Automated Sanitize System

Introduction: -

It is a model based on Internet of Things (IoT) by which people will sanitize the public transport or stations automatically just in few seconds. It will prevent further outspread of Covid-19. It contains some disinfectant which is used to disinfect the public transport.

The disinfectant solution used consists of a combination of sodium hypochlorite (NaOCl) and water (H2O). The disinfectant is non-volatile, thus enabling prolonged veridical and bactericidal activity and sanitizing the surfaces.

Components Required: -

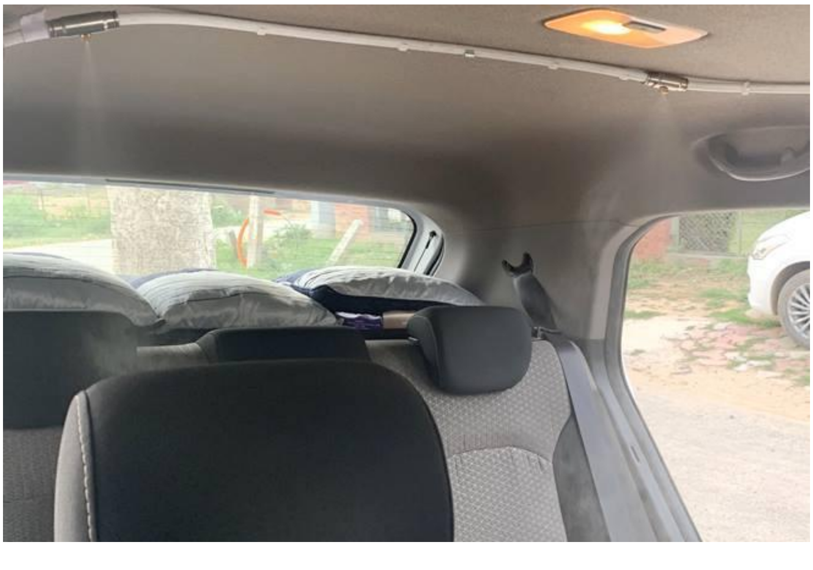
- Arduino UNO
- PIR motion sensor
- Relay
- SMPS

Apps and Software: -

- Arduino IDE

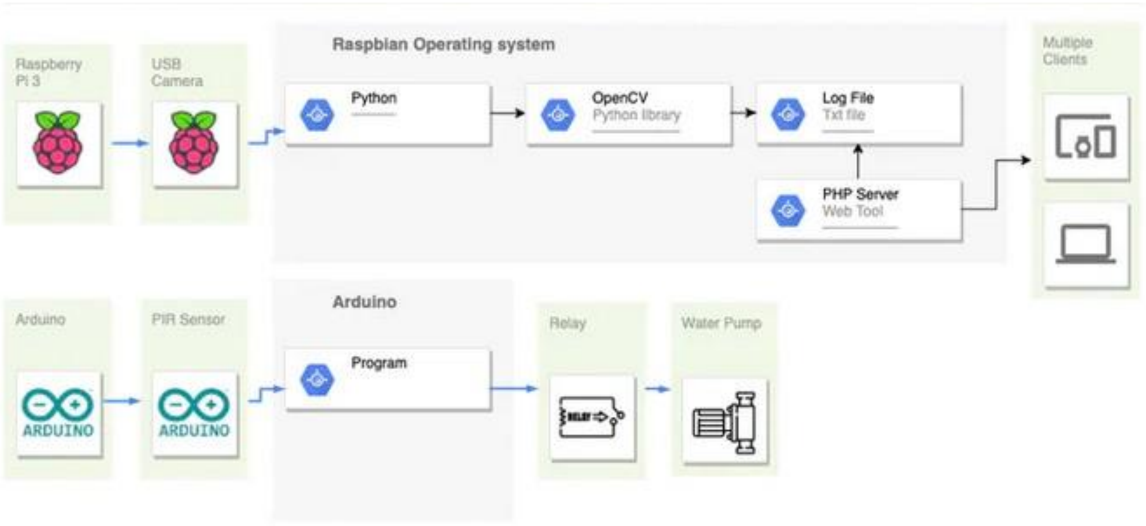
How it works: -

A 1HP Water Pump machine is placed on the side of each tunnel that takes the solution of 0.4% of Sodium hypochlorite solution in 100 Litre of water from the tank. As the machine is automatic it senses and if there is no one in the public transport then the pump will be auto start and disinfect the interior of transport. The misty disinfectant spray protects citizens from catching bacteria for a period of at least 60 minutes (Approx.). As it disinfects the air, exposed skin, and human clothing.



Some of the highlights of the apparatus are:

- If unapproved disinfectant/liquid is used in the apparatus, it will not function.
- It will operate only when the vehicle is at rest.
- It will operate only for the limited recommended/desired set duration (second/minutes).
- It will not operate if the sprayer pressure is high/low plus-minus 10% or as recommended.
- It can be linked to the IR temperature recorder if the temperature of a passenger sitting on any seat records higher than recommended or present, this apparatus will not function and a notification will be sent to driver/operator's helpdesk for necessary directions.
- It can also be linked to large operators' network/app for tracking monitoring of the set SOPs followed by the drivers or not.
- It will notify the user if the disinfectant level is critically low.
- It will be powered by vehicle battery or self-powered with chargeable batteries as the case may be.



Hardware Setup: -

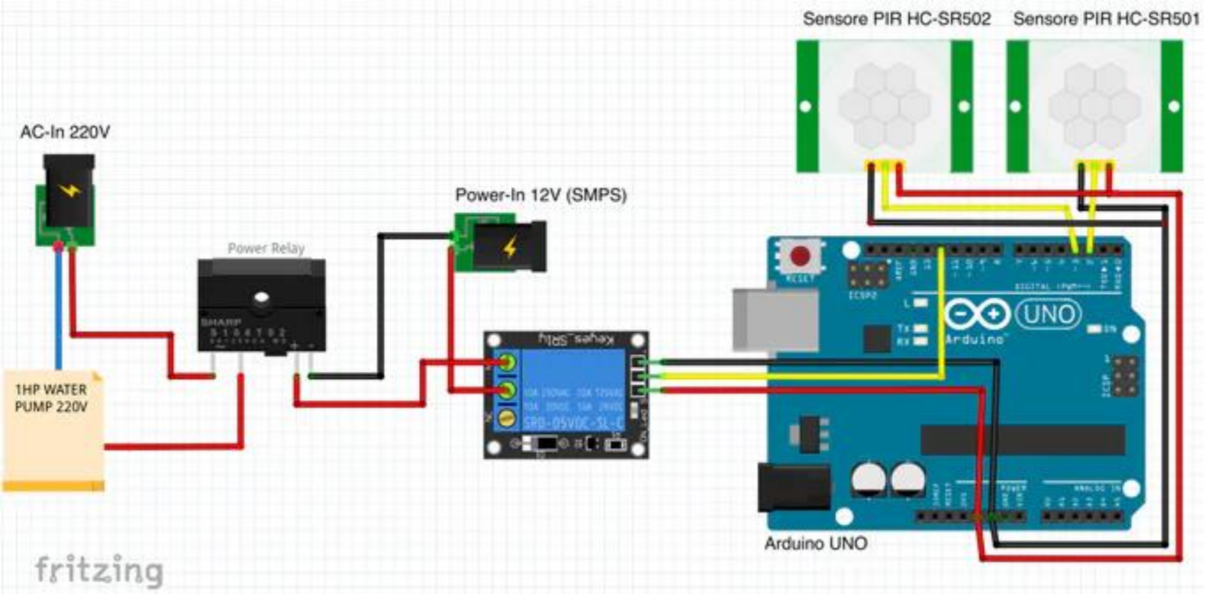
A plastic tank for 200 litres of solution and a pump for the high-pressure pipe system is located on the top of the vehicles which is connected with 4-Way Fogger Assembly. Micro tubbing pipes have been used throughout the vehicles to supply the solution to the fogger. Inside, there is a high-pressure pipeline with 4 4-Way Fogger which can be connected. The fog leaves no marks on clothes, while completely enveloping the incoming person and destroys the virus even in hard-to-reach folds of clothing and protects for some time after the exit.

Prepare the Arduino: -

The water pump actually works by detecting the infrared rays. Whenever there is no existence of human body comes near a motion sensor, the motion sensor detects this infrared ray and it gives us a HIGH signal through the output pin. This HIGH signal is then read by the Arduino. So, if Arduino reads a HIGH signal, it will give a HIGH signal to the relay module which means that the relay will turn on, and as a result the Power relay will turn on and it will turn on the Water Pump for 15 seconds (Can be changed). Similarly, if the Arduino reads a LOW signal, it will make the relay pin LOW and as a result the Water Pump will remain off.

In this I had used 2 PIR sensors to make it more precise if any of them sense the motion then the relay will turn on for 15 seconds (Can be changed).

We cannot use 5V Relay directly with the Water Pump because In my case, the water pump which I had used in this project has an Ampere(A) rating of 16Ampere and the 5V Relay has a max load of 10Ampere so to control the water pump I used one more relay with 5V Relay Module. which is 12V Power Relay.



Circuit Diagram

Code Section: -

```
/*
 * Automated Sanitize System
 */

int relayPin = 12;                // choose the pin for the Relay
Pin

int inputPin = 2;               // choose the input pin (for PIR
sensor)
int inputPin2 = 3;               // choose the input pin (for PIR
sensor 02)

int pirState = LOW;             // at start, assuming no motion
detected
int val = 0;                    // variable for reading the pin
status
int val2 = 0;                    // variable for reading the pin
status


void setup() {
  pinMode(relayPin, OUTPUT);      // declare Relay as output
  pinMode(inputPin, INPUT);     // declare sensor as input
  pinMode(inputPin2, INPUT);
  Serial.begin(9600);
}

void loop(){
  val = digitalRead(inputPin);  // read input value
  val2 = digitalRead(inputPin2);  // read input value
  if (val == HIGH || val2 == HIGH) {            // check if the
input is HIGH
    digitalWrite(relayPin, HIGH);  // turn Relay ON
    if (pirState == LOW) {
      // turned on
      Serial.println("Sanitization processing");
      // 15 sec delay
      delay(15000);
      pirState = HIGH;
    }
  } else {
    digitalWrite(relayPin, 0); // turn Relay OFF
    if (pirState == HIGH){
      // turned off
      Serial.println("Sanitization Completed");
      pirState = LOW;
    }
  }
}
```