SOURCE CODE:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(19, 18, 5, 17, 16, 4);
#include <Arduino.h>
#include "HX711.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 27;
const int LOADCELL_SCK_PIN = 26;

HX711 scale;

//Temperature Sensor Define I/O
const int analogtemp = 25; // This is the analog pin which is measuring the input voltage from the LM35 temperature sensor
double temp=0, Vin=0, samples[250];
const double Vref=3300.0;

const int trigPin = 32;
const int echoPin = 33;
//define sound speed in cm/uS
#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701
long duration;
int distanceCm;
float distanceInch;
```

```
int ke = 14;
int key = 0;
 //RFID int count = 0;
// count = 0 char S[12];
// character array of size 12
boolean flag = 0; // flag =0
 int i,j,k = 0;
int incomingByte = 0;
int motor = 23;
void setup() {
 // put your setup code here, to run once: pinMode(trigPin, OUTPUT);
// Sets the trigPin as an Output pinMode(echoPin, INPUT);
// Sets the echoPin as an Input pinMode(ke, INPUT);
 pinMode(motor, OUTPUT);
Serial.begin(9600);
 scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
scale.set_scale(-478.507);
scale.tare(); lcd.begin(16, 2);
//LCD Initialize lcd.clear();
lcd.setCursor(0,0);
lcd.print("EXCESS FOOD ");
 lcd.setCursor(0,1);
lcd.print("REDISTRIBUTION");
delay(2000);
 lcd.clear(); }
void loop()
{
// put your main code here, to run repeatedly:
 // 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);
// Calculate the distance distanceCm = duration * SOUND_SPEED/2;
key = digitalRead(ke);
Vin=0;
temp=0;
float tempi = (analogRead(analogtemp));
// Each sample is a value from 0 to 1023. Reading "j" values will help making the reading more accurate. Vin = Vin + ( tempi * Vref/4096.0); //Vin=Vin/250.0;
// Calculate the average value from all "j" readings. int tempvv = (Vin/10.0);
lcd.setCursor(0,0);
lcd.print("D:"); lcd.print(distanceCm);
lcd.print(" ");
lcd.setCursor(6,0);
lcd.print("W:");
lcd.print(key);
lcd.print(" ");
//lcd.setCursor(11,0);
//lcd.print("W:");
lcd.print(scale.get_units(10), 2);
lcd.print(" ");
lcd.setCursor(0,1);
lcd.print("T:"); l
cd.print(tempvv);
lcd.print(" ");
if(key == 1)
{ gsm_msg(1); }
myStream_rfid();
Scale.power_down();
```

```
delay(1000);
scale.power_up(); }
void gsm_msg(int a) {
 lcd.setCursor(6,1);
lcd.print("SDG.");
 delay(100);
Serial.println("AT");
 delay(500);
Serial.println("AT+CMGF=1"); //To send SMS in Text Mode delay(2000);
 if (a == 1){ Serial.println("AT+CMGS=\"+919894451609\"\r");
// change to the phone number you using lcd.setCursor(6,1);
 lcd.print("SDG..1"); delay(100); }
delay(2000);
 lcd.setCursor(6,1);
lcd.print("SDG..");
Serial.println("Food Donated in Box... Pls Collect...!");//the content of the message delay(500);
lcd.setCursor(6,1); lcd.print("SDG...");
 delay(1000); Serial.println((char)26);//the stopping character delay(2000); lcd.setCursor(6,1);
lcd.print("SMS SENT....");
delay(2000)
lcd.setCursor(6,1);
lcd.print(" ");
delay(100); }
void myStream_rfid()
{ if(Serial.available())
{ count = 0; while(Serial.available() && count <= 12) // Read 12 characters and store them in S array
{ S[count] = Serial.read();
 count++;
delay(50);
 }
//2B00E425EF05 if(S[0]=='2' && S[1]=='B' && S[2]=='0' && S[3]=='0' && S[4]=='E' && S[5]=='1' &&
S[6]=='D' && S[7]=='5' && S[8]=='F' && S[9]=='9' && S[10]=='E' && S[11]=='6')
```

```
{ delay(100);

lcd.setCursor(10,1);

 lcd.print("P:P1 ");

delay(100);

digitalWrite(motor, HIGH);

delay(2000);

digitalWrite(motor, LOW); }

//2B00E8A45136 else if(S[0]=='2' && S[1]=='B' && S[2]=='0' && S[3]=='0' && S[4]=='E' && S[5]=='1' && S[6]=='C' && S[7]=='0' && S[8]=='8' && S[9]=='F' && S[10]=='8' && S[11]=='5')

 { delay(100); lcd.setCursor(10,1); lcd.print("P:P2 "); delay(100); digitalWrite(motor, HIGH); delay(2000)

digitalWrite(motor, LOW);

}

 }

 }
```