SMART DUSTBIN FOR ECONOMIC GROWTH

Article · May 2017

CITATION
1

READS
115,514

1 author:

Nagaraju Urlagunta
VIT University
14 PUBLICATIONS **14** CITATIONS

SEE PROFILE

SMART DUSTBIN FOR ECONOMIC GROWTH

PROJECT REPORT

Submitted in fulfilment for the Component of ITE6004 – Internet of Things

CAL COURSE

in

M.Tech – CCE

by

U. NAGARAJU (16MCM0004)
RITU MISHRA (16MCM0003)
Chaitanya Kumar (16MCM0002)
Rajkumar (16MCM0001)

Under the guidance of

Prof. B.R. Kavitha,

Assistant Professor(Senior),
SITE

School of Information Technology and Engineering

Winter Semester 2016-17

TABLE OF CONTENTS

	References	

Abstract-

In the recent decades, Urbanization has increased tremendously. At the same phase there is an increase in waste production. Waste management has been a crucial issue to be considered. This paper is a way to achieve this good cause. In this paper, smart bin is built on a microcontroller based platform Arduino Uno board which is interfaced with GSM modem and Ultrasonic sensor. Ultrasonic sensor is placed at the top of the dustbin which will measure the stature of the dustbin. The threshold stature is set as 10cm. Arduino will be programmed in such a way that when the dustbin is being filled, the remaining height from the threshold height will be displayed. Once the garbage reaches the threshold level ultrasonic sensor will trigger the GSM modem which will continuously alert the required authority until the garbage in the dustbin is squashed. Once the dustbin is squashed, people can reuse the dustbin. At regular intervals dustbin will be squashed. Once these smart bins are implemented on a large scale, by replacing our traditional bins present today, waste can be managed efficiently as it avoids unnecessary lumping of wastes on roadside. Foul smell from these rotten wastes that remain untreated for a long time, due to negligence of authorities and carelessness of public may lead to long term problems.[1] Breeding of insects and mosquitoes can create nuisance around promoting unclean environment. This may even cause dreadful diseases.

1.) Problem Statement-

To implement a smart bin built on a microcontroller based platform Arduino Uno board which is interfaced with GSM modem and Ultrasonic sensor which can gives the status of the waste present in the dustbin to the municipal authority.

2) System design:

For communication purpose Bluetooth technology can also be used in the transmitter section. Bluetooth is a wireless networking standard that is aimed at remote control and sensor applications which is suitable for operation in harsh radio environments and in isolated locations. But, the main disadvantages of Bluetooth is short range, low complexity, and low data speed. Therefore, GSM is more advantages over Bluetooth for communication. Hence author use GS modem. A GSM modem is a specialized type wireless modem that works with a GSM wireless network. It accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. A GSM modem can be an external device or a PC Card / PCMCIA Card. An external GSM modem is connected to a computer through a serial cable or a USB cable. When a GSM modem is connected to a computer, this allows the computer to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile

internet connectivity, many of them can also be used for sending and receiving SMS and MMS message. GSM Modem sends and receives data through radio waves. In this project GSM 900 modem is used to send the messages which is shown in figure 2.[2] It consists of a GSM/GPRS modem with standard communication interfaces like RS-232 (Serial Port), USB, so that it can be easily connected to the other devices. The power supply circuit is also built in the module that can be turn ON by using a suitable adaptor.

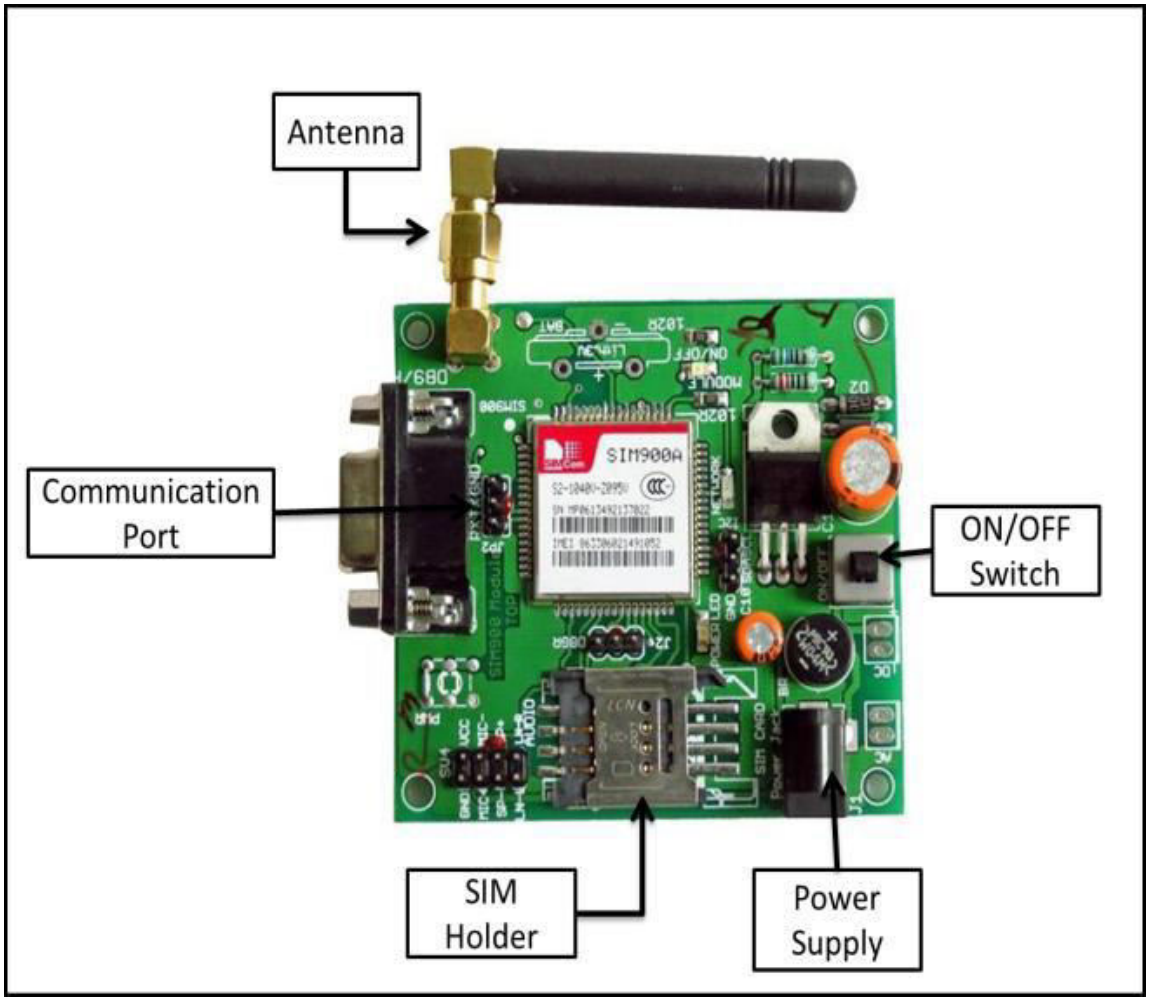


Fig 1: GSM

The block diagram of transmitter section. Level detector consists of IR sensors which is used to detect the level of the garbage in the dustbin. The output of level detector is given to microcontroller. Four IR sensors are used to indicate the different levels of the amount of the garbage collected in the dustbin which is placed in public area. When the dustbin is filled up to the highest level, the output of fourth IR receiver becomes active low. This output is given to microcontroller to send the message to the Control room via GSM module as shown in below.

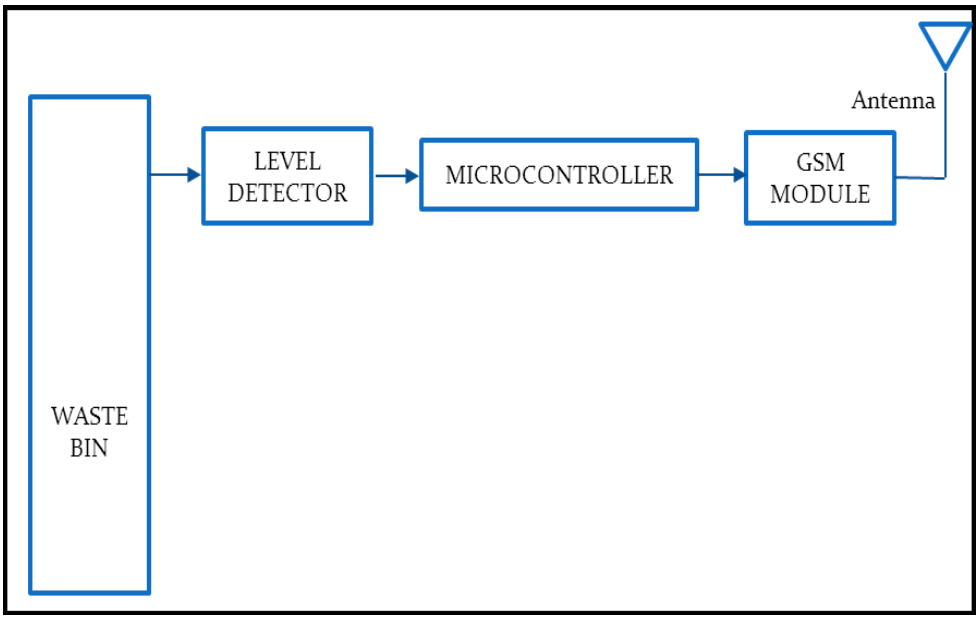


Fig-2 Transmitter Part from Dustbin

The figure 4 shows the block diagram of receiver section. At receiver, control room is present where all the activities are managing. The number of the control room is depending on the dustbins present in the area. The person sitting in the control room monitors the entire system. A GSM Module is connected to the computer of the control room through microcontroller. The entire system is monitor by the person sitting in the control room. The same GSM Module is used to send the message to the contractor for cleaning the dustbin. GUI is developed using MATLAB software. This GUI will be displayed on the computer screen in the control room to display the status of the garbage level in the dust bin as shown in below.

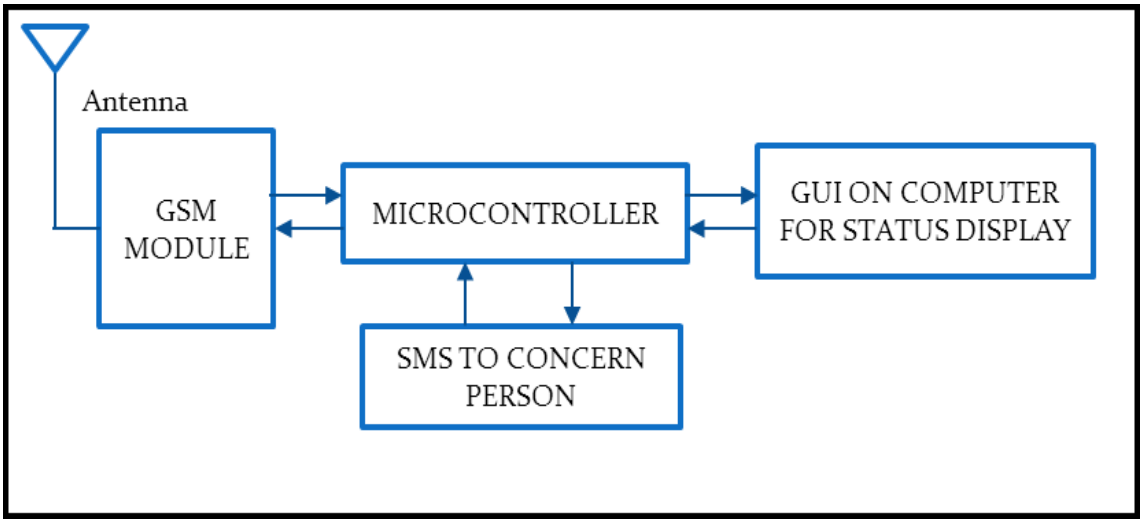


Fig-3: Reception Part to Dustbin

2.1 Flow chat of Woking Principle:

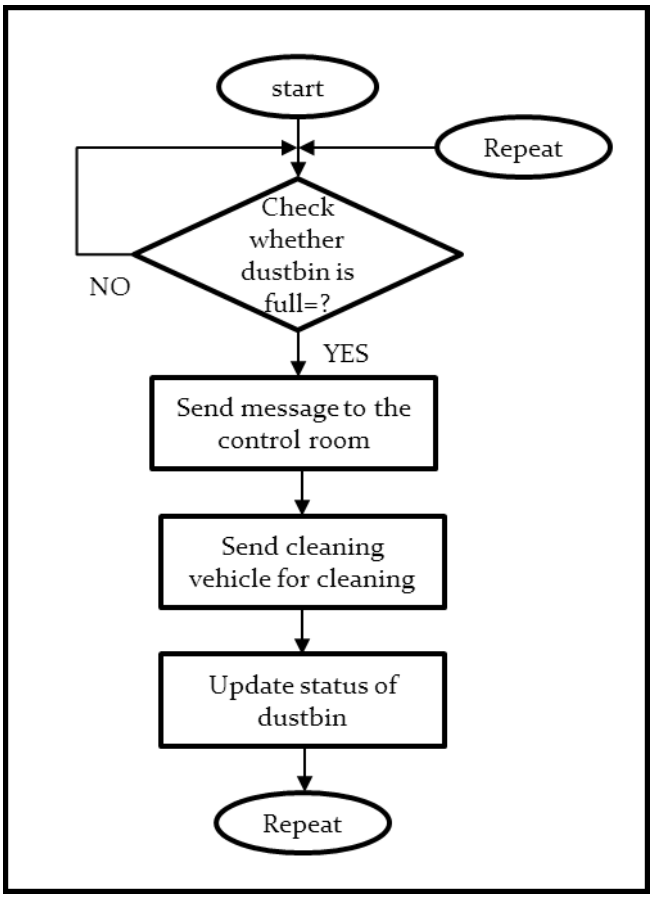
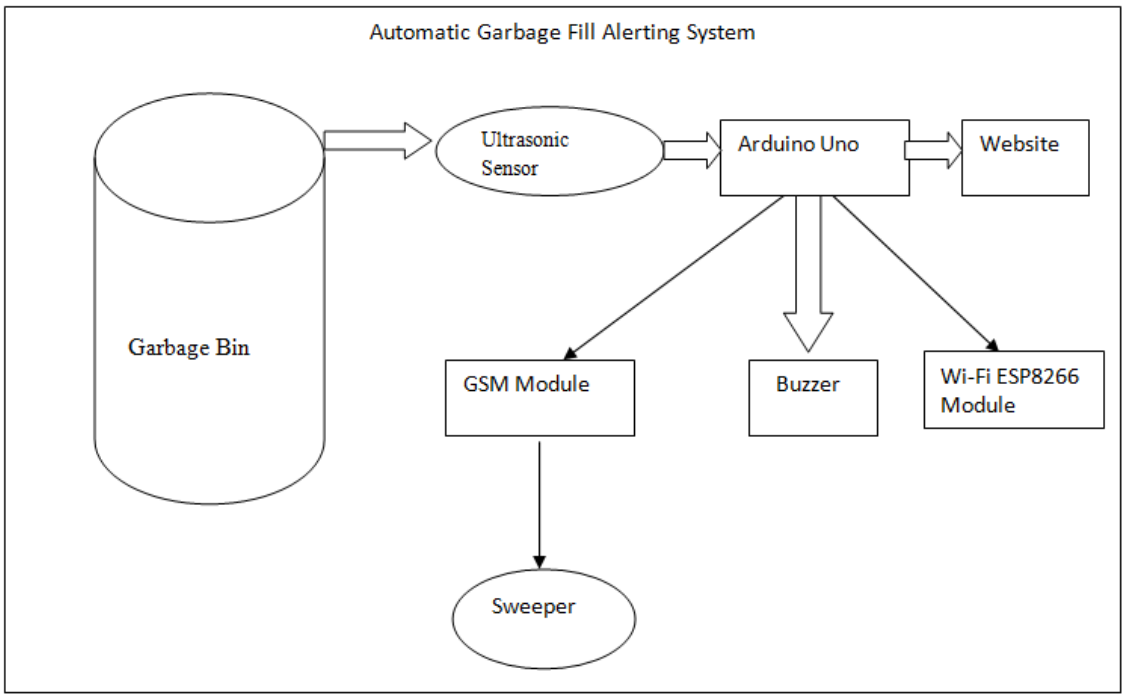


Fig 4: Flowchart

The PIR sensor will observe a person nearby dustbin. If motion is detected the lid of dustbin is opened, the servo motor activates and as GSM connected it will send an alert message to user if dustbin is filled.

Dustbin placed in public place, people throw garbage in dustbin, place the ultrasonic sensor in top of the garbage bin. If dustbin reach in 75% then arduino send message through GSM module. When dustbin level is reach threshold level buzzer will give alert sound for don't again put waste in dustbin. This all process updated in IOT GECKO platform for monitoring garbage bin.



Working of Line Follower Robot using Arduino Working of line follower is very interesting. Line follower robot senses black line by using sensor and then sends the signal to Arduino. Then Arduino drives the motor according to sensors' output. Working of line follower Here in this project we are using two IR sensor modules namely left sensor and right sensor. When both left and right sensor senses white then robot move forward.

2.2 Literature survey:

The authors in [4] have made a quantitative analysis between existing dustbins and their serving population. The study first analyses the spatial distribution of dustbins in some areas of Dhaka city using average nearest neighbor functions of GIS. Remarkably, the spatial circulation of the current dustbins has appeared to be dominatingly in clustered pattern. Next, an optimal number of additional dustbins were calculated. It is shown that the number of existing dustbins is insufficient in the study area. The extent of pollution caused by the existing dustbins was calculated using spatial analyst functions of GIS. It is found that all the dustbins are burnt with wastes and causing pollution to the environment. The results thus obtained would help to understand the present situation of the waste management of Research Article Volume 6 Issue No. 6 International Journal of Engineering Science and Computing, June 2016 7114 http://ijesc.org/ Dhaka city and to optimally place the required number of dustbins to prevent further pollution to environment.

The authors in [5] have equipped the smart bins with ultrasonic sensors which measure the level of dustbin being filled up. The container is divided into three levels of garbage being collected in it. Every time the garbage crosses a level the sensors receives the data of the filled level. This data is further sent to the garbage analyzer as instant message using GSM module. Placing three ultrasonic sensors at three different levels of the container may be a disadvantage as the cost of the dustbin increases due to the sensors and also the sensors can be damaged due to the rough action by the users. An IoT-based smart garbage system (SGS) is proposed to reduce the amount of food waste

The authors in [6]. In an SGS, battery-based smart garbage bins (SGBs) exchange information with each other using wireless mesh networks, and a router and server collect and analyze the information for service provisioning. Furthermore, the SGS includes various IoT skills considering user convenience and increases the battery lifetime through two types of energy-efficient operations of the SGBs: stand-alone operation and cooperation based operation. The proposed SGS had been functioned as a pilot project in Gangnam district, Seoul, Republic of Korea, for a one-year period. The test demonstrated that the normal measure of food waste could be decreased by 33%.

The authors in [7] has built a framework in which a Camera will be set at each garbage collection point alongside load cell sensor at base of the trash can. The camera will take continuous snapshots of the garbage can. A threshold level is set which compares the output of camera and load sensor. The comparison is done with help of microcontroller. After analyzing the image an idea about level of garbage in the can and from the load cell sensor, weight of garbage can be known. Accordingly, information is processed that is controller checks if the threshold level is exceeded or not. This is convenient to use but economically not reliable

3.) Requirement Specification-

Smart bin is built on Arduino board platform. It is interfaced with a GSM modem (SIM 900A) and the bin is equipped with Ultrasonic sensor (HC-SR04) and PIR sensor.

3.1) Components Used-

1.) Arduino

2.) GSM

3.) PIR Sensor

4.) Ultrasonic Sensor

5.) Servo Motor

6.) Breadboard

7.) Connecting wires

1.) Arduino-

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.[3]

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). 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.

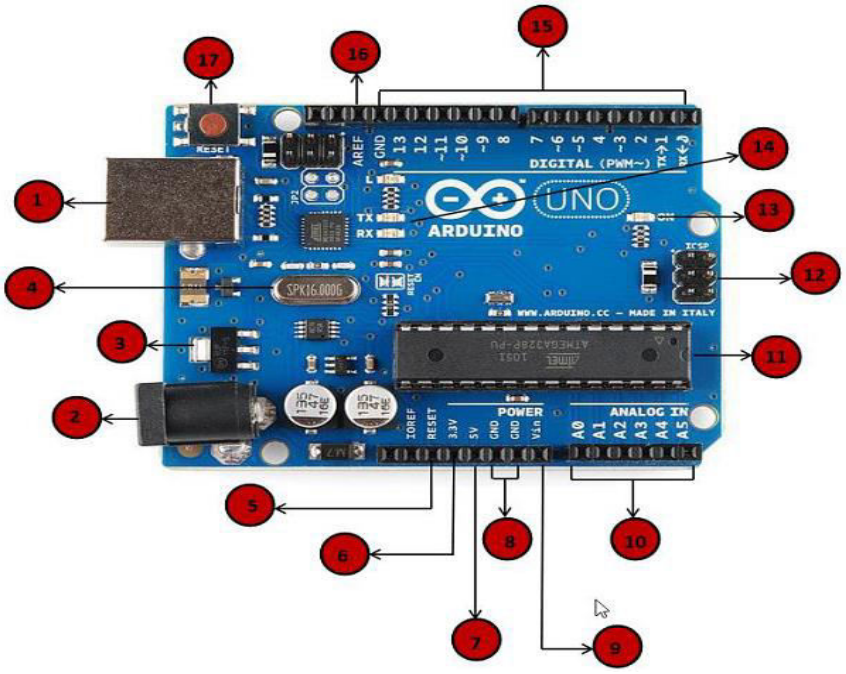


Fig 5: Pin diagram of Arduino

(i) Power USB- Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection

(ii) Power (Barrel Jack)

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

(iii) Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

(iv) Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

(v, xvii) Arduino 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, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

(vi, vii, viii, ix) Pins (3.3, 5, GND, Vin)

- 3.3V (6) − Supply 3.3 output volt
- 5V (7) − Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
- 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 also can be used to power the Arduino board from an external power source, like AC mains power supply.

(x) Analog pins- The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

(xi) Main microcontroller- Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

(xii) ICSP pin- Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

(xiii) Power LED indicator- This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

(xiv) TX and RX LEDs- On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

(xv) Digital I/O- The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled can be used to generate PWM.

(xvi) AREF- AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Microcontroller- ATmega2560

Operating Voltage -5V

Input Voltage (recommended)- 7-12V

Input Voltage (limit)- 6-20V

Digital I/O Pins 54 (of which 15 provide PWM output)

Analog Input 16

2.) Gsm (Global System for Mobile Communication)-

GSM Modem can accept any GSM network operator SIM card and act just like a mobile phone with its own unique phone number. Advantage of using this modem will be that you can use its RS232 port to communicate and develop embedded applications. Applications like SMS Control, data transfer, remote control and logging can be developed easily using gsm. The modem can either be connected to PC serial port

directly or to any microcontroller through MAX232. It can be used to send and receive SMS or make/receive voice calls. It can also be used in GPRS mode to connect to internet and do many applications for data logging and control. In GPRS mode you can also connect to any remote FTP server and upload files for data logging. This GSM modem is a highly flexible plug and play quad band SIM900A GSM modem for direct and easy integration to RS232 applications. Supports features like Voice, SMS, Data/Fax, GPRS and integrated TCP/IP stack.

A GSM network comprises of many functional units. These functions and interfaces are explained in this chapter. The GSM network can be broadly divided into:

- The Mobile Station (MS)
- The Base Station Subsystem (BSS)
- The Network Switching Subsystem (NSS)
- The Operation Support Subsystem (OSS)

Given below is a simple pictorial view of the GSM architecture.

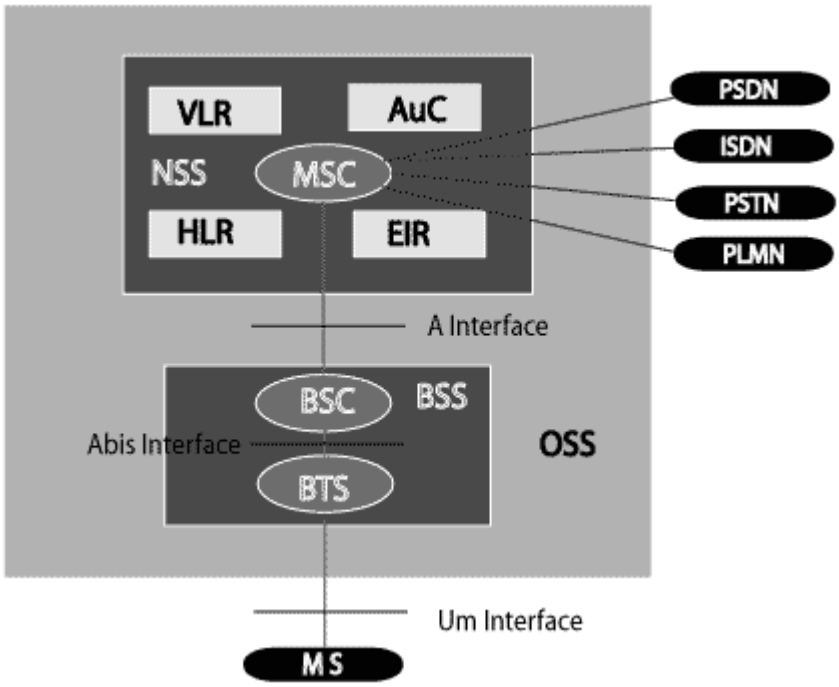


Fig 6: Architecture of GSM

The additional components of the GSM architecture comprise of databases and messaging systems functions:

- Home Location Register (HLR)
- Visitor Location Register (VLR)
- Equipment Identity Register (EIR)
- Authentication Centre (AuC)
- SMS Serving Centre (SMS SC)
- Gateway MSC (GMSC)
- Chargeback Centre (CBC)
- Transcoder and Adaptation Unit (TRAU)

The following diagram shows the GSM network along with the added elements:

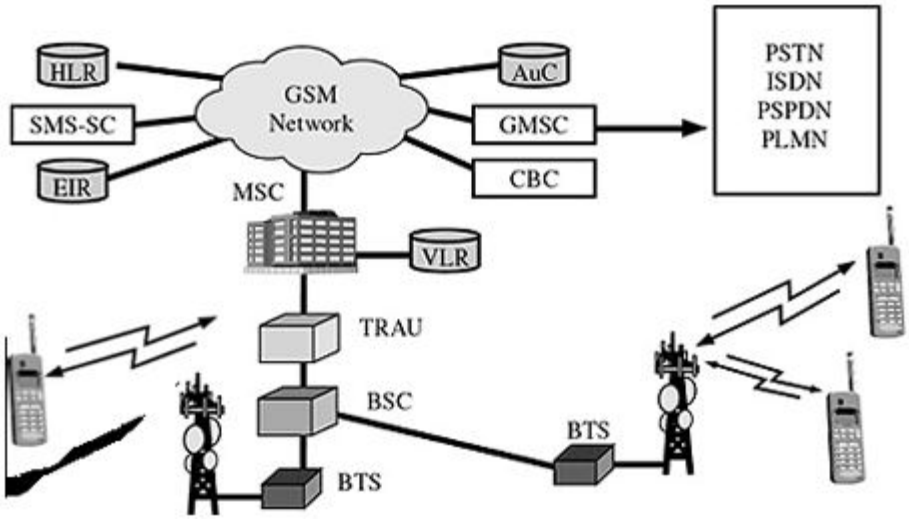


Fig 7: GSM Network architecture

The MS and the BSS communicate across the Um interface. It is also known as the *air interface* or the *radio link*. The BSS communicates with the Network Service Switching (NSS) centre across the *A* interface.

GSM network areas

In a GSM network, the following areas are defined:

- **Cell**: Cell is the basic service area; one BTS covers one cell. Each cell is given a Cell Global Identity (CGI), a number that uniquely identifies the cell.

- **Location Area**: A group of cells form a Location Area (LA). This is the area that is paged when a subscriber gets an incoming call. Each LA is assigned a Location Area Identity (LAI). Each LA is served by one or more BSCs.

- **MSC/VLR Service Area**: The area covered by one MSC is called the MSC/VLR service area.

- **PLMN**: The area covered by one network operator is called the Public Land Mobile Network (PLMN). A PLMN can contain one or more MSCs.

3.) PIR Sensor

PIR sensors allow you to sense motion. They are used to detect whether a human has moved in or out of the sensor's range. They are commonly found in appliances and gadgets used at home or for businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

Following are the advantages of PIR Sensors −

- Small in size
- Wide lens range
- Easy to interface
- Inexpensive
- Low-power
- Easy to use
- Do not wear out

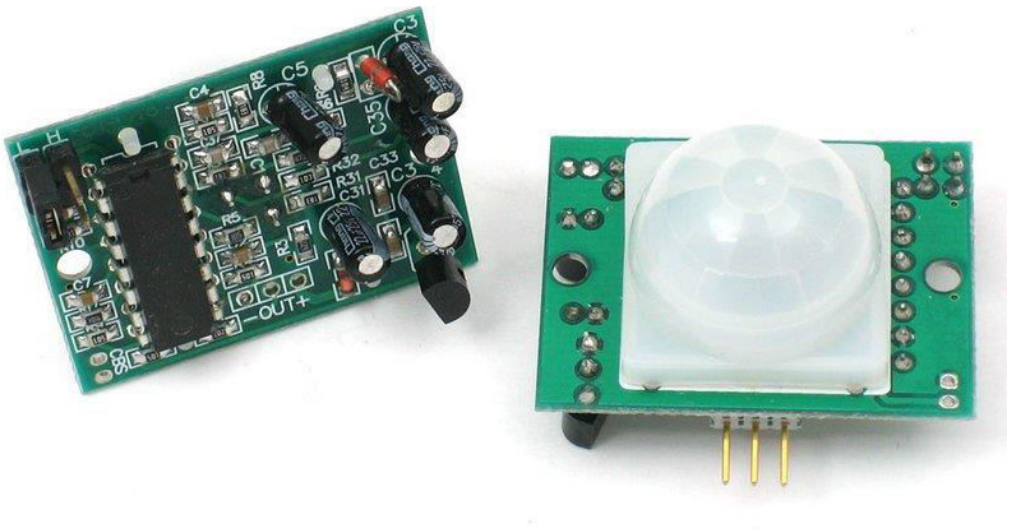


Fig 8: PIR sensor

PIRs are made of pyroelectric sensors, a round metal can with a rectangular crystal in the centre, which can detect levels of infrared radiation. Everything emits low-level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is split in two halves.[4] This is to detect motion (change) and not average IR levels. The two halves are connected so that they cancel out each other. If one-half sees more or less IR radiation than the other, the output will swing high or low.

PIRs have adjustable settings and have a header installed in the 3-pin ground/out/power pads.

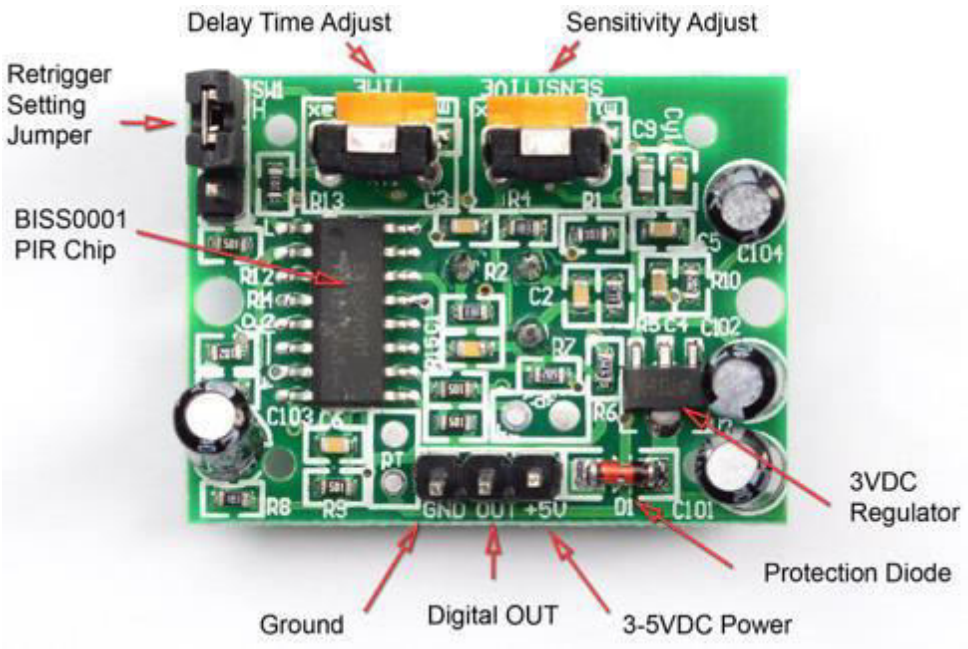
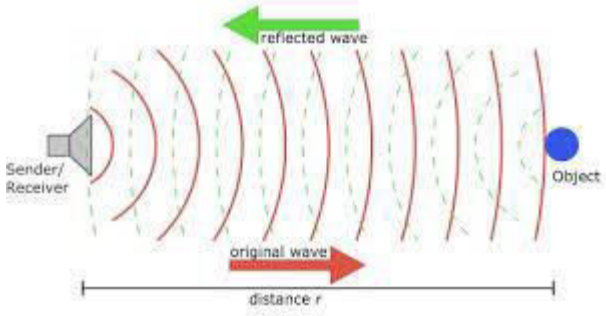


Fig 9: PIR Sensor Diagram

4.) ULTRASONIC SENSOR-

The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1" to 13 feet. The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module.



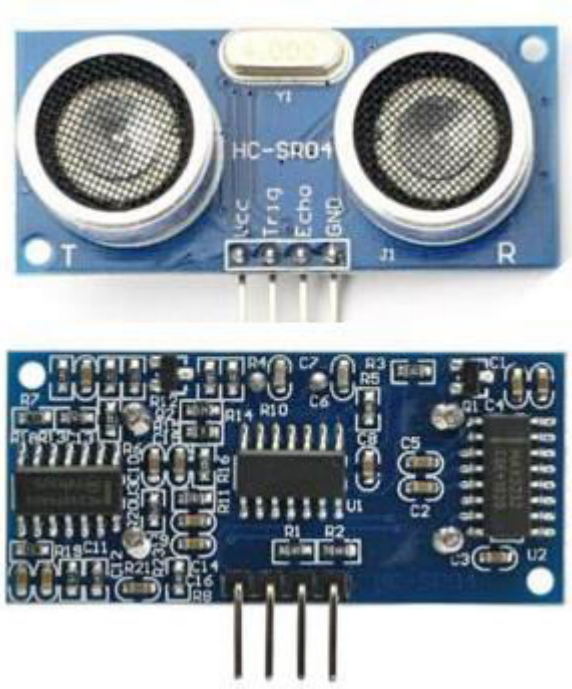


Fig 10: Ultra sonic sensor

Technical Specifications

- Power Supply − +5V DC
- Quiescent Current − <2mA
- Working Current − 15mA
- Effectual Angle − <15°
- Ranging Distance − 2cm – 400 cm/1″ – 13ft
- Resolution − 0.3 cm
- Measuring Angle − 30 degree

5.) Servo Motor-

A Servo Motor is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. If the coded signal changes, the angular position of the shaft changes. [8] In practice, servos are used in radio-controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio-controlled cars, puppets, and of course, robots.

Fig 11: Servo Motor

Servos are extremely useful in robotics. The motors are small, have built-in control circuitry, and are extremely powerful for their size. A standard servo such as the Futaba S-148 has 42 oz/inches of torque, which is strong for its size. It also draws power proportional to the mechanical load. A lightly loaded servo, therefore, does not consume much energy. The guts of a servo motor is shown in the following picture. You can see the control circuitry, the motor, a set of gears, and the case. You can also see the 3 wires that connect to the outside world. One is for power (+5volts), ground, and the white wire is the control wire.

Fig 12: Servo motor parts

Working of a Servo Motor

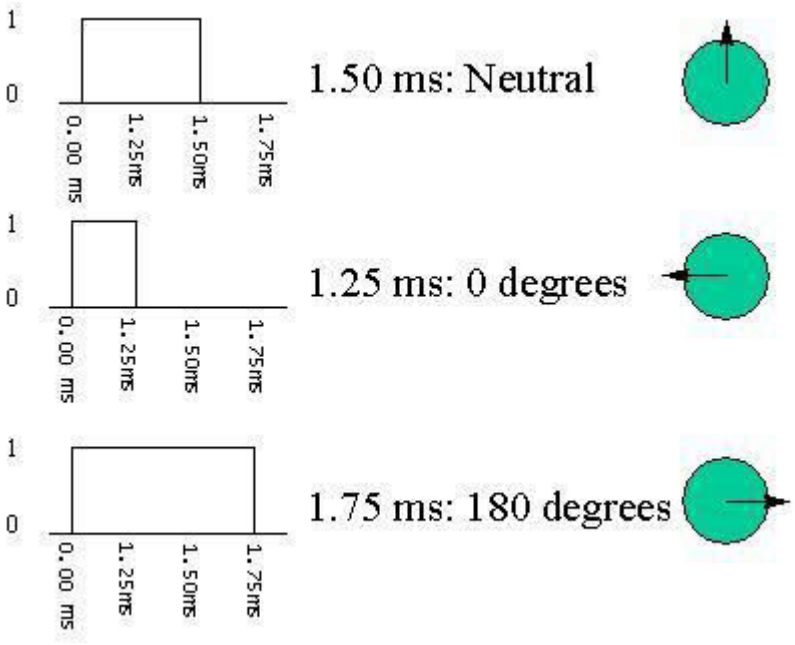
The servo motor has some control circuits and a potentiometer (a variable resistor, aka pot) connected to the output shaft. In the picture above, the pot can be seen on the right side of the circuit board. This pot allows the control circuitry to monitor the current angle of the servo motor.

If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor until it is at a desired angle. The output shaft of the servo is capable of traveling somewhere around 180 degrees. Usually, it is somewhere in the 210-degree range, however, it varies depending on the manufacturer. A normal servo is used to control an angular motion of 0 to 180 degrees. It is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear.

The power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called **proportional control**.

How Do You Communicate the Angle at Which the Servo Should Turn?

The control wire is used to communicate the angle. The angle is determined by the duration of a pulse that is applied to the control wire. This is called **Pulse Coded Modulation**. The servo expects to see a pulse every 20 milliseconds (.02 seconds). The length of the pulse will determine how far the motor turns. A 1.5 millisecond pulse, for example, will make the motor turn to the 90-degree position (often called as the neutral position). If the pulse is shorter than 1.5 milliseconds, then the motor will turn the shaft closer to 0 degrees. If the pulse is longer than 1.5 milliseconds, the shaft turns closer to 180 degrees.



6.) Breadboard-

A breadboard is a construction base for prototyping of electronics. Originally it was literally a bread board, a polished piece of wood used for slicing bread. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also extremely popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Vero board) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

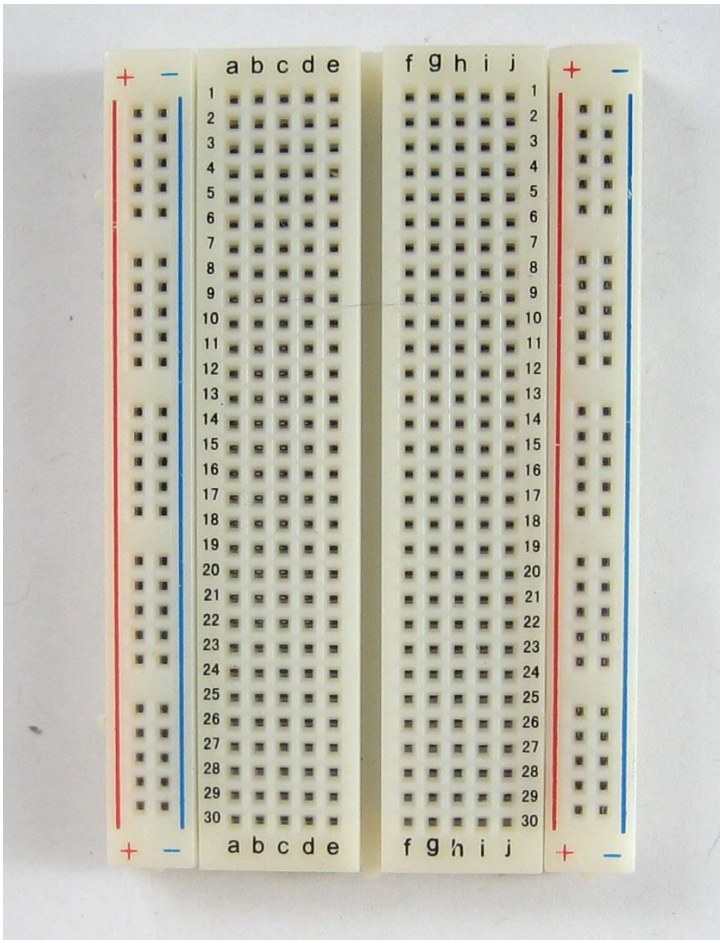


Fig 13: Breadboard

4) Implementation

4.1) Source code:

4.1.1.) PIR Code for detecting a person near dustbin:

```
int led = 13;                 // the pin that the LED is attached to

int sensor = 2;               // the pin that the sensor is attached to

int state = LOW;              // by default, no motion detected

int val = 0;                  // variable to store the sensor status (value)
```

```
void setup() {
  pinMode(led, OUTPUT);          // initalize LED as an output
  pinMode(sensor, INPUT);       // initialize sensor as an input
  Serial.begin(9600);             // initialize serial
}


void loop(){
  val = digitalRead(sensor);     // read sensor value
  if (val == HIGH) {                 // check if the sensor is HIGH
    digitalWrite(led, HIGH);     // turn LED ON
    delay(100);                         // delay 100 milliseconds


    if (state == LOW) {
      Serial.println("Motion detected!");
      state = HIGH;           // update variable state to HIGH
    }
  }
  else {
      digitalWrite(led, LOW); // turn LED OFF
```

```
        delay(200);                    // delay 200 milliseconds


        if (state == HIGH){
           Serial.println("Motion stopped!");
           state = LOW;           // update variable state to LOW
      }
   }
}
```

4.1.2.) Servo Code to open LID of Dustbin:

```
#include <Servo.h>           //Servo library
 Servo name_servo;          //initialize a servo object for the connected servo
   int servo_position =0;
 void setup()
 {
  name_servo.attach(9);     // attach the signal pin of servo to pin9 of arduino
 }
 void loop()
 {
  for (servo_position=0;servo_position<=180;servo_position +=1){
     name_servo.write(servo_position);
```

```
    delay(10);
  }
  for(servo_position=180; servo_position>=0;servo_position-=1)
  {
    name_servo.write(servo_position);
    }
delay(10);
 }
```

4.1.3.) Ultra sonic Code to measure the dustbin height:

```
/*
 HC-SR04 Ping distance sensor:
 VCC to arduino 5v
 GND to arduino GND
 Echo to Arduino pin 7
 Trig to Arduino pin 8
*/
#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin
#define LEDPin 13 // Onboard LED
int maximumRange = 200; // Maximum range needed
```

```
int minimumRange = 0; // Minimum range needed
long duration, distance; // Duration used to calculate distance


void setup() {
 Serial.begin (9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
}
void loop() {
/* The following trigPin/echoPin cycle is used to determine the
 distance of the nearest object by bouncing soundwaves off of it. */
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);
 //Calculate the distance (in cm) based on the speed of sound.
 distance = duration/58.2;
```

```
 if (distance >= maximumRange || distance <= minimumRange){
 /* Send a negative number to computer and Turn LED ON
 to indicate "out of range" */
 Serial.println("-1");
 digitalWrite(LEDPin, HIGH);
 }
 else {
 /* Send the distance to the computer using Serial protocol, and
 turn LED OFF to indicate successful reading. */
 Serial.println("Dustbin is at distance of");
 Serial.println(distance);
 digitalWrite(LEDPin, LOW);
 }
 //Delay 50ms before next reading.
 delay(50);
}
```

4.1.4.) GSM code to send Message to use when it is filled:

```
#include "SIM900.h"
#include <SoftwareSerial.h>
//#include "inetGSM.h"
```

```
#include "sms.h"

SMSGSM sms;

int numdata;

int value;

boolean started=false;

char smsbuffer[160];

char n[20];

int PIN = A0;

int sensor = 450;

int LED1 = 11;

void setup()

{

 pinMode(LED1, OUTPUT);

   pinMode (PIN,INPUT);

   value = analogRead (PIN);

   Serial.begin(9600);

   Serial.println(value);

   Serial.println("GSM Shield testing.");
```

```
 if (value > sensor){
    digitalWrite(LED1, HIGH);
 }
  if (gsm.begin(2400)){
    Serial.println("\nstatus=READY");
    started=true;
  }
  else Serial.println("\nstatus=IDLE");
  if (value > sensor){
    digitalWrite(LED1, HIGH);
    (sms.SendSMS("+917530001174", "RJ27 CB 3**9 Pollution Level Exceeding , Attention Required "));
       Serial.println("\nSMS sent OK");
  }


};
void loop()
{
  if(started){

    if(gsm.readSMS(smsbuffer, 160, n, 20))
```

```
    {
      Serial.println(n);
      Serial.println(smsbuffer);
    }
  }
};
```

4.1.5.) Finally intergrated code of project:

```
#include "SIM900.h"
#include <SoftwareSerial.h>
//#include "inetGSM.h"
#include "sms.h"
SMSGSM sms;
int numdata;
int value;
boolean started=false;
char smsbuffer[160];
char n[20];
int PIN = A0;
int sensor = 450;
int LED1 = 11;
```

```
#include <Servo.h>

Servo myservo;   //creates a servo object

                        //a maximum of eight servo objects can be created

  int pos = 0;          //variable to store servo position

//amount of time we give the sensor to calibrate(10-60 secs according to the datasheet)

int calibrationTime = 30;//the time when the sensor outputs a low impulse

long unsigned int lowIn;        //the amount of milliseconds the sensor has to be low
//before we assume all motion has stopped

long unsigned int pause = 5000;

boolean lockLow = true;

boolean takeLowTime;

int pirPin = 12;                //digital pin connected to the PIR's output

int pirPos = 13;                //connects to the PIR's 5V pin

void setup(){

 pinMode(LED1, OUTPUT);

 pinMode (PIN,INPUT);

 value = analogRead (PIN);

  Serial.begin(9600);

  Serial.println(value);

  Serial.println("GSM Shield testing.");
```

```
 if (value > sensor){
    digitalWrite(LED1, HIGH);
 }
  if (gsm.begin(2400)){
    Serial.println("\nstatus=READY");
    started=true;
  }
  else Serial.println("\nstatus=IDLE");
  if (value > sensor){
    digitalWrite(LED1, HIGH);
  (sms.SendSMS("+917530001174", "RJ27 CB 3**9 Pollution Level Exceeding , Attention Required "));
       Serial.println("\nSMS sent OK");
  }
  Serial.begin(9600);      //begins serial communication
  pinMode(pirPin, INPUT);
  pinMode(pirPos, OUTPUT);
  digitalWrite(pirPos, HIGH);

  //give the sensor time to calibrate
  Serial.println("calibrating sensor ");
```

```
  for(int i = 0; i < calibrationTime; i++){
    Serial.print(calibrationTime - i);
    Serial.print("-");
    delay(1000);
  }
  Serial.println();
  Serial.println("done");

  //while making this Instructable, I had some issues with the PIR's output
  //going HIGH immediately after calibrating
  //this waits until the PIR's output is low before ending setup
  while (digitalRead(pirPin) == HIGH) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("SENSOR ACTIVE");
}

void loop(){
  if(started){
```

```
      if(gsm.readSMS(smsbuffer, 160, n, 20))
      {
        Serial.println(n);
        Serial.println(smsbuffer);
      }
  }
if(digitalRead(pirPin) == HIGH){   //if the PIR output is HIGH, turn servo
      /*turns servo from 0 to 180 degrees and back
      it does this by increasing the variable "pos" by 1 every 5 milliseconds until it hits
  180
      and setting the servo's position in degrees to "pos" every 5 milliseconds
      it then does it in reverse to have it go back
      to learn more about this, google "for loops"
      to change the amount of degrees the servo turns, change the number 180 to the
  number of degrees you want it to turn
      **/
      for(pos = 0; pos < 180; pos += 1)   //goes from 0 to 180 degrees
      {                                                    //in steps of one degree
        myservo.write(pos);                          //tells servo to go to position in variable
  "pos"
        delay(5);                                          //waits for the servo to reach the
  position
```

```
    }
    for(pos = 180; pos>=1; pos-=1)      //goes from 180 to 0 degrees
    {
      myservo.write(pos);                    //to make the servo go faster, decrease
the time in delays for
      delay(5);                                   //to make it go slower, increase the
number.
    }
    if(lockLow){
      //makes sure we wait for a transition to LOW before further output is made
      lockLow = false;
      Serial.println("---");
      Serial.print("motion detected at ");
      Serial.print(millis()/1000);
      Serial.println(" sec");
      delay(50);
    }
    takeLowTime = true;
  }

  if(digitalRead(pirPin) == LOW){
```

```
      if(takeLowTime){

        lowIn = millis();                //save the time of the transition from HIGH to
LOW

        takeLowTime = false;      //make sure this is only done at the start of a LOW
phase

      }

      //if the sensor is low for more than the given pause,

      //we can assume the motion has stopped

      if(!lockLow && millis() - lowIn > pause){

        //makes sure this block of code is only executed again after

        //a new motion sequence has been detected

        lockLow = true;

        Serial.print("motion ended at "); //output

        Serial.print((millis() - pause)/1000);

        Serial.println(" sec");

        delay(50);

      }   }

}
```

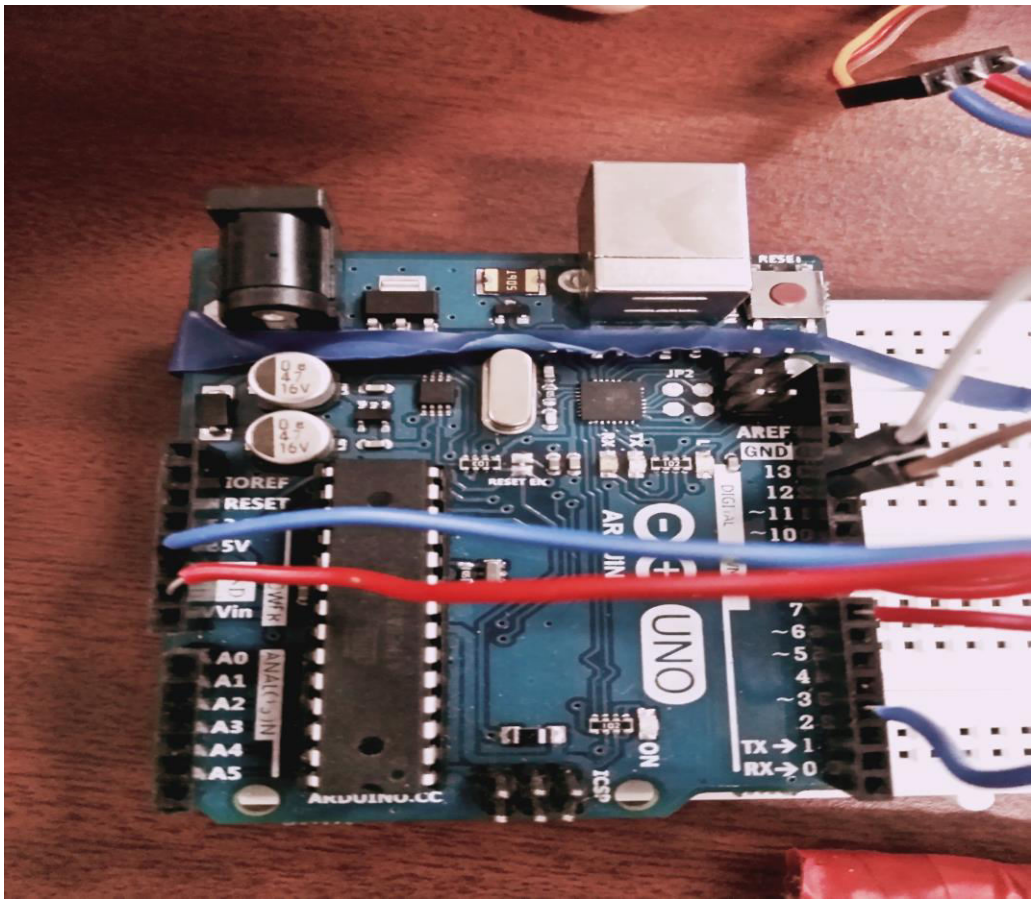
5.) Conclusion:

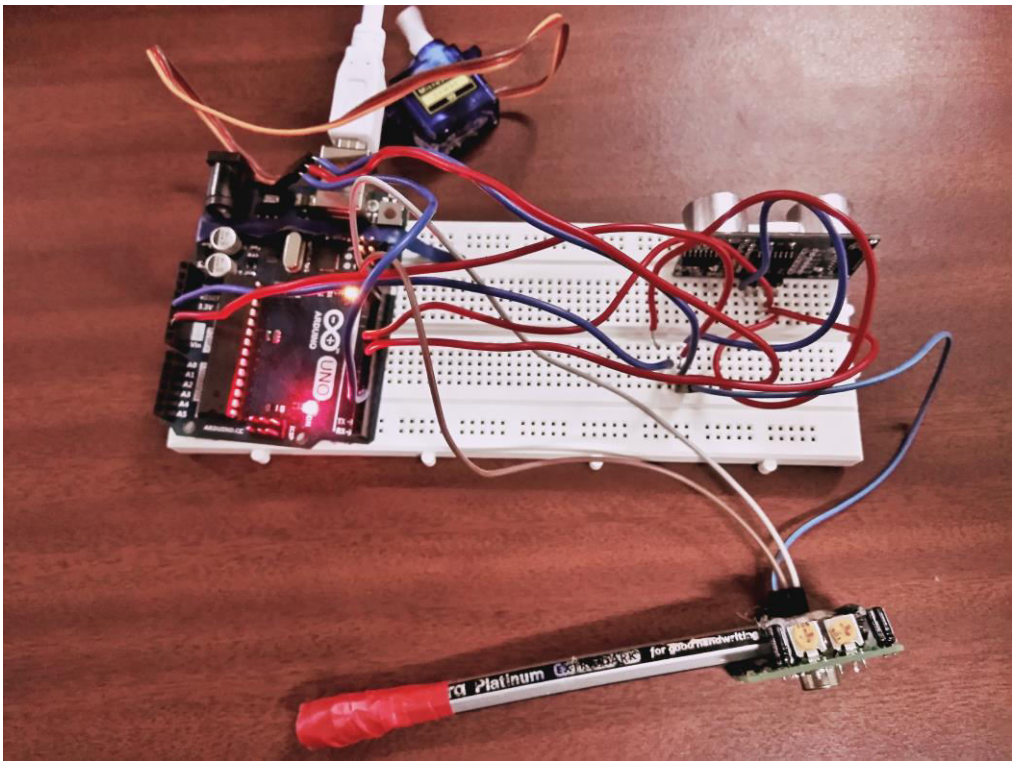
This project work is the implementation of Automatic Garbage Fill Alerting system using Ultrasonic sensor, Arduino Uno, Buzzer and Wi-Fi module. This system assures the cleaning of dustbins soon when the garbage level reaches its maximum. It will take power supply with the help of Piezoelectric Device .If the dustbin is not cleaned in specific time,

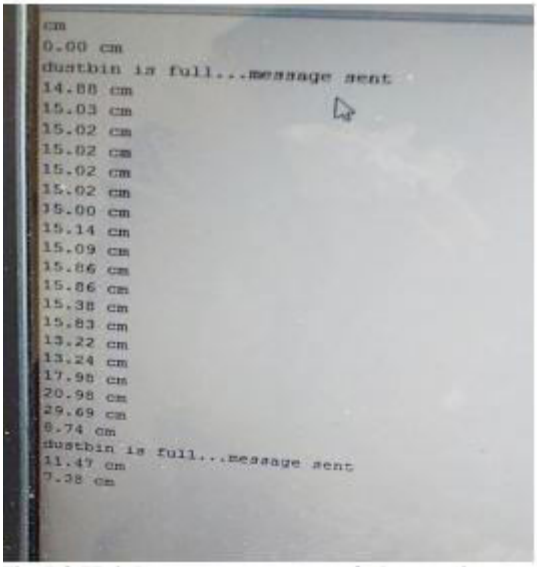
then the record is sent to the Sweeper or higher authority who can take appropriate action against the concerned contractor. This system also helps to monitor the fake reports and hence can reduce the corruption in the overall management system. This reduces the total number of trips of garbage collection vehicle and hence reduces the overall expenditure associated with the garbage collection. It ultimately helps to keep cleanliness in the society. Therefore, the Automatic Garbage Fill Alerting system makes the garbage collection more efficient.

6.) Test Result-

The dustbin is able to open the lid with the help of servo motor and PIR sensor whenever it detects motion. The ultrasonic sensor is giving the details about the waste present in the dustbin. The status of the waste is transferred to the municipal authority whenever it is exceeding the threshold value.







7.) Future Enhancement:

Automatic Garbage Fill Alerting system helps us to reduce the pollution. Many times garbage dustbin is overflow and many animals like dog or cow enters inside or near the dustbin. Also some birds are also trying to take out garbage from dustbin. This project can avoid such situations. And the message can be sent directly to the cleaning vehicle instead of the contractor's office. Apart from this, differentiation can be made between dry trash

bin and wet trash bin collecting plastic dry waste and biodegradable waste respectively. To implement this methane and smell sensors can be used. This helps in distinguishing the waste at the source and hence reducing the requirement of manpower. To enhance it further, an automated system can be developed which is able to pick up waste in and around the bin, segregate them and put them in respective bins.

8.) References

[1]. S.S. Navghane, M.S. Killedar, Dr.V.M. Rohokale,‖ IoT Based Garbage and Waste Collection Bin‖, May 2016.

[2]. Ghose, M.K., Dikshit, A.K., Sharma, S.K. A GIS based transportation model for solid waste disposal – A case study on Asansol municipality. Journal of Waste Management‖.

[3]. Guerrero, L.A., Maas, G., Hogland, W.: Solid waste management challenges for cities in developing countries. Journal of Waste Management.

[4]. Alexey Medvedev, Petr Fedchenkov, ArkadyZaslavsky, Theodoros, Anagnostopoulos Sergey Khoruzhnikov,‖Waste Management as an IoT-Enabled Service in Smart Cities‖.

[5]. Meghana K C, Dr. K R Nataraj,‖ IOT Based Intelligent Bin for Smart Cities‖.

[6]. KasliwalManasi H., SuryawanshiSmitkumar B, A Novel Approach to Garbage Management Using Internet of Things for Smart Cities‖.

[7]. Vishesh Kumar Kurrel,‖ Smart Garbage Collection Bin Overflows Indicator using Internet of Things

[8]. Monika K A, Nikitha Rao, Prapulla S B, Shobha G, Smart Dustbin-An Efficient Garbage Monitoring System.