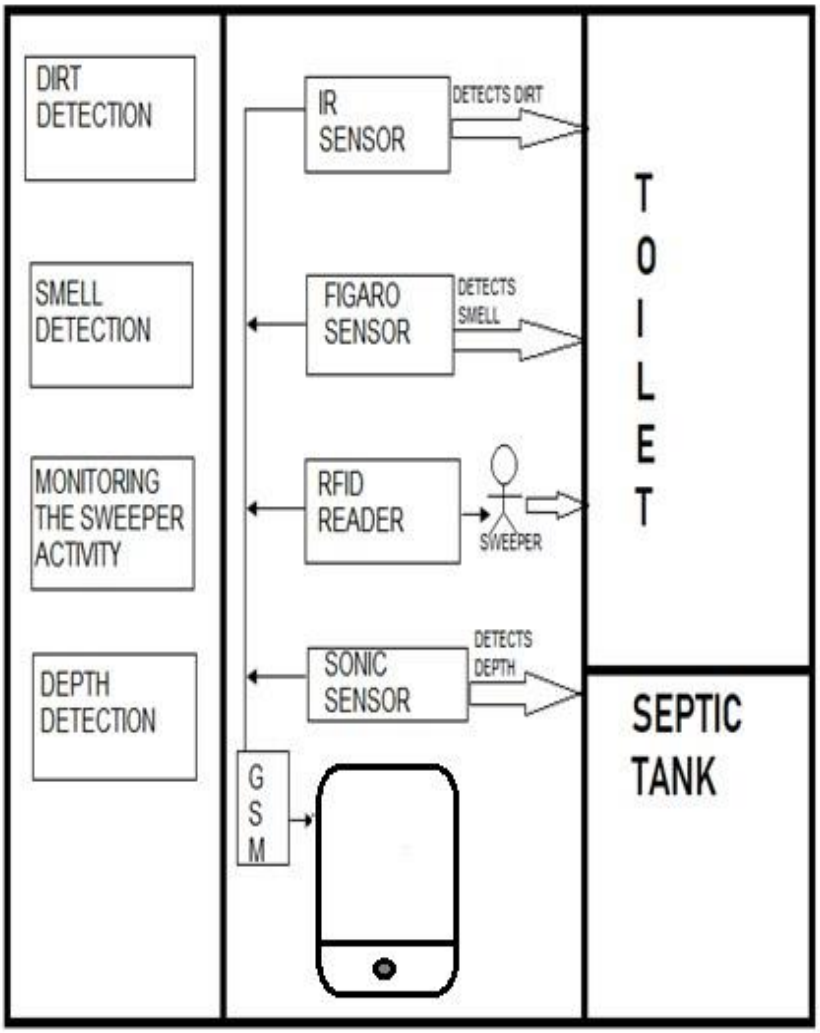
SMART PUBLIC RESTROOM

3.1.1 ARCHITECTURE OF PROPOSED SYSTEM



SYSTEM ENVIRONMENT

4.1 Hardware Requirement

Hardware tools are required:

- Microcontroller PIC 16F877
- LCD Display
- Power supply
- Buzzer
- Infrared sensor
- Sonic sensor
- Gas sensor
- RFID
- GSM modem

4.2Software Requirement

- Embedded C programming

SYSTEM DESIGN

5.1 IOT SYSTEM DESIGN:

5.1.1 Microcontroller

A microcontroller is a small computer on a single combined circuit holding a processor core, memory and programmable input/output peripherals. Program memory in the form of Ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general-purpose applications.

Figure 2: Microcontroller

PIC 16F877 is one of the most advanced microcontroller from Microchip. This controller is commonly used for experimental and modern applications because of its low price, wide range of requests, high quality, and ease of obtainability. It is ideal for applications such as machine control applications, measurement devices, study purpose, and so on. The PIC 16F877 features all the mechanisms which present microcontrollers usually have.

5.1.2 LCD

LCD stands for Liquid Crystal Display. By using the LCD, all the outputs are displayed. LCD doesn't know about the content (data or commands) supplied to its data bus. It is the user who has to specify whether the content at its data pins are data or commands.

Figure 3: LCD Display

For this, if a command is inputted then a certain arrangement of 0's and 1's has to be applied to the Control lines so as to specify it is a command on the other hand if a data is inputted at the data lines then an another combination of 0s and 1s has to be applied to the control lines to require it is Data.

5.1.3 BUZZER

Buzzer is also called as Beeper. It is a sound signalling mechanical device.

Figure 4: Buzzer

5.1.4 INFRARED SENSOR

The IR sensor is used to detect the dirt present in the toilet. Here we nourish the image models into the sensor. It can perceive the dirt by comparing the images we feed into it, after using the toilet. If it can detect the dirt, it raises the alarm, and the users may get embraced and they clean it. This system can create the responsiveness among the people.

Figure 5: IR sensor

5.1.5 SMELL SENSOR

The Smell Sensor is used to detect the unwanted smell and gases in the toilet. For this purpose, we are going to use the sensor called **Figaro** sensor.

Figure 6: Smell Sensor

It can intellect the dry gases present in the toilets such as NH_3, CO_2, CH_4, H_2S, etc. By taking those gases leads to Nausea, Drowsiness, instant loss of awareness, etc. After sensing the unwanted gases, it can blink the red light. Then the sweeper can clean it by using particular Cleaning Agents.

5.1.6 RFID READER

The RFID stands for Radio Frequency Identification. It can be used for monitoring the Sweeper. The Organization wishes to provide the identity tag for the Sweeper. The Sweeper desires to show the tag before the cleaning process is going to start and after it is finished.

Figure 7: RFID Reader

Then the CR4 sensor can spot the presence of dirt. If it is present, it can blink the red light. If it is clean, it can blink the blue light. It assistances to understand the responsibilities of sweeper by his/her own. If Sweeper is not clean the toilets for period of time, his/her absence in cleaning the toilet also reported to the dependable organization. These all the details are stored in the database.

5.1.7 SONIC SENSOR

The Sonic Sensor is used for computing the depth. Here it is used to measure the depth of the septic tank. The Sonic Sensor is fixed into the Septic tank. Then the Septic tank get filled means, it can sends the communications to particular organization. Then they will allot persons to clean the septic tank. Then septic tank cleaners will clean the tank. After cleaning it, the sensor can detect the level, and send messages to consistent organization.

Figure 8: Sonic sensor

This ultrasonic sensor can be used for measuring distance, object sensor, motion sensors etc. High sensitive module can be used with microcontroller to integrate with motion circuits to measure the distance, position & motion sensitive products.

In a nutshell, water depth sensing is using a sensor to measure the depth of water in a tank or container. Although various sensors can be used for this application, we will talk about ultrasonic sensor application.With ultrasonic sensors, we can find the water depth calculation by finding the distance between the transceiver and the surface of the water. The sensor will transmit a short ultrasonic pulse, and we can measure the travel time of that pulse to the liquid and back. We can then subtract that distance from the total depth of the tank to determine the water depth.

2.3.1GSM

GSM stands for Global System for Mobile communication. It establishes the mobile communication from one place to another place.

Figure 10: GSM Module

It transfers the information from main circuit to operator. It uses Time Division Multiple Access (TDMA).

GSM is mainly used for communicating and transferring message from one person to concerned organisation. GSM module is used to establish communication between a computer and a GSM and GPRS system.

Here we are using GSM LT-2 communication module makes it possible to use GSM paths to provide monitoring and messaging functions in alarm systems. It facilitates cooperation with SATEL and third party control panel diallers or correctly configured outputs.

He GSM LT-2 module makes it possible to implement monitoring as well as text and voice messaging functions. The caller ID retransmission function creates it likely to present the incoming callers number on telecommunication stations armed with this functionality. GSM alarm system built-in GSM communication module inside, work as a mobile handset. After purchased the GSM alarm system, people need to acquisition the SIM card, and select the mobile service package. GSM alarm system can program several phone numbers for alarm receiving. When any abnormal event happens, the system will response, then inform the owner via voice call and short message (SMS).

GSM will check the messaging activities for sweepers and also need to check with their cleanliness duty for their work. The sweepers need to check with particular activity of its work by their sensors.

BLOCK DIAGRAM OF PROPOSED SYSTEM:

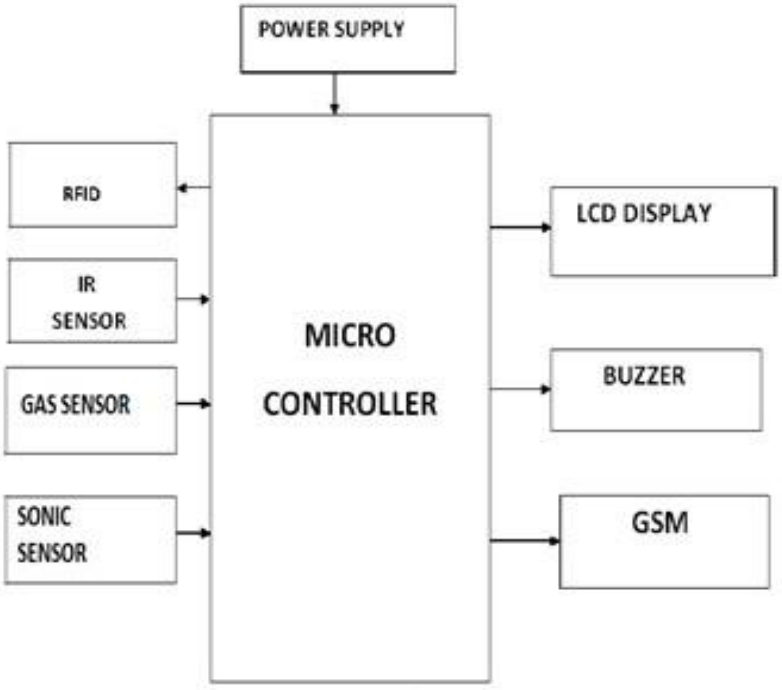


Figure11:Block diagram of proposed system

APPENDIX
SOURCE CODE

```
import requests
import time
import serial
import re
import urllib2
import datetime
import os
import sys
import RPi.GPIO as GPIO
import tm1637
ser = serial.Serial(
    port='/dev/ttyAMA0',
    baudrate=9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout=1
)
ser.flushInput()


CONT = 1
send = 0
```

```
string1 = '{"$id":"Group6_smart_toilet","room_temp":'
string2 = ',"room_lux":'
string3 = ',"slot1_num_of_p":'
string4 = ',"slot1_lux":'
string5 = ',"slot2_num_of_p":'
string6 = ',"slot2_lux":'
string7 = ',"slot3_num_of_p":'
string8 = ',"slot3_lux":'

Display = tm1637.TM1637(23, 24, tm1637.BRIGHT_TYPICAL)

Display.Clear()
Display.SetBrightnes(5)

a = 0
b = 0
c = 0
d = 0

while 1:

   try:
      data_string = ser.readline()    #DL read data from node 1
```

```
        data_num = re.findall('\d+(?:\.\d+)?', data_string) #DL divide string data and allocate to each data array
        NID = data_num[1]
    except:
        print "serial read error and will try again"
        CONT = 1
        send = 0
        continue    #DL when data fail to send the data, this loop return to first of while loop

    try:
        if (NID == "1") & (CONT == 1):  #YL pass the data from board 1 firstly
            Temperature1 = data_num[3]  #YL assign value to the Temperature1 from the data_num[3]
            Light1 = data_num[5]        #YL assign value to the Light1 from the data_num[5]
            if (data_num[7] == "1") & (len(data_num[3]) < 9) & (len(data_num[5]) < 9):#YL receive data only when the CRC=1 and correct length of data
                print "NODE :", NID, "Temperature :", Temperature1, "   Light :", Light1
                CONT = CONT + 1         #YL Self-added 1 to make sure received data is from board 2

        elif (NID == "2") & (CONT == 2):#YL pass the data from board 2 secondly
```

```
        User1 = data_num[3]        #YL value the User1 from the
data_num[3]

        Light2 = data_num[5]       #YL  value the Light2 from the
data_num[5]

        if (data_num[7] == "1") & (len(data_num[3]) < 9) &
(len(data_num[5]) < 9):#YL receive data only when the CRC=1 and
correct length of data

            print "NODE :", NID, "Number of people :", User1, "   Light :",
Light2

            CONT = CONT + 1

    elif (NID == "3") & (CONT == 3):#YL pass the data from board 3
thirdly

        User2 = data_num[3]        #YL value the User2 from the
data_num[3]

        Light3 = data_num[5]       #YL  value the Light3 from the
data_num[5]

        if (data_num[7] == "1") & (len(data_num[3]) < 9) &
(len(data_num[5]) < 9):#YL receive data only when the CRC=1 and
correct length of data

            print "NODE :", NID, "Number of people :", User2, "   Light :",
Light3

            CONT = CONT + 1

    elif (NID == "4") & (CONT == 4):#YL pass the data from board 4 in
the end

        User3 = data_num[3]        #YL value the User3 from the
data_num[3]

        Light4 = data_num[5]       #YL  value the Light4 from the
data_num[5]
```

```
        if (data_num[7] == "1") & (len(data_num[3]) < 9) &
(len(data_num[5]) < 9):#YL receive data only when the CRC=1 and
correct length of data

            print "NODE :", NID, "Number of people :", User3, "   Light :",
Light4

            CONT = CONT + 1

            send = 1              #YL  value the send=1 when a group of
data is received


      elif NID == "5":              #YL  CRC=0

         print "CRC incorrect (You really need to check your CRC)"


      elif (len(data_num[3]) > 9) & (len(data_num[5]) > 9):#YL a series
of incorrect and long length data will stop the program

         print "You realy need reboot your microcontrolers :", NID

         print "But program will try again "

         CONT = 1

         send = 0              #YL  initialization of send signal

         continue

   #JL the Indexerror usually came when the raspberrypi read the
data,

   #  it does not read it from the beginning of each print out but from
the middle of it

   except IndexError:

      print 'Index error, will flush serial and try again' #YL give a hint as
the data stop presenting if IndexError exist
```

```
        ser.flushInput()


    try:
        json_string = string1 + str(Temperature1) + string2 + str(Light1) \
                    + string3 + str(User1) \
                    + string4 + str(Light2) \
                    + string5 + str(User2) \
                    + string6 + str(Light3) \
                    + string7 + str(User3) \
                    + string8 + str(Light4) + '}'
    except:
        print('Json_string incorrect Read again')
        continue


    if send == 1:
        try:    #DL upload on cloud(devicepilot) to use a specific url
            resp = requests.post("https://api.devicepilot.com/devices",
                                 headers={"Authorization": "Token
4f07894ac4e9351140f190d3d0dc0696",
                                          "Content-Type": "application/json"},
                                 data=json_string)


        except: #DL when data fail to upload on the cloud, this loop
return to first of while loop
```

```
        print "There is a Connection error or error while sending data
to cloud"
        continue


    print 'Data have been ' + resp.reason + ' by cloud'
    print (datetime.datetime.now())


    int_user1 = int(float(User1))
    int_user2 = int(float(User2))
    int_user3 = int(float(User3))


    sum_of_p = int_user1 + int_user2 + int_user3


    if sum_of_p<10:
        d = sum_of_p
    if sum_of_p > 9:
        x = str(sum_of_p)
        c = int(x[0])
        d = int(x[1])
    if sum_of_p > 99:
        x = str(sum_of_p)
        b = int(x[0])
        c = int(x[1])
        d = int(x[2])
```

```
if sum_of_p > 999:
    x = str(sum_of_p)
    a = int(x[0])
    b = int(x[1])
    c = int(x[2])
    d = int(x[3])

display_num = [a, b, c, d]
Display.Show(display_num)

print str(sum_of_p) + ' People had used this toilet'

print '==================================='
print 'Program will Sleep 2 second '
time.sleep(2)
ser.flushInput()
print 'Serial Flushed'
os.system('clear')
CONT = 1
send = 0
a = 0
b = 0
c = 0
d = 0
```