# SMART PUBLIC RESTROOM

## INTRODUCTION:

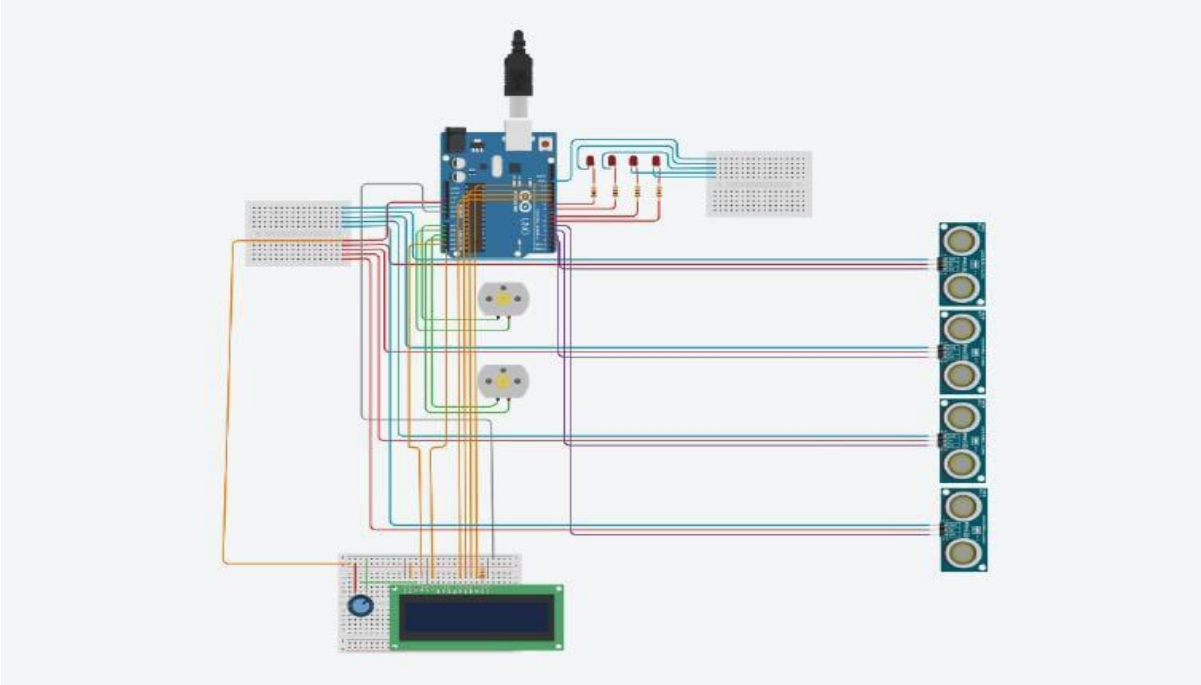
- The project Development of a smart toilet for automatic flushing deals with automatic cleaning of Indian toilets without requiring any human assistance.
- Smart restrooms in airports are technologically advanced restroom facilities that use various sensors, automation, and data analysis to improve hygiene, maintenance, and user experience.
- Many smart toilets have automatic flushing and hands-free operation (especially after COVID) to help keep surfaces and floors clean.
- Most of the public toilets are not clean due to the irresponsible peoples who often forget to flush the toilet after using it.
- In India all the state and central government are allotting numerous funds for constructing public toilets.
- The central government under "SWACH BHARAT MISSION" has built a vast amount of new toilets to provide the citizens a healthy and hygienic environment.
- Therefore cleaning of public toilets is equally important as cleaning of household toilets.
- So we have developed a mechanism to flush the toilets automatically by utilizing the human weight.
- The mechanism does not require any external power or human concern.Rather, it just works mechanically utilizing the weight of the person sitting on it.

- Our smart toilet is the only system in the markets offering concealed arms over the bowl to clean and dry the bowl and surrounding walls up to 80cm.
- High-pressure ejecting water is mixed with disinfectant; a floor-integrated high-pressure nozzle system ejects water and disinfectant on the floor.

## MATERIALS REQUIRED:

- Aurdino
- Aurdino IDE Software
- Ultrasonic sensor
- DC Motor
- LCD
- Potentiometer
- Breadboard
- Led
- Resistor

## DIAGRAM:

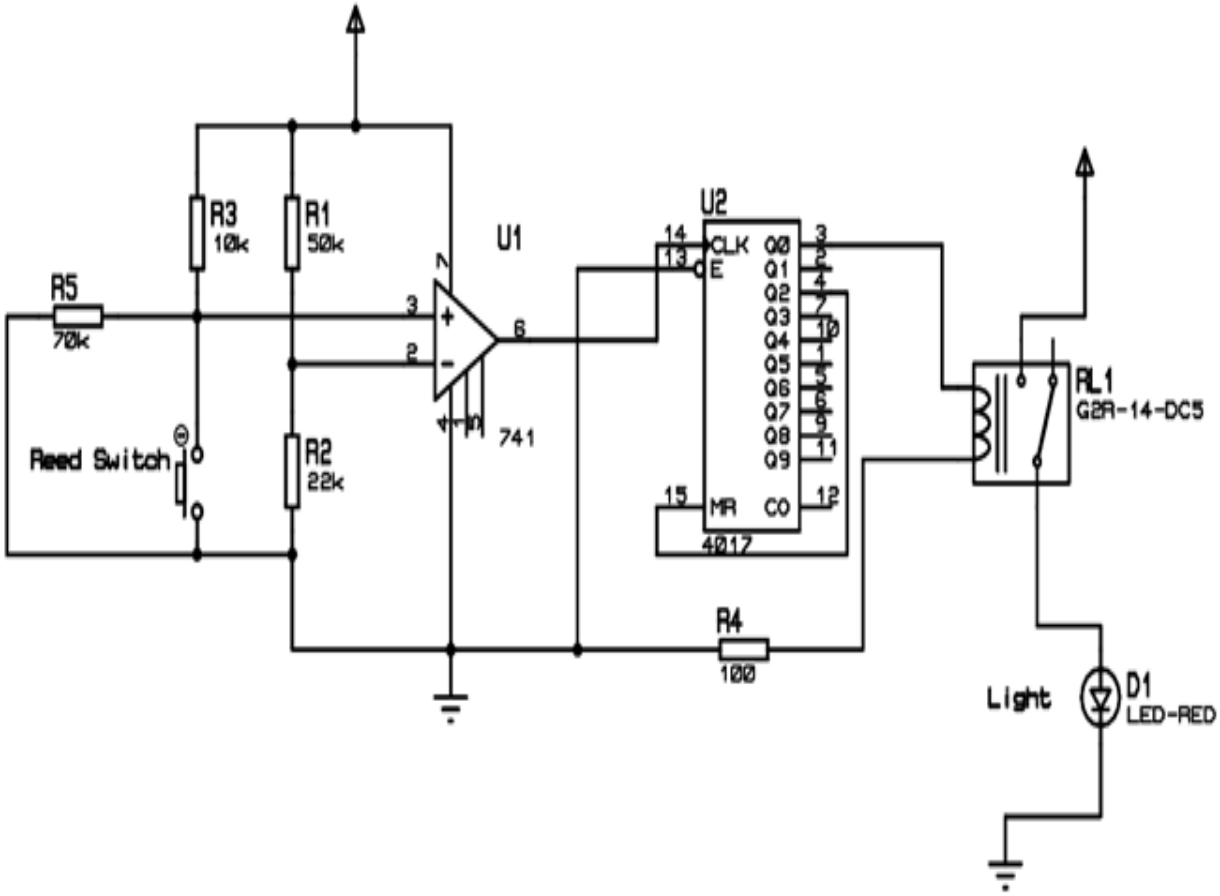


## SOFTWARE USED:

- Arduino Integrated Development Environment (IDE) is an open source IDE that allows users to write code and upload it to any Arduino board.
- Arduino IDE is written in Java and is compatible with Windows, macOS and Linux operating systems.
- Arduino is important part in robotics because it provide creativity and problem-solving.

- It is plugged in the computer and programmed with easy commands i.e. when Arduino is placed in a circuit, and it will manipulate the functioning of the device.
- It emphasizes the involvement of Arduino in many things around.

## CIRCUIT DIAGRAM:



# CONTENT :

- Automated or touchless fixtures like faucets, soap dispensers, and flush mechanisms to minimize contact with surfaces.
- Install remote monitoring and diagnostics.
- Implement predictive maintenance.
- Train staff on technology usage.
- Motion-sensor lighting and climate control to reduce energy consumption when the restroom is not in use.
- Solar panels or other renewable energy sources to power restroom facilities.

# CODING:

```
#include <Servo.h>
#include <LiquidCrystal.h>

const int niagraMinDistance = 3;
const int niagraMaxDistance = 335;
const int trig = 7;
const int echo = 8;
const int servo = 9;
const int pingPin = 2;
const int echoPin = 3;
```

```
const int d4 = 4, d5 = 5, d6 = 6, d12 = 12;

const int rs = 10;

const int en = 11;

const int a0 = 14;

const int a1 = 15;

const int a2 = 16;


bool sitOn = false;

long duration, distance, distanceSit, duration1, a[3], starttime, endtime;

float cm = 1.1;

Servo myServo;

Servo myServoSit;

LiquidCrystal lcd(rs, en, d4, d5, d6, d12);


void setup() {

  pinMode(pingPin, OUTPUT); // set ping Out

  pinMode(echoPin, INPUT);  // set echo In

  lcd.begin(16, 2);                  // initialize LCD

  Serial.begin(9600);                // Initialize Serial

  myServo.attach(servo);

  pinMode(trig, OUTPUT);

  pinMode(echo, INPUT);

  myServo.write(180);      // servo position 180 degree (can also be 0 if we want it to turn to the
other way)

  delay(1000);

  myServo.detach();

  myServoSit.attach(a2);

  pinMode(a1, OUTPUT);

  pinMode(a0, INPUT);

  myServoSit.write(180);      // servo position 180 degree (can also be 0 if we want it to turn to the
other way)

  delay(1000);
```

```
    myServoSit.detach();
}

// Measures the distance from our hand to the the sensor
void measureServo() {
    digitalWrite(trig, LOW);
    delayMicroseconds(6);
    digitalWrite(trig, HIGH);
    delayMicroseconds(15);
    digitalWrite(trig, LOW);
    pinMode(echo, INPUT);
    duration = pulseIn(echo, HIGH);
    distance = (duration/2) / 29.1;
}

void measureServoSit() {
    digitalWrite(a1, LOW);
    delayMicroseconds(6);
    digitalWrite(a1, HIGH);
    delayMicroseconds(15);
    digitalWrite(a1, LOW);
    pinMode(a0, INPUT);
    duration = pulseIn(a0, HIGH);
    distanceSit = (duration/2) / 29.1;
}

// Measures the distance from the float inside niagara to the sensor
void measureDistanceFloat() {
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(pingPin, LOW);
  duration1 = pulseIn(echoPin, HIGH);
}

// Convert data from input pins to cm
long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}

void printDistanceToLCD(int cm) {
        lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Progress until");
        lcd.setCursor(0, 1);
  lcd.print("full tank: ");
  lcd.print(100 - ((double)cm / (niagraMaxDistance - niagraMinDistance)) * 100.0);
  lcd.print("%");
}

void printDistanceToSerial(int cm) {
  Serial.print(cm);
  Serial.print(" cm");
  Serial.println();
}

void loop() {
  measureDistanceFloat();
  cm = microsecondsToCentimeters(duration1);
  printDistanceToLCD(cm);
  printDistanceToSerial(cm);
```

```
  for (int i = 0; i <= 2; i++) {
     measureServo();
          measureServoSit();
     a[i]=distance;
     delay(50);
  }
  distance = (a[0] + a[1] + a[2]) / 3;

  if (distance < 70) {
     myServo.attach(servo);
     delay(1);
     myServo.write(90);  // Servo position 90 degree
     delay(2000);
     myServo.write(180); // Servo position 180 degree
     delay(1000);       // Waiting for the water to flush down
     myServo.detach();
  }
if (distanceSit < 70 && !sitOn) {
          starttime = millis();
                    myServoSit.attach(a2);
          delay(1);
          myServoSit.write(90);
          sitOn = true;

}
else if (distanceSit < 70 && sitOn) {
                    myServoSit.attach(a2);
          delay(1);
          myServoSit.write(180);
          sitOn = false;
```

```
 }
 endtime = millis();
 if (((endtime - starttime) >= 900000) && sitOn)
 {
        myServoSit.attach(a2);
        delay(1);
        myServoSit.write(180);
        sitOn = false;

 }
}
```

## ADVANTAGES:

- No sensors or electronics involved.
- No human effort required.
- Mechanism is robust.
- Economical.

## DRAWBACKS:

- Continuous Monitoring.
- Battery Maintenance.
- Sensor Requirement.
- Costly.