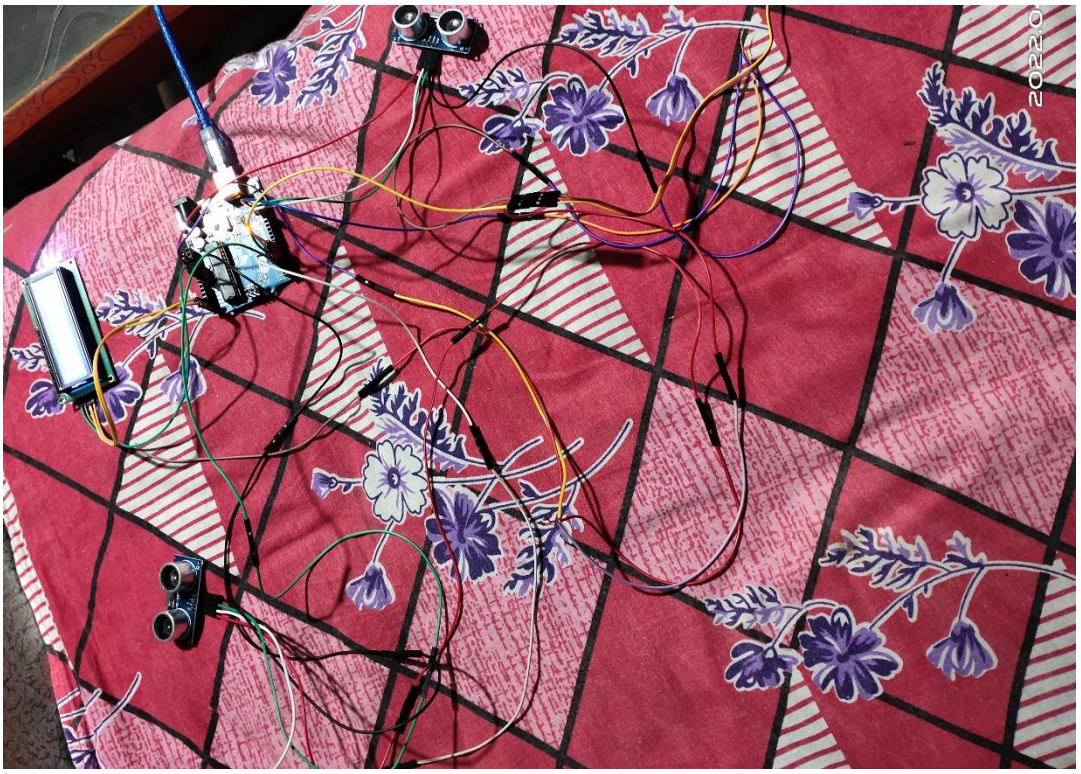


Fig: Wire connection through all components.

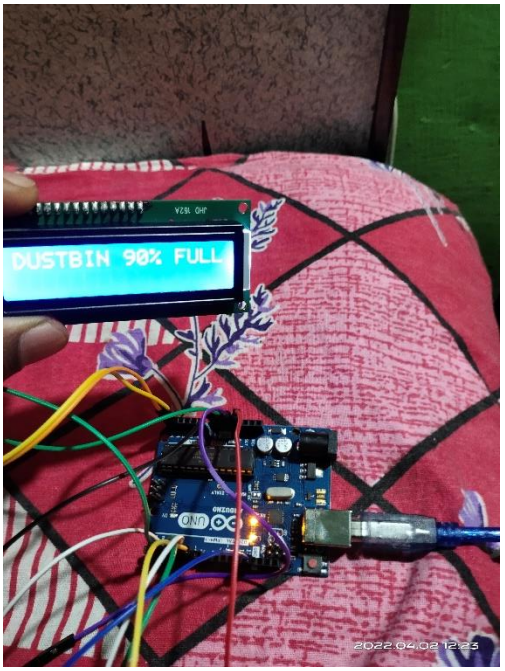


Fig: Sensor 2 claulate distance of garbage and displays on LCD.

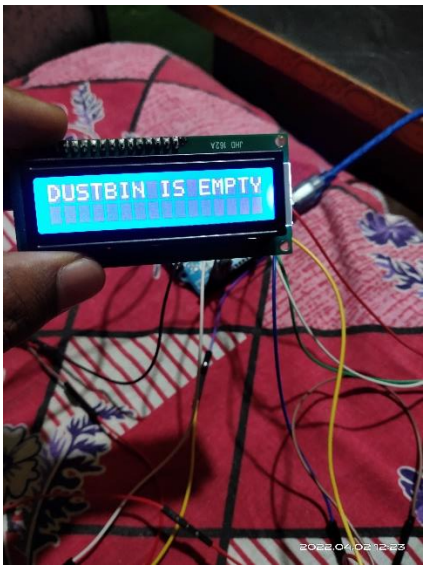


Fig: When sensor 2 detect nothing inside Dustbin it display empty.

Validation Checks

We have checked the validation for all components of the Smart dustbin for its working.

Ultrasonic Sensor 1	Working properly, and detecting objects within in range of 20 cm.
Ultrasonic Sensor 2	Working accurately and measuring the garbage level of the Dustbin.
Servo Motor	Checked No Error found working fine 0 to 180 degrees.
Lcd Display (16x2)	Working fine display garbage levels.
Arduino UNO	It is working and controlling all components of Smart Dustbin.

Implementation & Maintenance

Implementation: -

This system consists of different modules for better working of smart Dustbin. The Servo motor is used to open the lid of the Dustbin after detecting an object by ultrasonic sensor1. The LCD is used as the dashboard to show the garbage level of the Dustbin and empty or full status also. These allotment details are continuously updated by IOT sensors.

Team Involved: - This project is implemented by a team of three members.

Team Member Name	Task Completed
Sumit Singh	IOT based smart dustbin Design
Abhishek Kumar	Testing, Coding
Nikhil Thakur	Information Gathering and Report

Task: - We started developing the project, this project contains various tasks such as gathering information, preparing work breakdown Structure, preparing a work progress report, and then starting coding for command giving.

Implementation Schedule: - This project was approved in January 2022 and the complete month is April 2022.

Testing

The test is performed because it detects bugs before the delivery project to the customer, which ensures the quality of the software. It makes the software more reliable and more user friendly. Check the end-to-end workflow of an application or software as a user is known as the system test. These are end-to-end trials where the trial environment is like the production environment.

After generating the source code, the software must be tested to discover as many errors as possible prior to delivery to the customer. Our objective is to design a set of test cases that have a high probability of finding mistakes. Software testing techniques provide systematic guidance in test design. Practice the fields of program inputs and outputs to discover mistakes in program function, behaviour, and performance. During the first stages of testing, a software engineer carries out all testing. However, as the testing process progresses, test specialists can become tangled Examinations and alternative activities will uncover mistakes, but they are not enough. Whenever the program is run, the client tests it! So, you must run the program before it arrives at the customer with the intention of finding and removing all errors. Look for the best array of achievable error audits should be carried out in a consistent manner, and test cases should be designed with disciplinary victimization techniques.

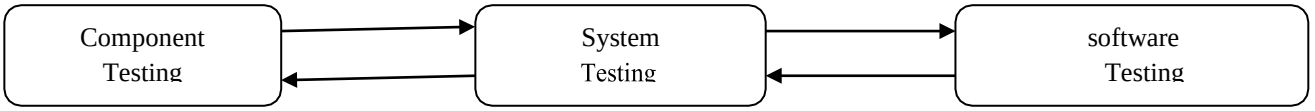


Fig: The Testing Process

Testing Objective:

- The test is the process by which a program is executed in order to find an error.
- A good test case has a high probability of finding a previously unknown mistake.
- A successful test reveals a mistake not yet discovered.

Unit Testing:

Unit verification is a software program machine enhancement technique whereby the smallest testable factors of an application, Devices, are personally scrutinized for proper operation. Verification unit is usually managed by machine, but it can moreover be executed manually. This check release can be a maximum programming detail, a reasonable method of improving the software package which takes a meticulous method to gather a product making use of persistent verification and revision.

Control of the unit completely includes traits that may be critical for the overall performance of the unit under one glance. This encourages the developer to switch the code at the same time as now no longer have instant problems regarding, how these settings may also affect the functioning of the devices or that system as an entire, Once all the features of an application operate in the maximum green manner and without blunders, possible, important factors of the system can be assessed via integration control. I reviewed each single a portion of the software; I reviewed each module personally. On the client-facet looked at modules such as the proper calculation of distance, opening-closing time, and others. In terms of hardware, I looked at modules such as the servomotor, ultrasonic sensors, and the LCD screen. As well, for each module, I ran the unit check at the same time as coding and earlier than depositing a demonstration. As result, the maximum number of errors was removed from the Device.

System Testing:

The software is tested on a comprehensive and integrated system to evaluate the many errors found in Arduino as the data works properly or not, I used the Arduino UNO board and worked on every error and exception I received during the tests and most of them are removed or corrected in such a way as not to happen again.

Another exception is when we use two ultrasound sensors, and the two sensors occasionally collide was not able to read the data from the external environment. These types of changes are made by me to make the system dependable and error-free.

Recovery Testing: This is a system test that forces the software to fail in various ways and checks that the recovery is carried out correctly.

Security Testing: This IoT device does not need to be secure as it operates offline.

Performance Testing: It is designed to test software performance in the context of an on-board system performance test that takes place at all stages within the testing process.

For the accuracy of data from Hardware, I used Waterfall Model testing.

It is also known as a linear-sequential life cycle version. It is straightforward to understand and use. In a waterfall version, each segment must be completed earlier than the next segment can begin and there can be no overlap inside the levels. The cascade version is the first SDLC method that has become used for software program enhancement. The waterfall model illustrates the software program or hardware enhancement method in a linear sequential stream. This method all segment within the improvement method starts better if the previous segment is complete. In this waterfall release, the levels do not overlap.

- **Information gathering** : All feasible requirements for the gadget to be advanced are captured in this section and documented in an application specification document.

- **System Design**: The requirements specifications of the main section are studied in this section and the layout of the gadget is prepared. This gadget layout makes it easier by specifying the hardware and gadget requires and makes it easier by defining the overall gadget architecture.

- **Implementation**: With inputs from the gadget layout, the gadget is first advanced in small applications referred to as devices, that are incorporated with the subsequent section. Each unit is advanced and reviewed for its functionality, which is referred to as Unit Testing.

- **Integration and Testing:** All advanced devices within the implementation section are integrated directly into a gadget after checking each unit. After integration the entire gadget is examined for all faults and malfunctions.

- **Deployment of the system**: When practical and impractical testing has been completed; the product is rolled out in the buyer's environment or placed on the market.

- **Maintenance**: Some problems arise inside the pattern's environment. To restore those problems, the manuals provide our facet for restoring that. Also to decorate the product some top variants are released. The service is completed to provide these settings within the buyer's frame.

Hardware Testing work Flow Diagram:

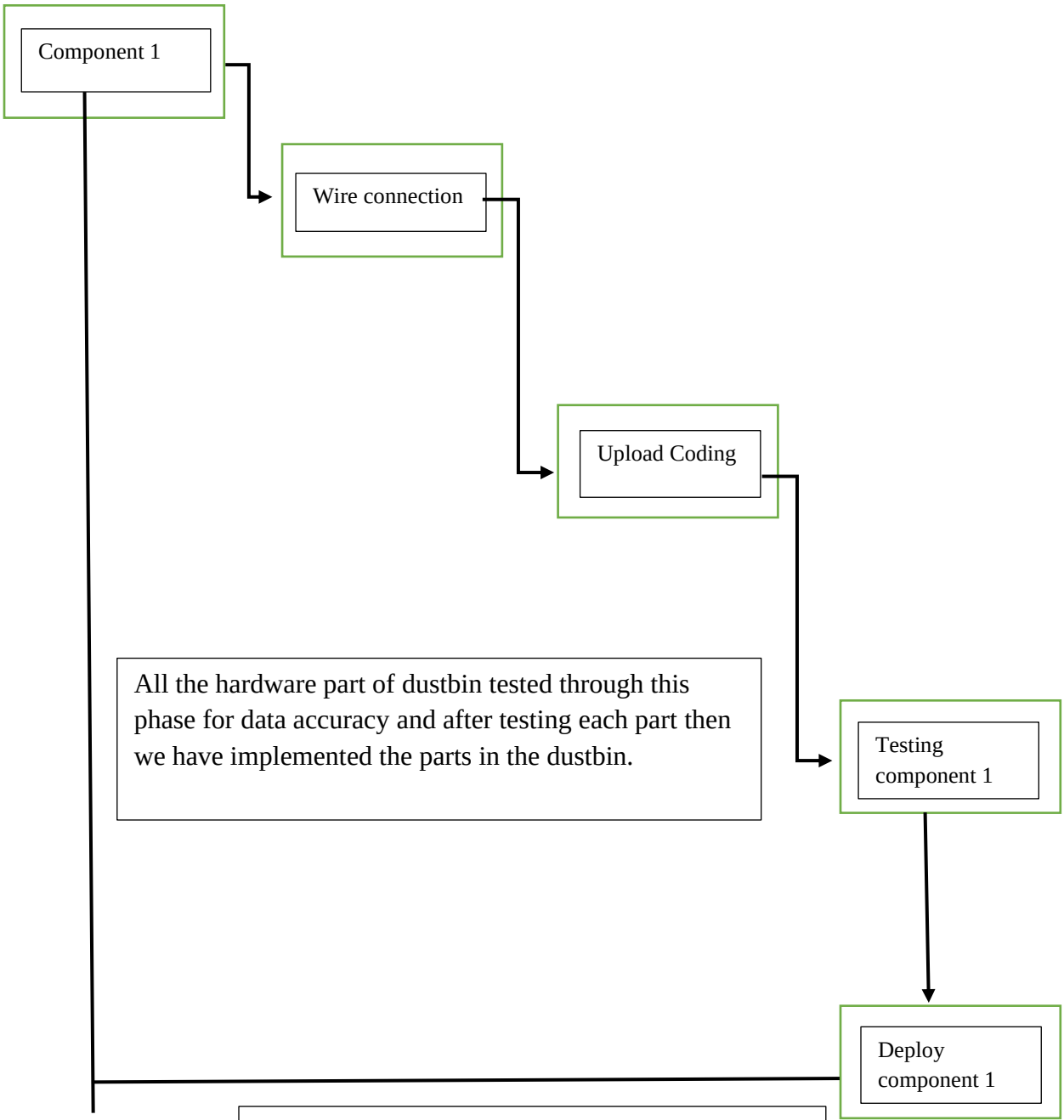


Fig: Dustbin Component Testing Phase

For system security we have checked the current flowing off the dustbin should not exceed 9 volts, because there is not any internet or database required for this.

For better connection and flowing of current in the dustbin, we have used good quality wires and for power supply, we have used a 5v adapter and a high data transfer USB cable.

Various Types of Reports & Modules

This device comprises 7 components:

1. Ultrasonic sensor
2. Arduino UNO Board
3. Servo Motor
4. LCD
5. Jumper Wires
6. USB cable
7. Power adapter

1. **Ultrasonic sensor:**

   This sensor allows users to put garbage in the dustbin by detecting the presence of any object.

2. **Arduino UNO Board:**

   This is the main component of the dustbin which contains code that will give the command to all the other components of the dustbin.

3. **Servo Motor:**

   This component is used to lift the lid for opening and closing of the dusting by moving 0 to 180 degrees and vice-versa.

4. **LCD:**

   This module is used to display the garbage level of the dustbin by calculating the distance inside the dustbin garbage and sensor distance.

5. **Jumper Wires:**

   This component is used to connect the whole sensor and other parts of the dustbin through the Arduino UNO board.

6. **USB cable:**

   This is used for high-rate data transfer to all parts of the dustbin.

7. **Power adapter:**

   It is used to power up the dustbin for its working.

Future Scope of Project

- **Features:**
  In the future, we'll add the remote-control wheels for an easier displacement of the dustbin and a Wi-Fi module that may send the garbage level of the dustbin to its mobile phone as a reminder.

- **Multi-Platform application:** Our project is presently Hardware and software-based. But after the implementation of the Wi-Fi module, this can be run from the client-side through an app.

Reference

- https://www.tutorialspoint.com/arduino/arduino_board_description.htm
- https://www.engineeringchoice.com/servo-motor/
- https://create.arduino.cc/projecthub/akshayjoseph666/interface-i2c-16x2-lcd-with-arduino-uno-just-4-wires-273b24
- https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor