PES UNIVERSITY

JAASWIN D KOTIAN - PES2UG19CS156

LOHITH - PES2UG19CS203

MANNAV DHAMIJA – PES2UG19CS220

Session: Aug – Dec 2021

SEMESTER – 5

UE19CS313: Internet of Things: Assignment-2

Question: Implementation of physical computing project using sensors/actuators/microcontrollers/ microprocessors

a. Identification of a use case for physical computing project

Waste management has been a crucial issue in the recent decade. The use of smart dustbins is a way to achieve this good cause. 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. The **smart dustbin** is a carefully designed solution that solves the social issue of waste disposal.

b. Listing the features planned

In this project smart dustbin is built on a microcontroller-based platform Arduino Uno board which is interfaced with the Servo motor 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 at a particular level. Arduino will be programmed in such a way that when someone will comes in front of dustbin the servo motor will come in action and open the lid for the person to put the waste material into the dustbin.

c. Listing the requirements of SW and HW components to realise the project

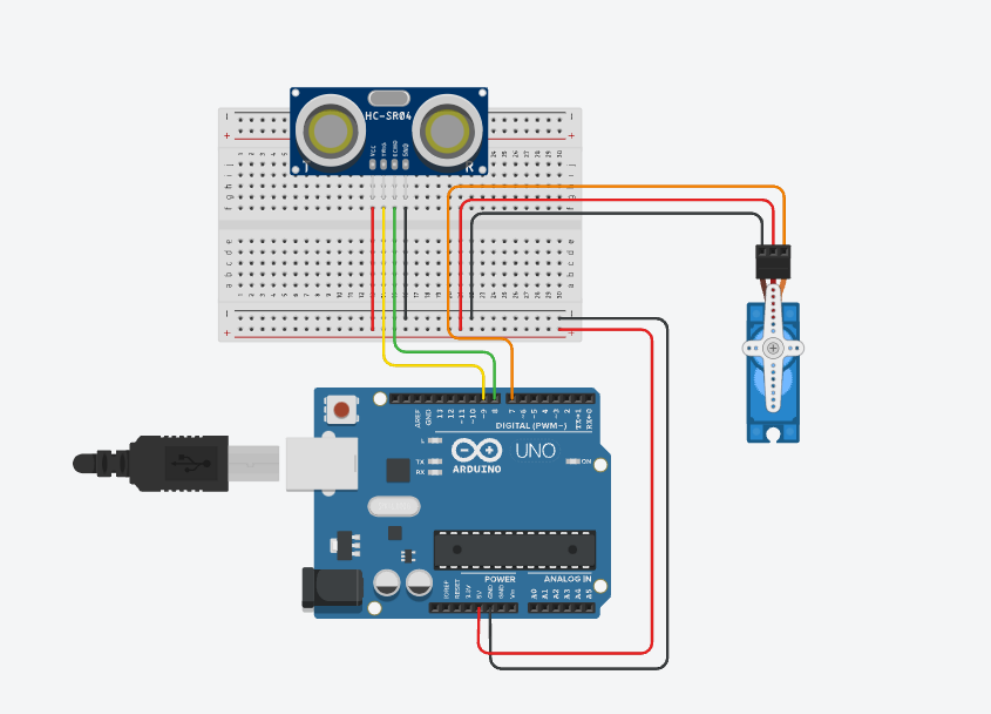
Hardware requirements:

Arduino Uno, Ultrasonic sensor, Servo motor, Jumper Wires, Battery, Household Dustbin

Software requirements:

Arduino IDE

d. Coming up with the circuit design for the project



e. Coming up with the necessary logic to implement all the listed features in (b)

The smart dustbin works on the logic that when an object comes within the range of 10 cm then the lid of the dustbin opens due to the rotation of the micro-servo. This has been with implemented with the help of a for loop and a delay.

Given below is the screenshot of the code:

```
#include <Servo.h>

Servo myservo;
const int servo_pin = 9;
const int trig_pin = 7;
const int echo_pin = 6;
const int inter_time = 200;
int time = 0;

void setup()
{
  Serial.begin(9600);
  myservo.attach(servo_pin, 500, 2400);
  myservo.write(120);
  pinMode (trig_pin, OUTPUT);
  pinMode (echo_pin, INPUT);
  delay(3000);
}

void loop()
{
  float duration, distance;
  digitalWrite(trig_pin, HIGH);
```

```
digitalWrite(trig_pin, HIGH);
delayMicroseconds(1000);
digitalWrite(trig_pin, LOW);
duration = pulseIn (echo_pin, HIGH);
distance = (duration/2)/29;
Serial.print(distance);
Serial.println(" cm");
time = time + inter_time;
delay(inter_time);
if (distance < 10)
{
  for(int i = 1500; i >= 1100; i-=25){
    myservo.writeMicroseconds(i);
    Serial.println("2");
    delay(100);
  }
  delay(5000);
  for(int i = 1100; i <= 1500; i+=25){
    myservo.writeMicroseconds(i);
    Serial.println("1");
    delay(100);
  }
}
```

f. Testing and Packaging the circuit

working model: demonstration in college