SMART PUBLIC RESTROOM

INTRODUCTION:

IoT (Internet of Things) technology has enabled a wide range of applications in various domains, and one such domain is the management of public restrooms. The use of IoT technology in smart public restrooms can provide better service to users, optimize maintenance and cleaning schedules, and save water and energy. Here are some of the IoT applications in smart public restrooms: Automated monitoring: IoT sensors can be installed in public restrooms to detect the occupancy of each stall, the availability of toilet paper, soap, and other supplies. Automated monitoring can help staff to respond promptly to the needs of the restroom users. Predictive maintenance: IoT sensors can IoT Applications in Smart Public Restrooms IoT (Internet of Things) technology has enabled a wide range of applications in various domains, and one such domain is the management of public restrooms. The use of IoT technology in smart public restrooms can provide better service to users, optimize maintenance and cleaning schedules, and save water and energy. Here are some of the IoT applications in smart public restrooms: Automated monitoring: IoT sensors can be installed in public restrooms to detect the occupancy of each stall, the availability of toilet paper, soap, and other supplies. Automated monitoring can help staff to respond promptly to the needs of the restroom users. Predictive maintenance: IoT sensors can A smart washroom is a customizable solution that integrates operational and environmental smart washroom sensors and IoT technology, in commercial and retail washrooms, to provide a clear view of data in real-time. Smart restroom sensors and mobile device integration make smart restroom technology enablement seamless and easy. The opportunity to improve restroom management is here. And the time to get smart about restrooms is now.Building leaders must be strategic about deploying their cleaning resources with a heightened awareness of building hygiene in today's post-pandemic era. Their efforts must be visible and reassure building occupants that their health and safety are at the forefront of the facility management team's efforts. Adding smart restroom technology that makes it easy for building managers to meet the public's expectations is not only suitable for occupants, but it's also good for business.

1.PROJECT DISCRIPTION:

- Say goodbye to unsanitary restrooms! Smart toilets utilize **self-cleaning technology** and **touchless operation** tominimize contact with germs. With **integrated UV sterilization** and **antibacterial surfaces**, these toilets ensure a safer and healthier restroom experience for everyone. Smart toilets are designed with **universal accessibility** in mind. Featuring **adjustable heights**, **handrail support**, and**braille instructions**, these facilities cater to individuals with diverse needs. By promoting inclusivity, smart toilets are transforming public restrooms into welcoming spaces for everyone

- Smart toilets are designed with **universal accessibility** in mind. Featuring **adjustable heights**, **handrail support**, and**braille instructions**, these facilities cater toindividuals with diverse needs. By promoting inclusivity, smart toilets are transforming public restrooms into welcoming spaces for everyone.

2.PROJECT IMPLIMENTATION:

Fig:2.1 Transparent public restroom.

Modern public restrooms are often cramped, dirty, and smelly. They are also usually located in the middle of a busy street or in a crowded mall. This makes it difficult for people to use them without feeling uncomfortable. Smart toilet restroom technology aims to improve the design and overall user experience of public restrooms. Making them more comfortable and convenient for everyone. One way to improve the design of smart public restrooms is by making them more user-friendly. For example, some restrooms now have touch screen-controlled toilets that allow users to select various settings such as water pressure and temperature. Other features that can make public restrooms more user-friendly include automatic flushing toilets, hands-free soap dispensers, and paper towel dispensers.

Another way to improve the design of public restrooms is by making them clean and sanitary. This can be done by installing automatic flush toilets, hands-free soap dispensers, and paper towel dispensers. Automatic flush toilets help to keep the Clean Toilet Bowl and sanitary by flushing it after each use. Hands-free soap dispensers help to prevent the spread of germs by allowing users to wash their hands without having to touch a dirty soap dispenser. Paper towel

dispensers help to keep the restroom clean by allowing users to dry their hands without

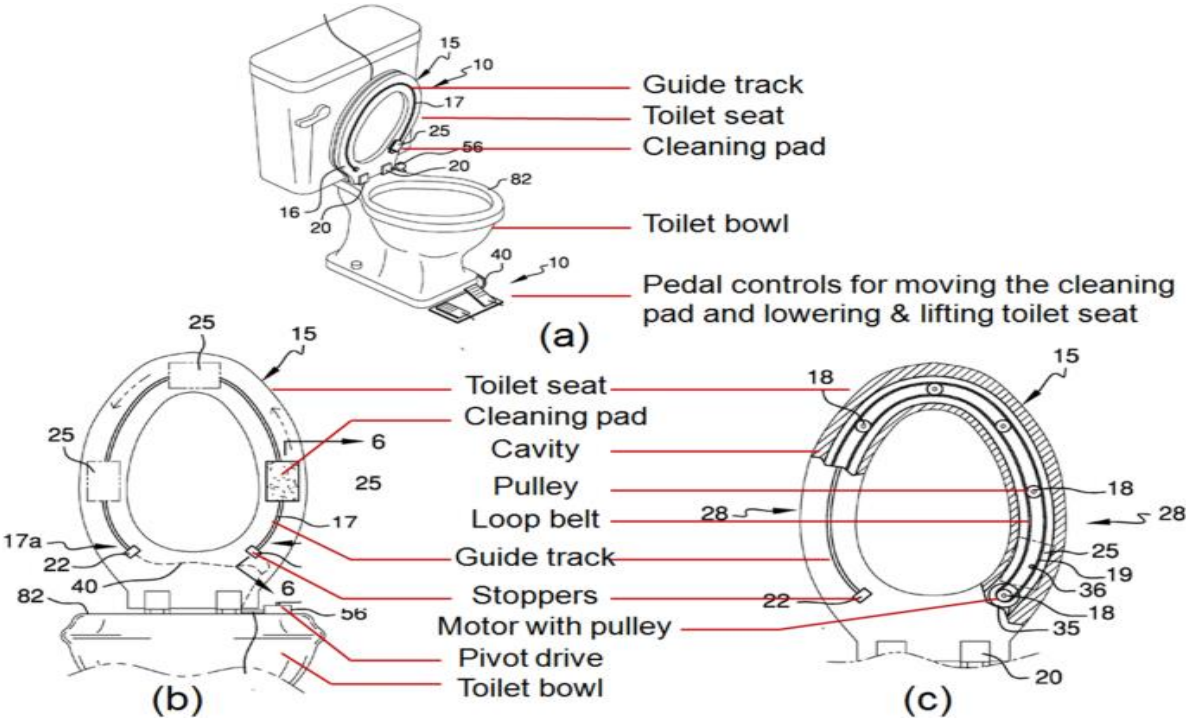

Fig:2.2 Smart punlic restroom implimentation

3.CIRCUIT DIAGRAM:

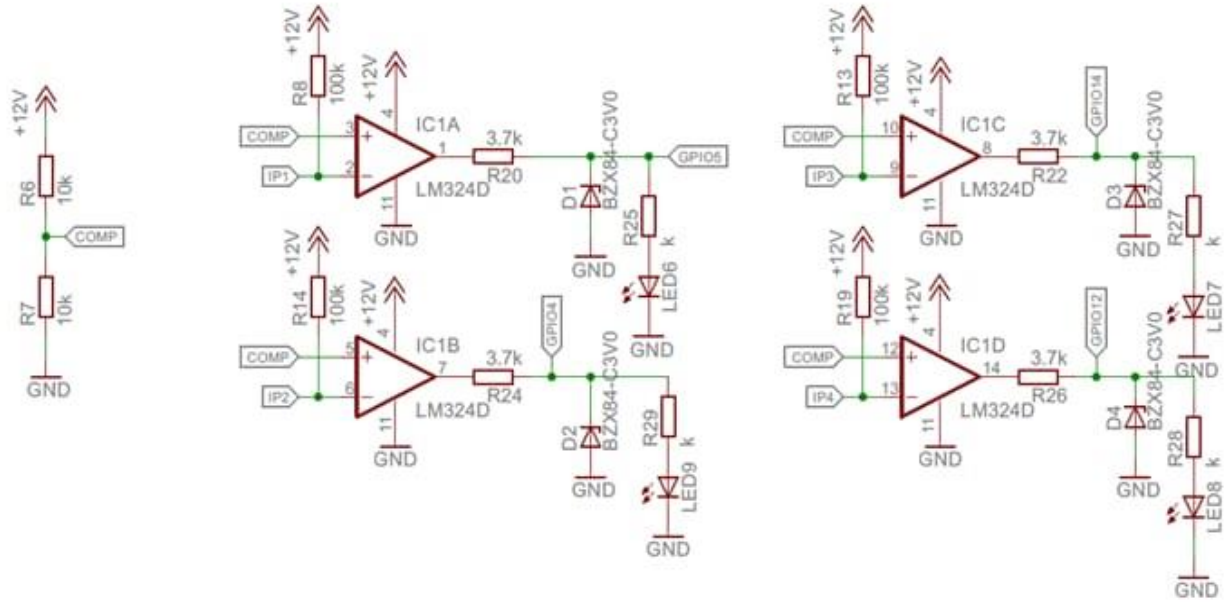

Fig:3.1 Smart public restroom Circuit diagram

4. SOURCE CODE:

FRONT END CODE :

```
#include<ESP32Servo.h>

#define TRIGGERPIN 32
#define ECHOPIN    35
#define RED_LED    33
#define GREEN_LED  25

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);
}
```

BACK-END CODE :

```
          {
 "version": 1,
 "author": "Budhaditya Biswas",
 "editor": "wokwi",
 "parts": [
  { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 57.34, "left": 242, "attrs": {} },
```

```
  {
    "type": "wokwi-hc-sr04",
    "id": "ultrasonic1",
    "top": -69.03,
    "left": 9.5,
    "attrs": { "distance": "139" }
  },
  { "type": "wokwi-servo", "id": "servo1", "top": 84.81, "left": -96.4, "attrs": {} },
  {
    "type": "wokwi-led",
    "id": "led1",
    "top": -44.13,
    "left": -86.2,
    "attrs": { "color": "red" }
  },
  {
    "type": "wokwi-led",
    "id": "led2",
    "top": -48.8,
    "left": 227.13,
    "attrs": { "color": "limegreen", "flip": "1" }
  }
],
"connections": [
  [ "esp:TX0", "$serialMonitor:RX", "", [] ],
  [ "esp:RX0", "$serialMonitor:TX", "", [] ],
  [ "servo1:GND", "esp:GND.2", "black", [ "h-44.1", "v112.72", "h280.67", "v-35.33" ] ],
  [ "esp:VIN", "servo1:V+", "red", [ "h-100.83", "v41.03", "h-297.33", "v-0.67" ] ],
  [ "servo1:PWM", "esp:D18", "green", [ "h-31.43", "v133.06", "h520.67", "v-2.67" ] ],
  [ "ultrasonic1:GND", "esp:GND.2", "black", [ "v183", "h8.03" ] ],
  [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v191", "h166.2" ] ],
  [ "ultrasonic1:TRIG", "esp:D32", "green", [ "v0" ] ],
  [ "ultrasonic1:ECHO", "esp:D35", "green", [ "v0" ] ],
  [ "led1:A", "ultrasonic1:VCC", "red", [ "v40.33", "h142" ] ],
```

```
    [ "led2:A", "ultrasonic1:VCC", "red", [ "v46.33", "h-161.33" ] ],
    [ "led1:C", "esp:D33", "green", [ "v56.33", "h225.37", "v94" ] ],
    [ "led2:C", "esp:D25", "green", [ "v56.33", "h-87.3", "v102" ] ]
  ],
  "dependencies": {}
}
```

5. SAMPLE OUTPUT SCREENSHOTS:

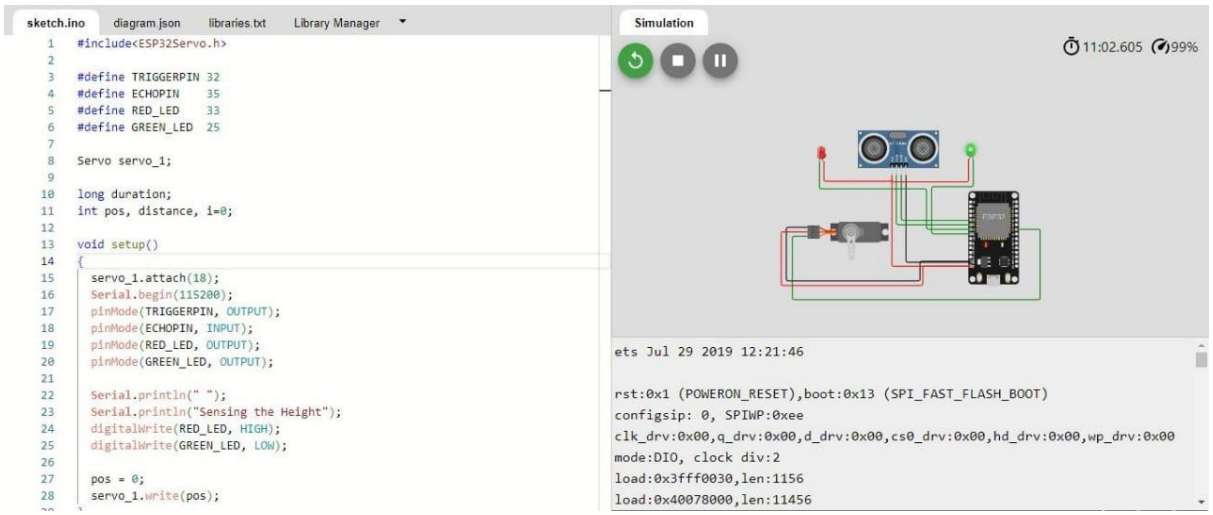

Fig:5.1

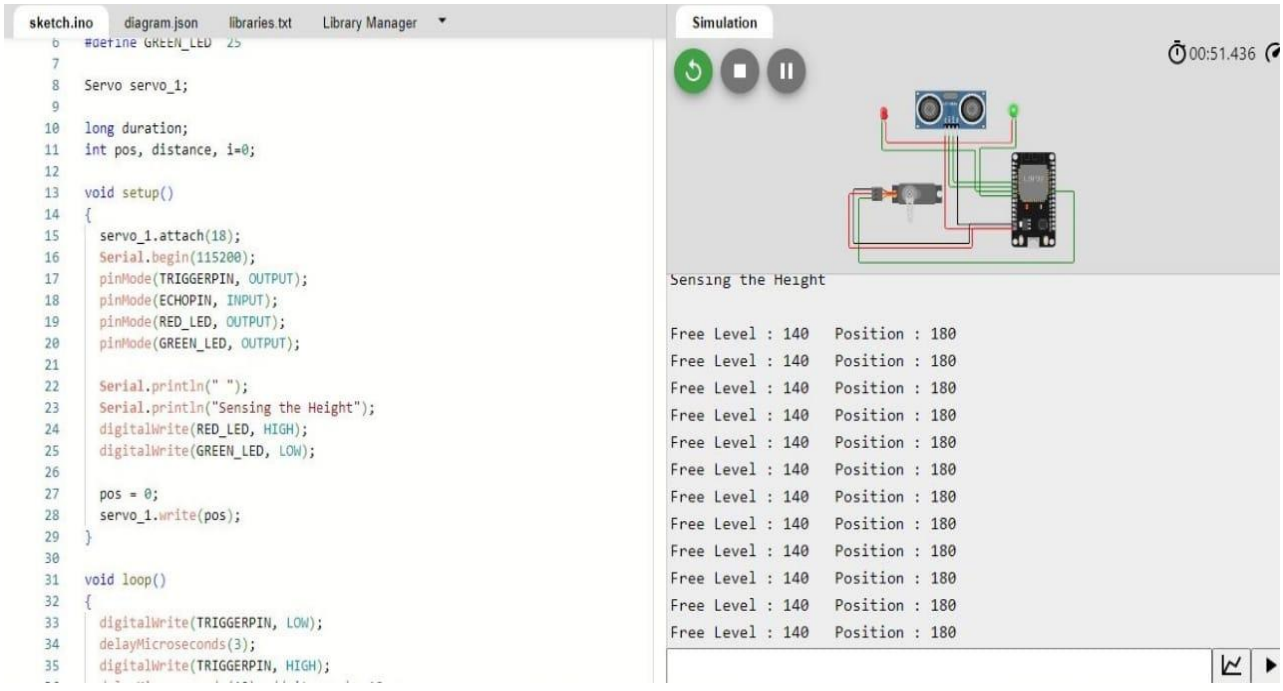

Fig:5.2

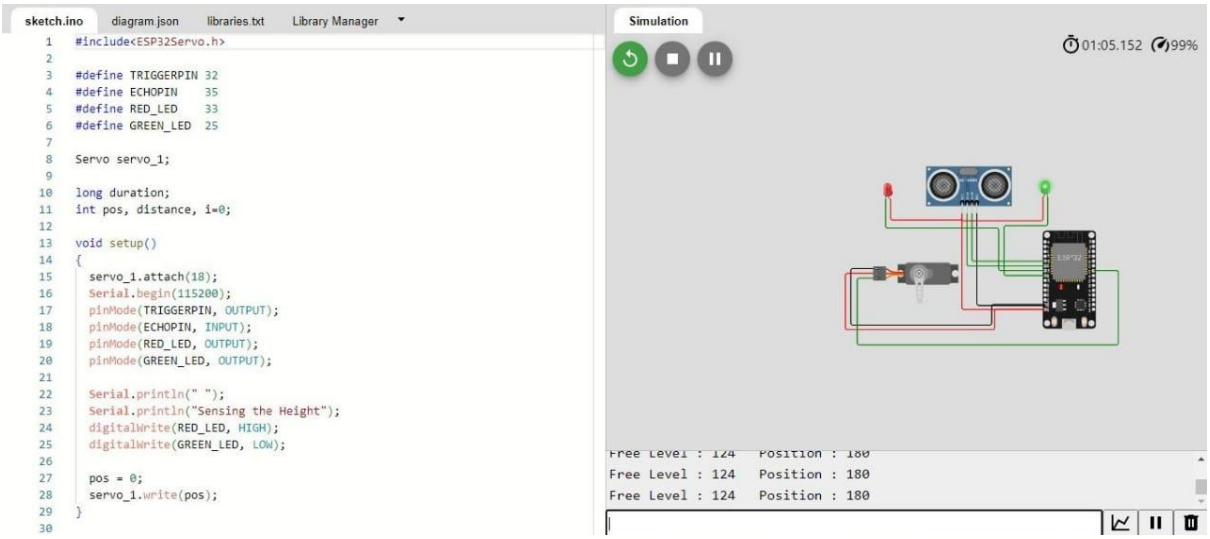
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
01:05.152
99%
Free Level : 124 Position : 180
Free Level : 124 Position : 180
Free Level : 124 Position : 180

Fig:5.3

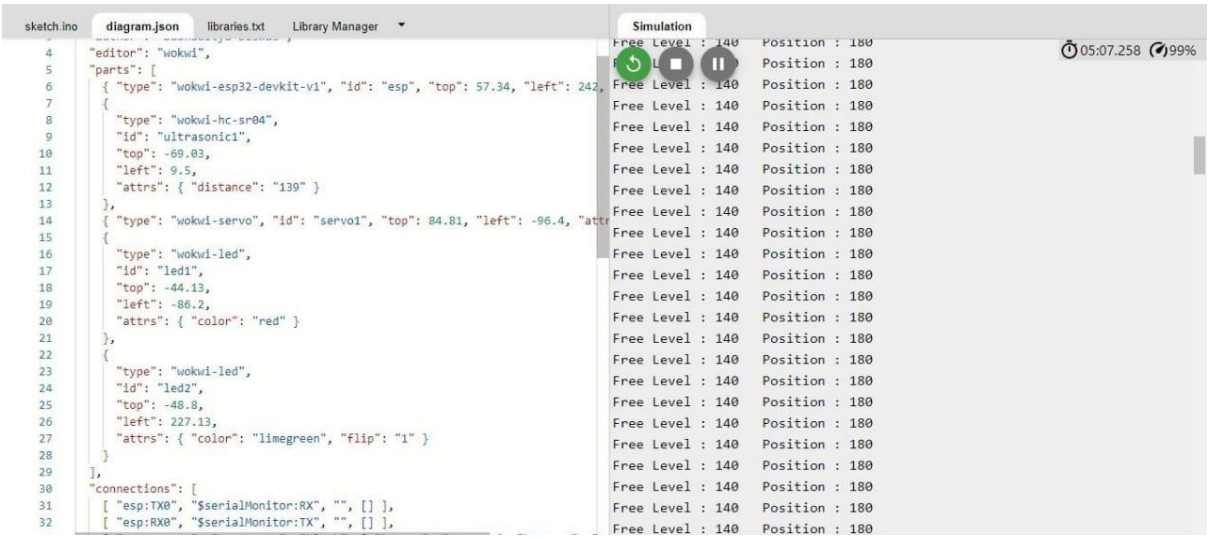
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
05:07.258
99%
Free Level : 140 Position : 180

Fig:5.4


sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
07:04.081
106%
"attrs": { "color": "red" }
"type": "wokwi-led",
"id": "led2",
"top": -48.8,
"left": 227.13,
"attrs": { "color": "limegreen", "flip": "1" }
"connections": [
["esp:TX0", "$serialMonitor:RX", "", []],
["esp:RX0", "$serialMonitor:TX", "", []],
["servo1:GND", "esp:GND.2", "black", ["h-44.1", "v112.72", "h280.67",
["esp:VIN", "servo1:V+", "red", ["h-100.83", "v41.03", "h-297.33",
["servo1:PWM", "esp:D18", "green", ["h-31.43", "v133.06", "h520.67",
["ultrasonic1:GND", "esp:GND.2", "black", ["v183", "h8.03"]],
["ultrasonic1:VCC", "esp:VIN", "red", ["v191", "h166.2"]],
["ultrasonic1:TRIG", "esp:D32", "green", ["v0"]],
["ultrasonic1:ECHO", "esp:D35", "green", ["v0"]],
["led1:A", "ultrasonic1:VCC", "red", ["v40.33", "h142"]],
["led2:A", "ultrasonic1:VCC", "red", ["v46.33", "h-161.33"]],
["led1:C", "esp:D33", "green", ["v56.33", "h225.37", "v94"]],
["led2:C", "esp:D25", "green", ["v56.33", "h-87.3", "v102"]]
],
"dependencies": {}
}
Free Level : 140 Position : 180

Fig:5.5

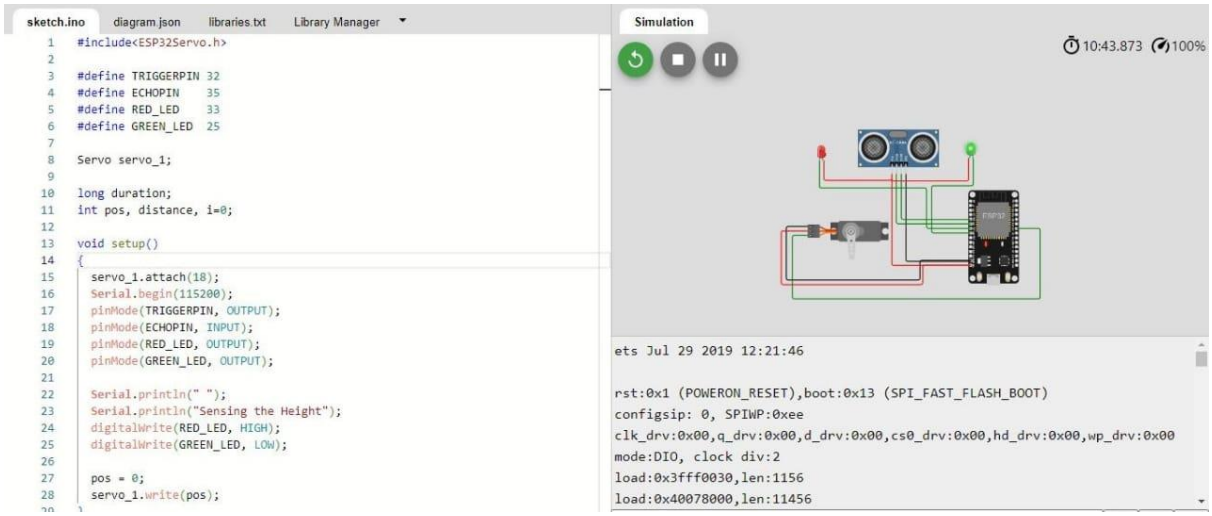
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
10:43.873
100%
#include<ESP32Servo.h>
#define TRIGGERPIN 32
#define ECHOPIN 35
#define RED_LED 33
#define GREEN_LED 25
Servo servo_1;
long duration;
int pos, distance, i=0;
void setup()
{
servo_1.attach(18);
Serial.begin(115200);
pinMode(TRIGGERPIN, 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);
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

Fig:5.6


sketch.ino
diagram.json
libraries.txt
Library Manager
#include<ESP32Servo.h>
#define TRIGGERPIN 32
#define ECHOPIN 35
#define RED_LED 33
#define GREEN_LED 25
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);
}
Simulation
11:02.605
99%
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

Fig:5.7

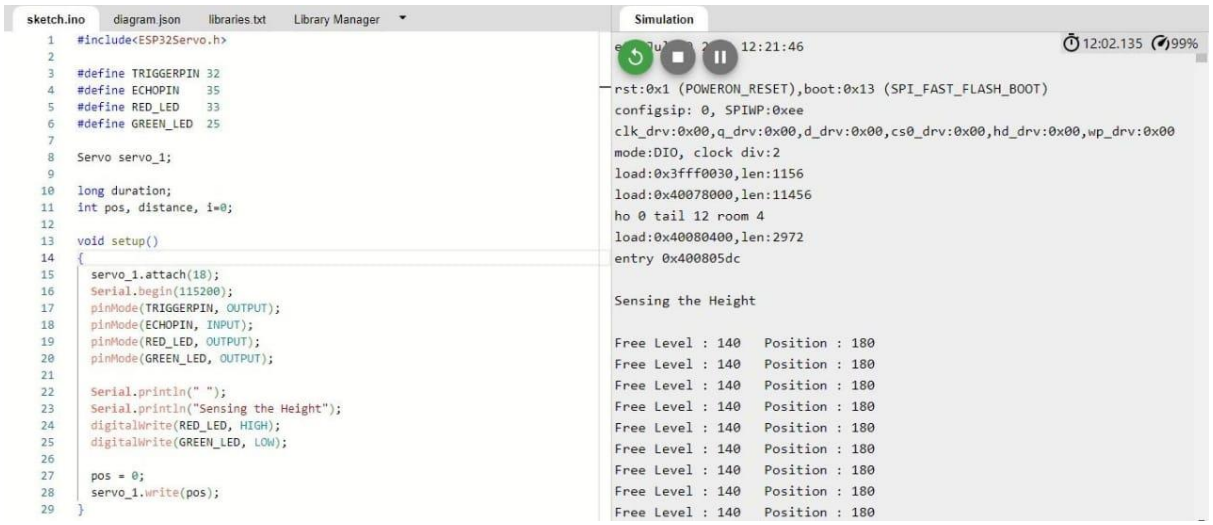
sketch.ino
diagram.json
libraries.txt
Library Manager
#include<ESP32Servo.h>
#define TRIGGERPIN 32
#define ECHOPIN 35
#define RED_LED 33
#define GREEN_LED 25
Servo servo_1;
long duration;
int pos, distance, i=0;
void setup()
{
servo_1.attach(18);
Serial.begin(115200);
pinMode(TRIGGERPIN, OUTPUT);
pinMode(ECHOPIN, 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);
}
Simulation
12:02.135
99%
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

Fig:5.8