IoT Project

Team Playbook

Team Members

Sriya Venepally

Mukta Chanda

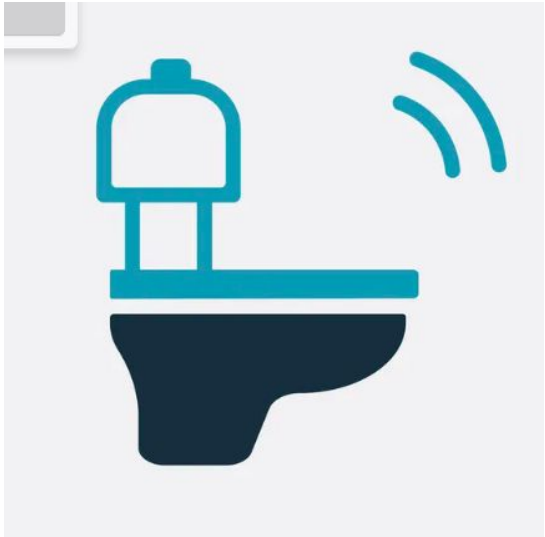
Anush Anand

Ronak Dhingra

Project Idea

A “smart” bathroom with automated light, flushing, and lid actuating mechanisms.

The bathroom can also be controlled via a mobile application.

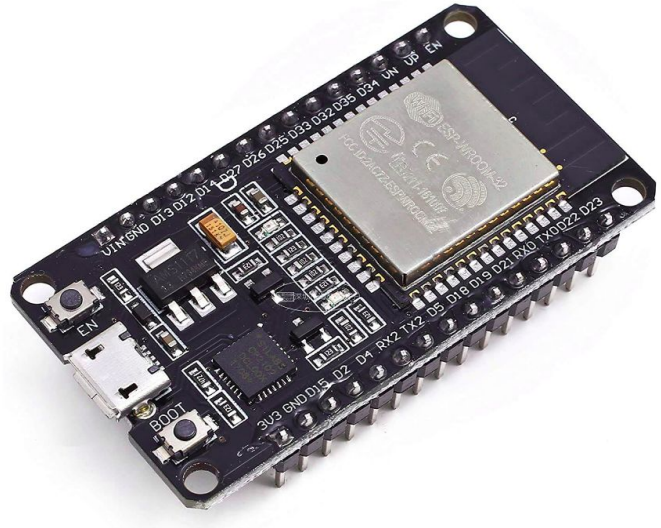


Why we chose this idea

People usually try to avoid public bathrooms as much as possible (for obvious hygiene reasons), but they still encounter situations where they are forced to use them. The purpose of the smart bathroom is to reduce the amount of contact between the user and the toilet so that their hygiene is maintained. In the wake of Covid pandemic, the need for such a concept is even greater.

Required components

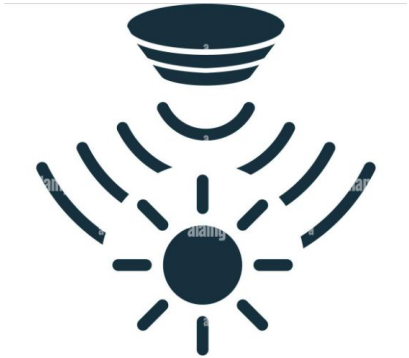
1. Ultrasonic Sensor
2. Water Level Sensor
3. Servo Motors
4. LEDs
5. ESP32 Module



Physical Quantities measured

Distance of user (in centimeters) from ultrasonic sensor.

Water level in flush tank (in millimeters) from water level sensor.



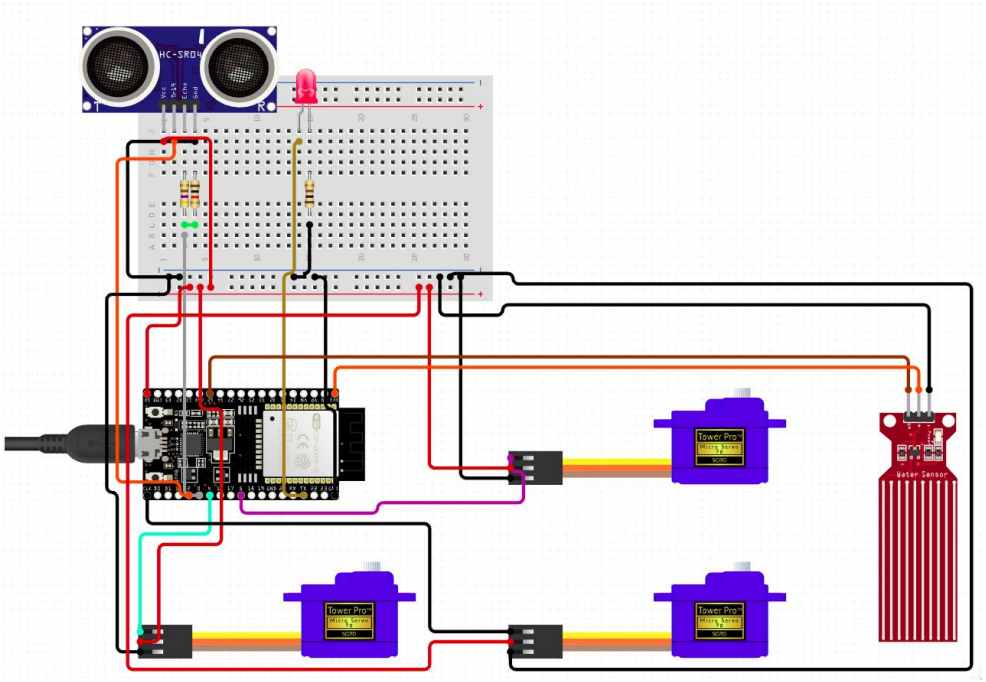
Using the inputs

1. Number of users
2. Number of flushes
3. Water Consumption

Output

Detailed statistical data about bathroom usage. We obtain the number of users, flushes and amount of water consumed over a given time interval. The organization also gets an alert to repair the toilet in the event that the flush does not work. The purpose of the output is to better monitor and manage the bathroom facilities.

Circuit Diagram



How it works

The proposed model makes use of an ultrasonic sensor to detect the user entering the bathroom, switches on the light and makes use of a servo motor to automatically lift the toilet lid. When the user leaves the bathroom, the light will switch back off, the lid will be closed, and the flush will get activated. In case the flush doesn't work, a message is sent to the bathroom organizer. Statistical data will be sent to the organization.

Code

```
#include <HTTPClient.h>
#include <WiFi.h>
#include "Servo.h"

#define trigpin 2   // yellow
#define echopin 4   // yellow
#define ledpin 32   // blue and brown
#define powerpin 13 // water level
#define sensorpin 34 //water level

#define servo_pin1 26   // blue and white
#define servo_pin2 27   // purple and brown

char* wifi_ssid = "Anush";
char* wifi_pwd = "12345678";
String cse_ip = "192.168.132.119";
String cse_port = "8080";
String server = "http://" + cse_ip + ":" + cse_port + "/~/in-cse/in-name/";
String ae = "IoT_Project";
String cnt1 = "People_Count";
String cnt2 = "Flush_Count";

HTTPClient http;

Servo servo_obj1;
Servo servo_obj2;

double prevdistance = 70;
int count_ppl=0;
int count_flush = 0;

int createCI1(String& val) {
  HTTPClient http;
  http.begin(server + ae + "/" + cnt1 + "/");

  http.addHeader("X-M2M-Origin", "admin:admin");
  http.addHeader("Content-Type", "application/json;ty=4");

  int code = http.POST("{\"m2m:cin\": {\"cnf\":\"application/json\",\"con\": " + String(val) + "}}");

  Serial.print("Response Code: ");
  Serial.println(code);
  if (code == -1) {
    Serial.println("Unable To Connect To The Server");
  }

  http.end();
}

int createCI2(String& val) {
  HTTPClient http;
  http.begin(server + ae + "/" + cnt2 + "/");

  http.addHeader("X-M2M-Origin", "admin:admin");
  http.addHeader("Content-Type", "application/json;ty=4");

  int code = http.POST("{\"m2m:cin\": {\"cnf\":\"application/json\",\"con\": " + String(val) + "}}");

  Serial.print("Response Code: ");
  Serial.println(code);
  if (code == -1) {
    Serial.println("Unable To Connect To The Server");
  }
```

Code

```
64
65    http.end();
66  }
67
68  void setup() {
69    Serial.begin(9600);
70    pinMode(echopin,INPUT);
71    pinMode(trigpin,OUTPUT);
72    pinMode(ledpin, OUTPUT);
73    pinMode(powerpin, OUTPUT);
74    digitalWrite(powerpin, LOW);
75    servo_obj1.attach(26);
76    servo_obj2.attach(27);
77    servo_obj2.write(45);
78
79      WiFi.begin(wifi_ssid,wifi_pwd);
80
81      while (WiFi.status() != WL_CONNECTED) {
82        delay(500);
83        Serial.print(".");
84      }
85    Serial.println("Succesfully connected to WiFi.");
86    Serial.println("IP address: ");
87    Serial.println(WiFi.localIP());
88    delay(10);
89  }
90
91  void loop() {
92    digitalWrite(trigpin, HIGH);
93    delayMicroseconds(10);
94    digitalWrite(trigpin,LOW);
95    long duration=pulseIn(echopin,HIGH);
```

```
96    double distance=duration*0.034/2;
97    Serial.print("Distance:");
98    Serial.print(distance);
99    Serial.println(" cm");
100     // user entering - open toilet lid and switch on the LED
101    if(distance < 25)
102    {
103     digitalWrite(ledpin, HIGH);
104
105     if(distance < 15)
106     {
107       servo_obj1.write(180);
108       if(prevdistance > 15)
109       {
110         count_ppl++;
111       }
112     }
113     else
114     {
115       servo_obj1.write(0);
116
117       // flush the toilet
118       if(prevdistance < 15)
119       {
120         servo_obj2.write(0);
121         delay(500);
122         servo_obj2.write(45);
123         delay(1000);
124         digitalWrite(powerpin, HIGH);
125         int reading = analogRead(sensorpin);
126         double waterlevel = (reading*40)/4096;
127         digitalWrite(powerpin, LOW);
```

Code

```
128         Serial.print("Water Level: ");
129         Serial.print(waterlevel);
130         Serial.println("mm");
131         if(waterlevel > 14)
132         {
133           Serial.println("Flush not working");
134         }
135         else
136         {
137           count_flush++;
138         }
139       }
140     }
141   }
142
143   // user leaving
144   else
145   {
146     // switch off LED and close the lid
147     digitalWrite(ledpin, LOW);
148     servo_obj1.write(0);
149
150     if(prevdistance < 15)
151     {
152       servo_obj2.write(0);
153       delay(500);
154       servo_obj2.write(45);
155       delay(1000);
156
157       digitalWrite(powerpin, HIGH);
158       int reading = analogRead(sensorpin);
159       double waterlevel = (reading*40)/4095;
```

```
160       digitalWrite(powerpin, LOW);
161       Serial.print("Water Level: ");
162       Serial.print(waterlevel);
163       Serial.println("mm");
164
165       if(waterlevel > 14)
166       {
167         Serial.println("Flush not working");
168       }
169       else
170       {
171         count_flush++;
172       }
173
174     }
175   }
176   prevdistance = distance;
177
178   String val1 = String(count_ppl);
179   createCI1(val1);
180
181   String val2 = String(count_flush);
182   createCI2(val2);
183   delay(1000);
184 }
```

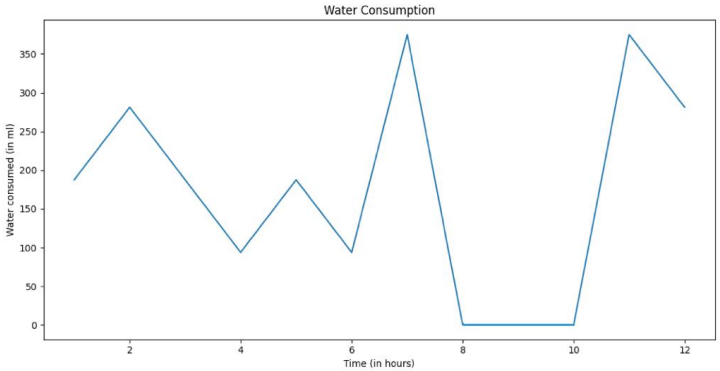
Objectives

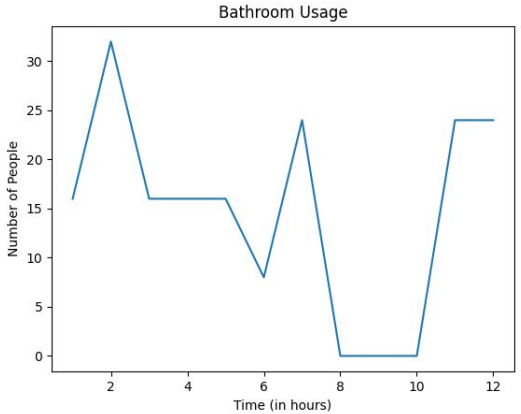
1. User detection
2. Turn on lights
3. Automatically open and close the toilet seat lid
4. When user leaves, automatically activate the flush
5. Send a message to the organizer in case the flush doesn’t work

Implementation

1. User’s entry/exit to and from the toilet is measured using ultrasonic sensor.
2. Water level in flush tank is measured using water level sensor.
3. Data is stored in onem2m server using WiFi client.
4. Mobile Application to control bathroom is implemented using Bluetooth client.

Graphs - For 29th June





Data Analysis

The data in the first graph represents the number of people visiting the bathroom. It is updated hourly. For example, at time 2:00 pm, the number of people is 30. This means that 30 people used the bathroom from 1:00 - 2:00 pm.

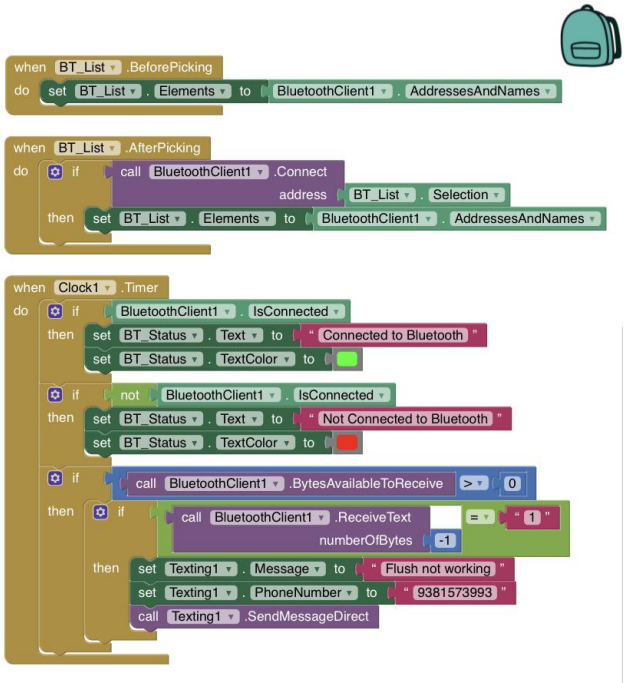
Similarly, the data in the second graph represents the water consumed in ml. It is updated hourly. For example, at time 2:00 pm, the water consumed is approximately 275 ml. This means that 275 ml of water was consumed from 1:00 - 2:00 pm.

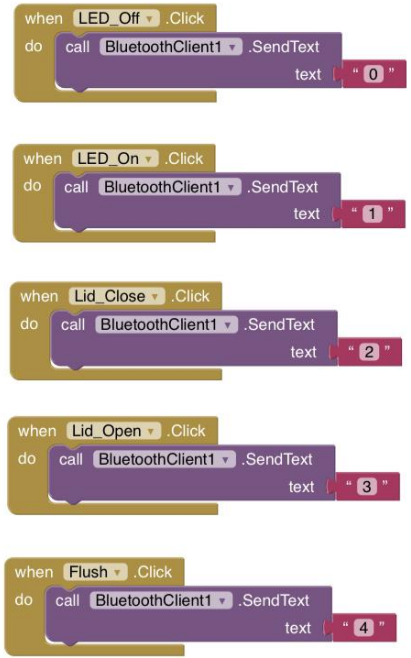
App - Playbook

The app provides separate buttons for turning on and off the lights, opening and closing the toilet seat lid, and for flushing the toilet.

It also has automatic messaging capability if it detects that there is no change in water level in flush tank.

App Code


when BT_List .BeforePicking
do set BT_List . Elements to BluetoothClient1 . AddressesAndNames
when BT_List .AfterPicking
do if call BluetoothClient1 .Connect
address BT_List . Selection
then set BT_List . Elements to BluetoothClient1 . AddressesAndNames
when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then set BT_Status . Text to " Connected to Bluetooth "
set BT_Status . TextColor to
if not BluetoothClient1 . IsConnected
then set BT_Status . Text to " Not Connected to Bluetooth "
set BT_Status . TextColor to
if call BluetoothClient1 .BytesAvailableToReceive > 0
then if call BluetoothClient1 .ReceiveText = " 1 "
numberOfBytes -1
then set Texting1 . Message to " Flush not working "
set Texting1 . PhoneNumber to " 9381573993 "
call Texting1 .SendMessageDirect


when LED_Off .Click
do call BluetoothClient1 .SendText
text " 0 "
when LED_On .Click
do call BluetoothClient1 .SendText
text " 1 "
when Lid_Close .Click
do call BluetoothClient1 .SendText
text " 2 "
when Lid_Open .Click
do call BluetoothClient1 .SendText
text " 3 "
when Flush .Click
do call BluetoothClient1 .SendText
text " 4 "

Challenges

The PIR Sensor was not suitable for our requirements.

How we overcame the issue:

PIR sensors detect occupants' presence by sensing the difference between heat emitted by moving people and background heat.

Ultrasonic sensors detect the presence of people by sending out ultrasonic sound waves into a space and measuring the speed at which they return. They look for frequency changes caused by a moving person. So it can detect the user behind obstacles also.

Ultrasonic sensors, meanwhile, are highly suitable for spaces in which a line of sight is not possible, such as partitioned spaces, and in spaces requiring a higher level of sensitivity. Examples of such spaces include bathrooms.

Challenges

Difficulty with faulty components and connections.

Showing the implementation of lid and flush using cardboard in the prototype.

Bluetooth and WiFi clients can’t work simultaneously.

Challenges

Corrosivity of water level sensor.

How we overcame the issue:

If power is applied to the sensor constantly, it speeds the rate of corrosion. To overcome this, we made sure to not power the sensor constantly, and power it only when we were taking measurements. We connected the VCC pin to another GPIO pin and set it to high and low in our code.

Application Scenario

Measuring data like number of flushes and users, and amount of water and electricity consumption is beneficial to organizations to monitor and control bathroom facilities. For example, pay toilets would want to keep track of the number of users in a given day.

Automatic flush system

The sensor situated exactly above the seat gets activated based on person's movement. A controllable flushing system using a microcontroller based circuit is then activated. A solenoid operated water valve is used to actuate the flush system. When a person comes near the IR LEDS, the IR Receiver module receives the reflected signal and alert LED lights up. The circuit makes use of buffers, RST pins, capacitors and diodes.

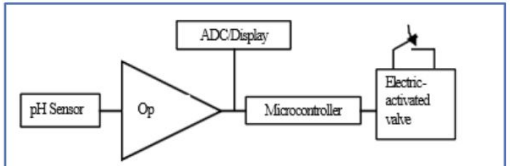


Fig. 1: A block diagram of the proposed smart flushing system.

Figure 2: Installation of Automatic Flush system

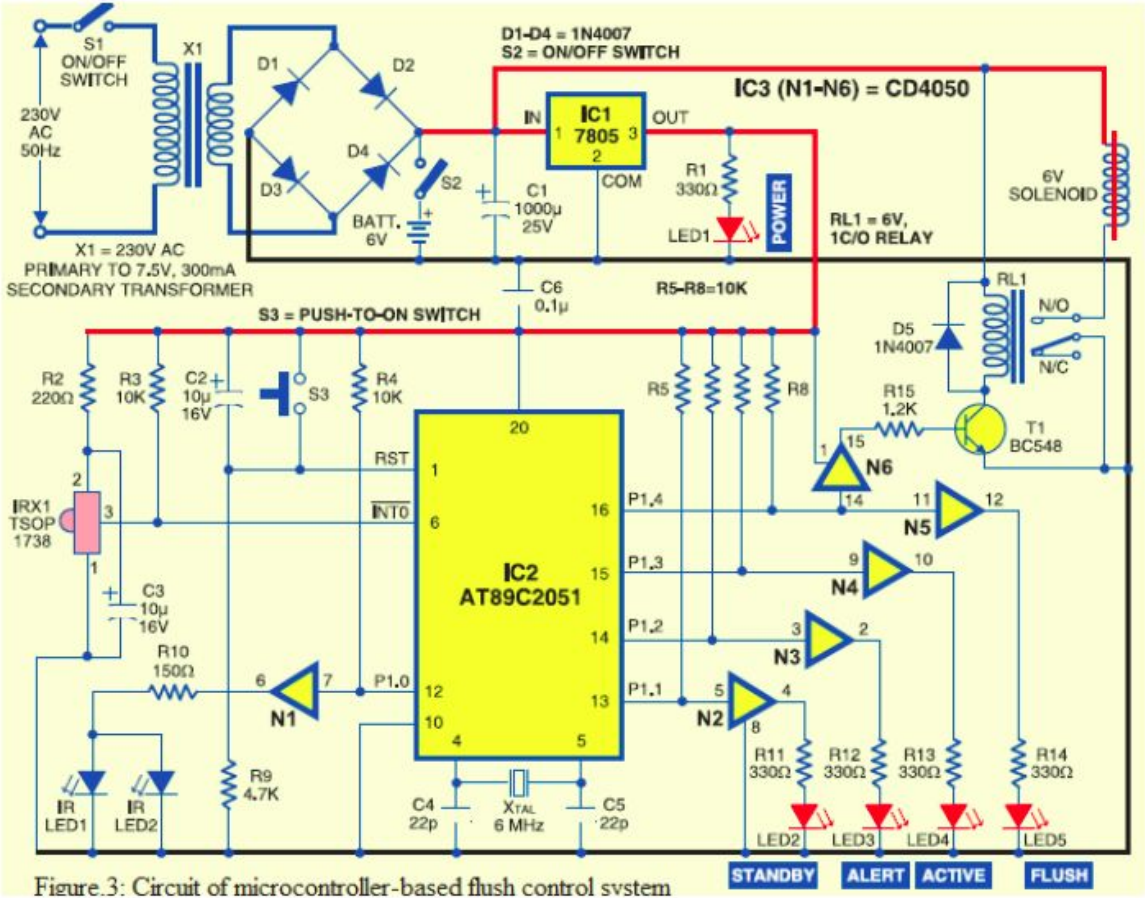
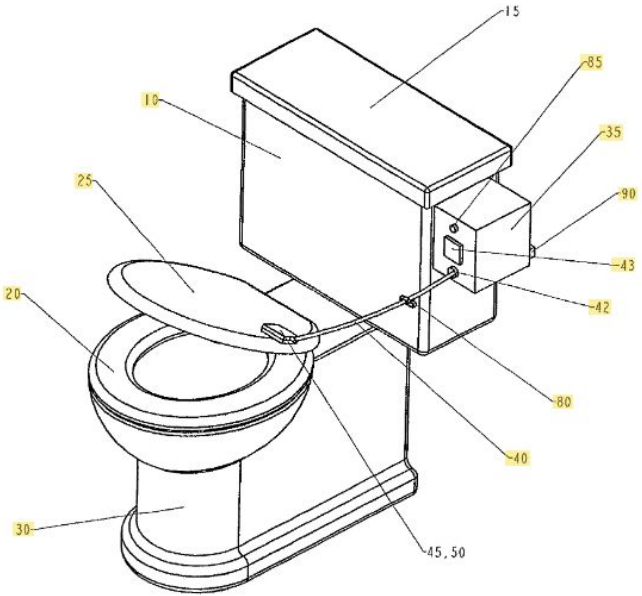


Figure.3: Circuit of microcontroller-based flush control system

Automatic toilet seat system

A toilet seat lid raising and lowering device relies on a motion sensor to trigger its operation. A housing containing the sensor, a motor and a spool is removably attached to a toilet tank. Upon detecting a user approaching the toilet area, the sensor sends an activation signal to the motor causing the motor to rotate the spool. A line is attached to the spool at one end and to the seat lid at a second end such that the rotation of the spool causes the line to raise the seat lid to an open position slightly angled from the tank. Once a user exits the toilet area, the motor returns to a neutral position allowing the spool to rotate in an opposite direction thereby lowering the seat lid by means of gravity. Resistance between the motor and spool prevent the seat lid from “crashing” to a closed position.



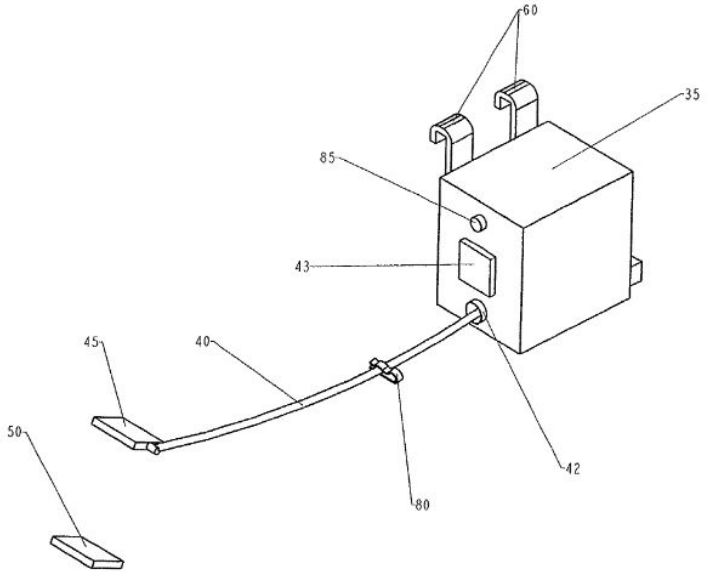


FIG. 3

References

Research Paper: DESIGN AND IMPLEMENTATION OF AUTOMATIC FLUSH SYSTEM FOR SANITATION IN PUBLIC TOILETS

Link: https://ijrbat.in/upload_papers/03122015081500ET%2013.pdf

Patent: Automatic toilet seat lid raising and lowering device

Link: https://patents.google.com/patent/US20030154541A1/en

Thank you