PROJECT: SMART PUBLIC RESTROOMS

- The need for smart public restrooms has grown in recent years as technology and urbanization continue to shape our modern world. These advanced public restroom facilities incorporate various technologies and design elements to enhance user experience, improve hygiene, and contribute to overall urban sustainability. Here are some reasons why smart public restrooms are increasingly in demand:

  - Hygiene and Health
  - Accessibility
  - Efficiency
  - Maintenance and Management
  - Water Conservation

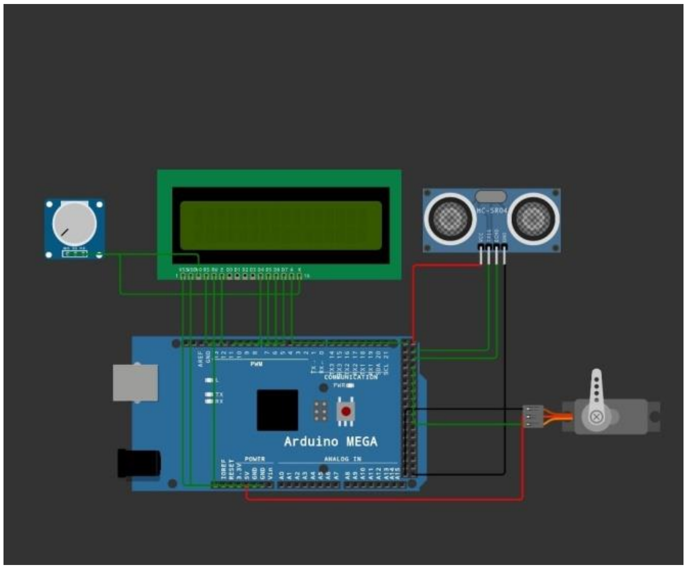
Automatic Door System:

- Automatic door systems in public restrooms serve several important purposes and provide numerous benefits, contributing to a more convenient, and accessible and hygienic restroom experience for users.
- Automatic door systems in public restrooms are particularly beneficial for specially challenged individuals, including those with disabilities, as they enhance accessibility and convenience.

Components:

- LCD1602
- Knob-controlled variable resistor (linear potentiometer)
- A standard Micro Servo Motor
- HC-SR04 Ultrasonic Distance Sensor
- Arduino Mega
- Connecting Wires

Circuit:



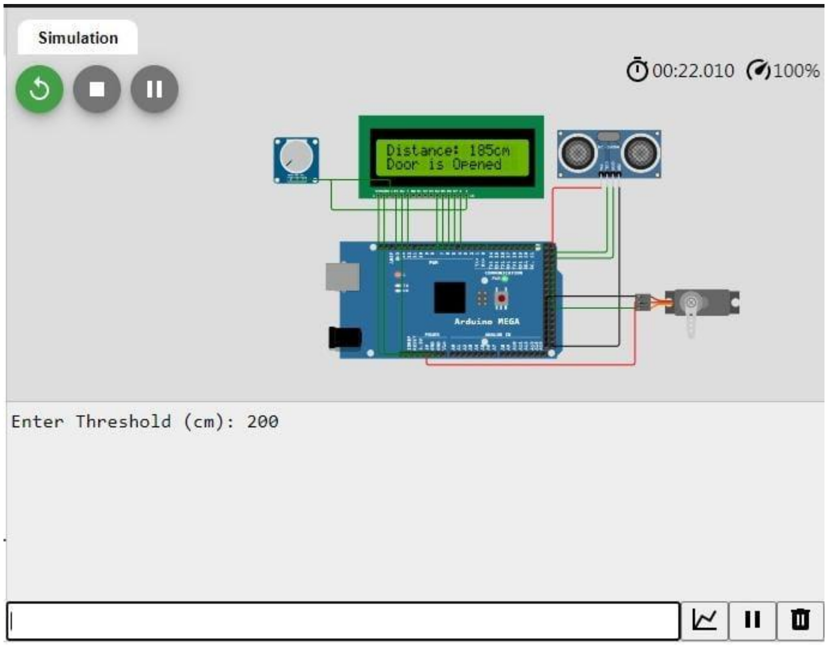
Code:

➢ Implement the given code in wokwi for simulation.

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

//Connecting LCD to the Arduino with the corresponding pins
const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
//Defining the pins as we connected then in the simulation
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

//Connecting Ultrasonic sensor to the Arduino with the corresponding pins
const int trigPin = 23;
const int echoPin = 25;

long duration;
double distance = 2;

//Create servo object
Servo servo;

void printout(String st1, String st2){
  lcd.print(st1);
  lcd.setCursor(0,1);
  lcd.print(st2);
  delay(3000);
  lcd.clear();
}

void setup() {
  //The following sets up the assembly to run once
  //Setting the size of the LCD (cols,rows)
  lcd.begin(16,2);

  //Setting up Ultrasonic sensor
  pinMode(trigPin,OUTPUT);  //The trigPin is the transmitter (output)
  pinMode(echoPin, INPUT);  //The echoPin is the receiver (input)

  Serial.begin(9600);

  String intro = "Welcome to the smart restroom";
  lcd.print(intro);
  for (int i=0; i<=(intro.length()-16); i++){
    lcd.scrollDisplayLeft();
    delay(250);
  }
  lcd.clear();
  delay(700);

  lcd.print( "~Instruction");
  delay(2000);
  lcd.clear();

  printout("~Open the Door","Move hand closer");
  printout("~Close the Door","Move further");

  //Connecting the rotor to the Arduino with the corresponding pin
  servo.attach(27);

  int threshold;
  Serial.print("Enter Threshold (cm): ");
  while (Serial.available()==0){}
  String input = Serial.readStringUntil('\n');
  Serial.println(input);

  while ((input.toInt()<=2)|| (input.toInt()>=400)){
    lcd.print( "ERROR!!");
    delay(1500);
    lcd.clear();
    printout("Number should be","in range 2-400");
    delay(700);
    Serial.print("Enter Threshold (cm): ");
    while (Serial.available()==0){}
    String input = Serial.readStringUntil('\n');
    Serial.println(input);
    if ((input.toInt()>=2)&&(input.toInt()<=400)) break;
  }
  EEPROM.write(0, input.toInt());

}


int initial = 0;

void loop(){
  //This code is run repeatedly''
  lcd.clear();

    //Clearing trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);

    //Set trigPin on HIGH mode for 10 microseconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    //Reading echoPin to store time duration from output pulse
    duration = pulseIn(echoPin, HIGH);
    distance = duration*0.034/2;

   // print out integer part of distance
   lcd.print("Distance: "+String(distance,0)+"cm");

  if((distance >= 2)||(distance <= 4)){ //rang distance is 2~400cm
    //check if the new distance is same or not same as privious distance
    // accuracy is also considerd
    if(distance != ((initial+0.3)||(initial-0.3))){

      if(distance<=EEPROM.read(0)){
        servo.write(180);
        lcd.setCursor(0,1);
        lcd.print("Door is Opened");
      }
      else if (distance>EEPROM.read(0)){
        servo.write(0);
        lcd.setCursor(0,1);
        lcd.print("Door is Closed");
      }
      //store distance as initial and do loop again
      initial = distance;
      delay(1000);
    }
  }
}
```

Output:

- ➢ `Use Wokwi for simulation and set the threshold value for 200cm.`



```
Distance: 185cm
Door is Opened
```

```
Enter Threshold (cm): 200
```

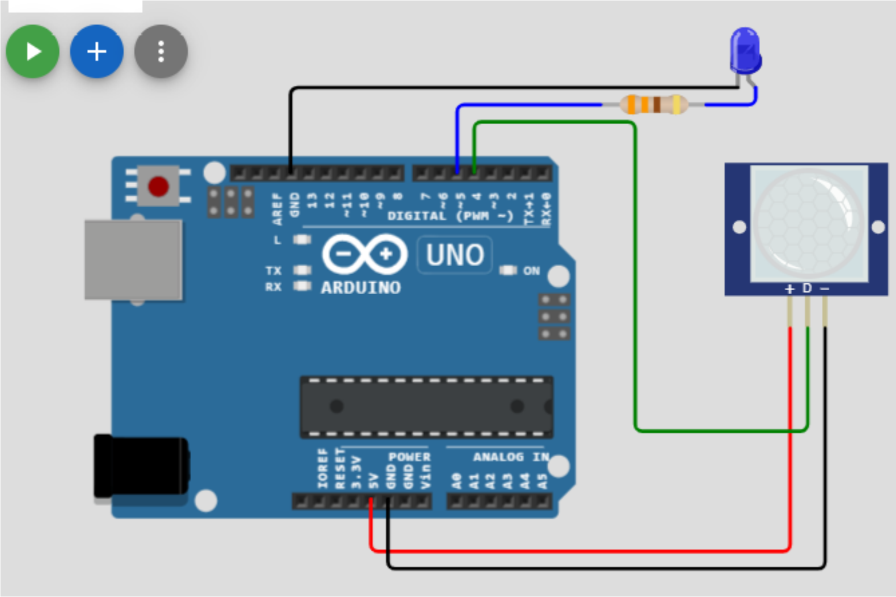
AUTOMATIC LIGHTING SYSTEM

- ➢ An automatic lighting system in a public restroom serves several purposes and offers various benefits:
  - It provides an energy efficiency by reducing the excess use of power.
  - Users don't have to fumble for light switches when entering a restroom. The lights come on automatically, providing a convenient and hands-free experience.
  - Automatic lighting systems can enhance safety by ensuring that the restroom is well-lit when in use, reducing the risk of accidents or falls in dimly lit areas.
  - Cost Savings: Over time, the energy savings and reduced maintenance costs associated with automatic lighting systems can lead to cost savings for building owners and managers.

Components:

- Arduino UNO
- LED
- Resistor(330 Ω)
- PIR Sensor
- Connecting Wires

Circuit:

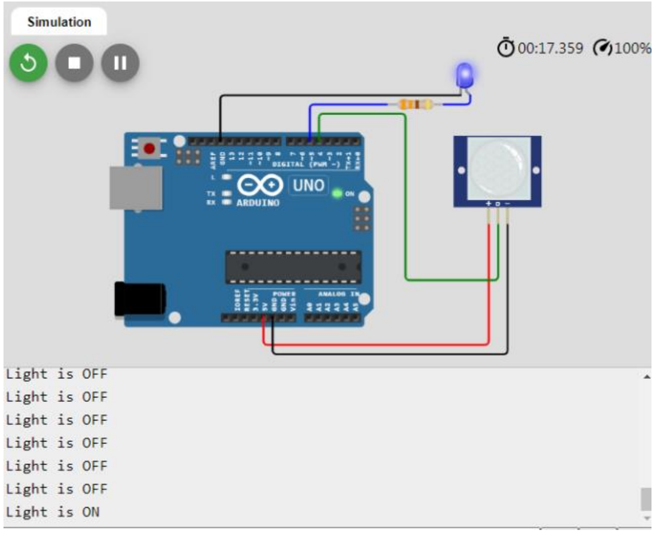


Code:

➢ Implement the given code in wokwi for simulation.

```
int pirpin = 4;
int PIRstatus = 0;

void setup() {
  // put your setup code here, to run once:
pinMode(4, INPUT);
pinMode(5, OUTPUT);
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
PIRstatus = digitalRead(pirpin);
if(PIRstatus == HIGH){
  Serial.println("Light is ON");
  digitalWrite(5, HIGH);
  delay(300000);
}
if(PIRstatus == LOW){
  Serial.println("Light is OFF");
  digitalWrite(5, LOW);
}
delay(300);
}
```

Output:



```
Light is OFF
Light is OFF
Light is OFF
Light is OFF
Light is OFF
Light is OFF
Light is ON
```

WATER LEVEL MONITORING

➢ Water level monitoring in public restrooms using the Internet of Things (IoT) technology offers several additional benefits and applications beyond traditional water level monitoring. Here's how IoT-based water level monitoring can be used in public restrooms:

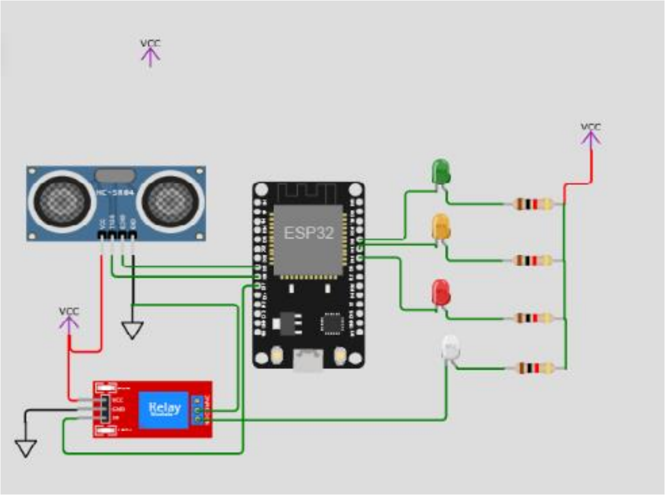
- IoT sensors can provide real-time data on water levels in toilets, urinals, sinks, and other fixtures. This data can be accessed remotely through a web interface or a mobile app, allowing facility managers to monitor water usage and detect anomalies in real-time.
- IoT systems can send alerts and notifications via email or SMS when water level anomalies are detected, allowing for immediate response to issues. This can be especially valuable for preventing water damage and minimizing downtime.

- IoT-based monitoring can play a crucial role in water conservation efforts. By providing detailed data and insights into water usage, facility managers can make informed decisions to reduce water waste, set targets for water conservation, and track progress.

Components:

- ESP32
- HC-SR04 Ultrasonic Distance Sensor
- Relay Module
- LED - 4
- Resistor(1kΩ) - 3
- Connecting Wires

Circuit:



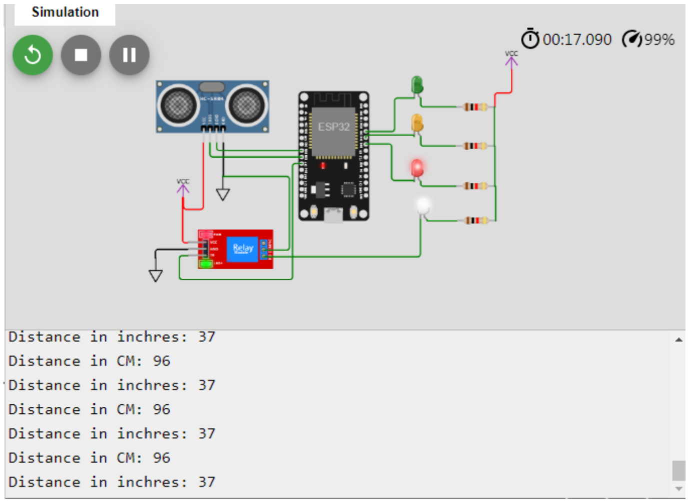
Code:

➢ Implement the given code in wokwi for simulation.

```
#define PIN_TRIG 26
#define PIN_ECHO 25
#define LOWLED   18
#define MIDLED   19
#define HIGHLED  21
#define MOTOR    27
```

```
 unsigned int level = 0;

 void setup(){

  pinMode(LOWLED,OUTPUT);
  pinMode(MIDLED,OUTPUT);
  pinMode(HIGHLED,OUTPUT);
  pinMode(MOTOR,OUTPUT);
  digitalWrite(LOWLED, HIGH);
  digitalWrite(MIDLED, HIGH);
  digitalWrite(HIGHLED, HIGH);
  digitalWrite(MOTOR, HIGH);

  Serial.begin(115200);
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
 }

void loop(){
//start a new measurement:
digitalWrite(PIN_TRIG, HIGH);
delayMicroseconds(10);
digitalWrite(PIN_TRIG, LOW);

//Read te result:
int duration = pulseIn(PIN_ECHO,HIGH);
Serial.print("Distance in CM: ");
Serial.println(duration / 58);
Serial.print("Distance in inchres: ");
Serial.println(duration / 148);

level = duration / 58;

if (level < 100)
{

       digitalWrite(LOWLED,LOW);
       digitalWrite(MIDLED,HIGH);
       digitalWrite(HIGHLED, HIGH);
       digitalWrite(MIDLED, HIGH);

}
else if ((level > 100) && (level<300))
{
       digitalWrite(LOWLED,HIGH);
       digitalWrite(HIGHLED, HIGH);
       digitalWrite(MIDLED, LOW);

}
else if (level >= 300)
{
      digitalWrite(LOWLED,HIGH);
       digitalWrite(MIDLED,HIGH);
        digitalWrite(HIGHLED, LOW);

}
delay(1000);
}
```

Output:



```
Distance in inchres: 37
Distance in CM: 96
Distance in inchres: 37
Distance in CM: 96
Distance in inchres: 37
Distance in CM: 96
Distance in inchres: 37
```