SMART PUBLIC RESTROOM

PHASE3: Development part1

To build the IOT smart public restroom

REQUIREMENTS:

- User Counter. The requirement is that sensor that can be mount on the head (top) of the door/ gate to count the user, with reasonable accuracy. ...
- Smell Sensor. ...
- Water Level Sensor. ...
- User Feedback Machine. ...
- Star Light Display.

WOKWI:

- Wokwi is an online Electronics simulator. You can use it to simulate Arduino, ESP32, STM32, and many other popular boards, parts and sensors

THING SPEAK:

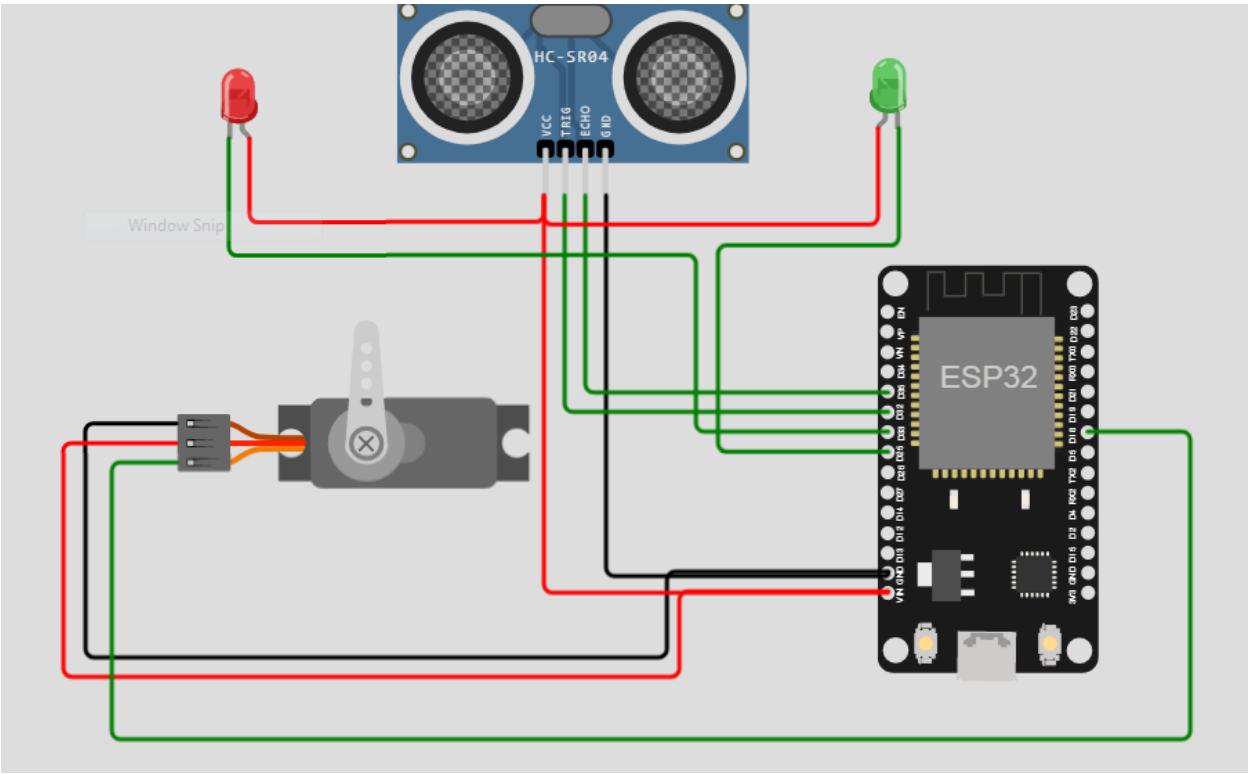
- As an "away-from-home" toilet room, a public toilet can provide far more than access to the toilet for urination and defecation. People also wash their hands, use the mirrors for grooming, get drinking water (e.g. refilling water bottles), attend to menstrual hygiene needs, and use the waste bins

PROCEDURE:

- Be Patient.

- Leave Conversations at the Door.
- Put the Seat Down.
- Avoid Neighbour Stalls.
- Keep it in the Bowl.
- Wash Your Hands.
- Don't Shake Dry.
- Check for Feet.

SIMUATION:



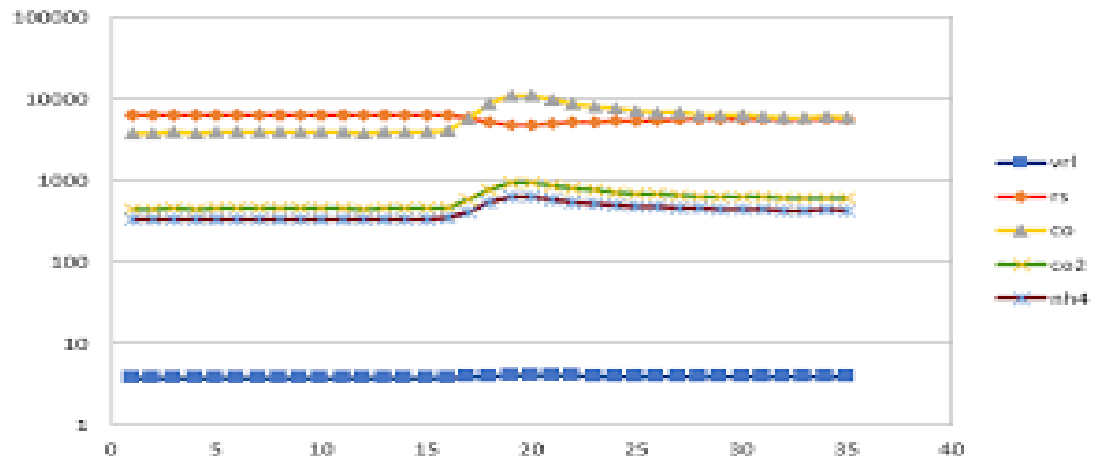
SOURCE CODE:

```
#include<ESP32Servo.h>

#define TRIGGERPIN 32
#define ECHOPIN    35
#define RED_LED    33
```

```
#define GREEN_LED  25
const char* server = "api.thingspeak.com";
const char* apiKey = "TDLDUMEY0KWNVG0I";
const unsigned long channelId = 2308576;
Servo servo_1;

long duration;
int pos, distance, i=0;

void setup()
{
  servo_1.attach(18);
  Serial.begin(115200);
  pinMode(TRIGGERPIN, OUTPUT);
  pinMode(ECHOPIN, INPUT);
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);

  Serial.println(" ");
  Serial.println("Sensing the Height");
  digitalWrite(RED_LED, HIGH);
  digitalWrite(GREEN_LED, LOW);

  pos = 0;
  servo_1.write(pos);
}

void loop()
{
  digitalWrite(TRIGGERPIN, LOW);
  delayMicroseconds(3);
  digitalWrite(TRIGGERPIN, HIGH);
  delayMicroseconds(12); // it may be 10 us
  digitalWrite(TRIGGERPIN, LOW);
```

```
// Reads the echoPin, returns the sound wave travel time in
microseconds
  duration = pulseIn(ECHOPIN, HIGH);
// Calculating the distance
  distance = (duration/2) / 29.1;


  // for Adult
  if (distance >= 100 && distance <= 150)
    {
      i = 1;
      if (pos != 180)
      {
        servo_1.write(180);
        pos = 180;
        i = 1;
      }
    }

  // for Child
    else if (distance >= 200 && distance <= 250)
      {
        i = 1;
        if (pos != 0)
        {
         servo_1.write(0);
         pos = 0;
         i = 1;
        }
      }

    else if (distance > 300 && i == 1)
      {
        digitalWrite(RED_LED, LOW);
        digitalWrite(GREEN_LED, HIGH);
        delay(5000);
```

```
        digitalWrite(RED_LED, HIGH);
        digitalWrite(GREEN_LED, LOW);
        i = 0;
      }

       delay (500);

  Serial.println(" ");
  Serial.print("Free Level : ");
  Serial.print(distance);
  Serial.print("    ");
  Serial.print("Position : ");
  Serial.print(pos);
  delay (500);
}
```



OUTPUT:

```
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13
(SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_
drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc

Sensing the Height

Free Level : 140   Position : 180
Free Level : 140   Position : 180
Free Level : 140   Position : 180
Free Level : 140   Position : 180
Free Level : 140   Position : 180
Free Level : 140   Position : 180
Free Level : 140   Position : 180
Free Level : 140   Position : 180
Free Level : 140   Position : 180
```