

PROGRAM CODE:

```
#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)
    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)
    StartTime = time.time()
    StopTime = time.time()
    # save StartTime
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()
    # save time of arrival
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()
    # time difference between start and arrival
    TimeElapsed = StopTime - StartTime
    # multiply with the sonic speed (34300 cm/s)
    # and divide by 2, because there and back
    distance = (TimeElapsed * 34300) / 2
    return distance
23
if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print ("Measured Distance = %.1f cm" % dist)
            percent = (100.0 - (dist * 100/40.0))
            url =
            "http://localhost:80/demoaddbin.php?bin_id=1&percent_filled="+str(percent)
            x= urllib.urlopen(url)
            print(x.read)
            time.sleep(5)
    # Reset by pressing CTRL + C
    except KeyboardInterrupt:
        print("Measurement stopped by User")
        GPIO.cleanup()
```

PROGRAM CODE FOR ACCESS DATABASE:

```

package com.bin;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Handler;
import android.os.IBinder;
import android.support.annotation.Nullable;
import android.support.v4.app.NotificationCompat;
import android.util.Log;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
27
public class GetData extends Service {
String BASE_URL = "http://dustbin.000webhostapp.com/";
String POPMOVIES_BASE_URL = BASE_URL + "getresponsefrombin.php";
SharedPreferences preferences;
SharedPreferences.Editor editor;
@Override
public void onCreate() {
Toast.makeText(this, "Service Called", Toast.LENGTH_SHORT).show();
Log.d("Create:", "called");
//addNotification();
super.onCreate();
}
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
Log.d("onStartCommand:", "called");
preferences = getSharedPreferences("DustBin", MODE_PRIVATE);
final Handler handler = new Handler();
Runnable runnable = new Runnable() {

```

```

@Override
public void run() {
    Log.d("handler","run()");
    new DustbinTask().execute();
    handler.postDelayed(this, 5000);
}
};
//Start
handler.postDelayed(runnable, 1000);
return START_STICKY;
}
@Override
public void onDestroy() {
    Log.d("Destroy:","called");
    28
    super.onDestroy();
}
@Nullable
@Override
public IBinder onBind(Intent intent) {
    Log.d("Bind:","called");
    return null;
}
public class DustbinTask extends AsyncTask<Void, Void,Void>{
@Override
protected void onPreExecute() {
    Log.d("onPreExecute","initiate");
    try {
        if (!new Network(GetData.this).isConnected()) {
            Log.d("onPreExecute","No Internet Available!!");
            cancel(true);
        }
    }
    catch (InterruptedException | IOException e) {
        e.printStackTrace();
    }
@Override
protected Void doInBackground(Void... params) {
    HttpURLConnection urlConnection = null;
    BufferedReader reader = null;
    URL url;
    String MoviesJsonStr;
    try {
        url = new URL(POPMOVIES_BASE_URL);

```

```

urlConnection = (HttpURLConnection) url.openConnection();
urlConnection.setRequestMethod("GET");
urlConnection.connect();
InputStream inputStream = urlConnection.getInputStream();
29
StringBuilder buffer = new StringBuilder()
reader = new BufferedReader(new InputStreamReader(inputStream));
String line;
while ((line = reader.readLine()) != null) {
buffer.append(line).append("\n");
}
MoviesJsonStr = buffer.toString();
getMovieNames(MoviesJsonStr);
} catch (IOException | JSONException e1) {
e1.printStackTrace();
} finally {
if (urlConnection != null) {
urlConnection.disconnect();
}
if (reader != null) {
try {
reader.close();
} catch (final IOException ignored) {}
}
}
return null;
}
}

private void getMovieNames(String MovieJsonStr) throws JSONException {
JSONObject MovieJson = new JSONObject(MovieJsonStr);
JSONArray movieLists = MovieJson.getJSONArray("bin_info");
for (int i = 0; i < movieLists.length(); i++) {
JSONObject jMovieDetails = movieLists.getJSONObject(i);
String name = jMovieDetails.getString("bin_id");
int id = jMovieDetails.getInt("percent_filled");
Log.d("DATA", name + " " + id);
MainActivity.percent = id;
if(id>=80){
30
addNotification(id);
}
}
//Log.v("Length: ", String.valueOf(movieLists.length()));
}

```

```

//Show a notification
private void addNotification(int id) {
    int min, max;
    int percentage = preferences.getInt("last_percent",0);
    min = percentage - 5;
    max = percentage + 5;
    if (min > id || id > max) {
        Intent intent = new Intent(this, MainActivity.class
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        editor = preferences.edit();
        editor.putInt("last_percent",id);
        editor.apply();
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0/*Request code*/,
        intent, PendingIntent.FLAG_ONE_SHOT);
        //Set sound of notification
        Uri notificationSound =
        RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        NotificationCompat.Builder notifiBuilder = new NotificationCompat.Builder(this)
        .setSmallIcon(R.mipmap.ic_launcher
        .setContentTitle(id + "% Dustbin Full")
        .setContentText("Please clear your trash")
        .setAutoCancel(true)
        .setSound(notificationSound)
        .setContentIntent(pendingIntent);
        NotificationManager notificationManager = (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(999 /*ID of notification*/, notifiBuilder.build());
        //stopSelf()
    }
}

```

PROGRAM FOR CONNECTING APPLICATION TO INTERNET:

```

package com.bin;
import android.content.Context;
import android.net.ConnectivityManager;
import android.util.Log;
import java.io.IOException;
/**
 * Created by Sylvester on 03-Mar-17.
 */
class Network {
    private Context mContext;
    Network(Context mContext) {
        this.mContext = mContext;
    }
}

```

```
private boolean isNetworkAvailable() {  
    final ConnectivityManager connectivityManager = ((ConnectivityManager)  
        mContext.getSystemService(Context.CONNECTIVITY_SERVICE));  
    return connectivityManager.getActiveNetworkInfo() != null &&  
        connectivityManager.getActiveNetworkInfo().isConnected();  
}  
boolean isConnected() throws InterruptedException, IOException  
{  
    if (isNetworkAvailable()) {  
        String command = "ping -c 1 google.com";  
        return (Runtime.getRuntime().exec (command).waitFor() == 0);  
    }  
}
```