|  |
| --- |
| Difference Between NAAC & NBA Accreditation - Haq Se EngineerPREC LONIJai Shriram Engineering College (@JSREC09) / Twitter**JAI SHRIRAM ENGINEERING COLLEGE**  **TIRUPPUR – 638 660**  Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai  Recognized by UGC & Accredited by NAACandNBA (CSE and ECE) |

**DEPARTMENT OF**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**IBM - Naan Mudhalvan**

**Internet of Things**

**Group 3**

**Phase 3- Project Submission**

**NAME : MANIVEL SHANKAR P**

**NM ID :AU711221106015**

**YEAR : III**

**Smart Public Restroom**

**Python code for Airquality monitering:**

import time

import board

import busio

from digitalio import DigitalInOut, Direction, Pull

from adafruit\_pm25.i2c import PM25\_I2C

reset\_pin = None

i2c = busio.I2C(board.SCL, board.SDA, frequency=100000)

pm25 = PM25\_I2C(i2c, reset\_pin)

print("Found PM2.5 sensor, reading data...")

while True:

time.sleep(1)

try:

aqdata = pm25.read()

except RuntimeError:

print("Unable to read from sensor, retrying...")

continue

print()

print("Concentration Units (standard)")

print("---------------------------------------")

print( "PM 1.0: %d\tPM2.5: %d\tPM10: %d" ,aqdata["pm10 standard"], aqdata["pm25 standard"], aqdata["pm100 standard"])

print("Concentration Units (environmental)")

print("---------------------------------------")

print("PM 1.0: %d\tPM2.5: %d\tPM10: %d",aqdata["pm10 env"], aqdata["pm25 env"], aqdata["pm100 env"])

print("---------------------------------------")

print("Particles > 0.3um / 0.1L air:", aqdata["particles 03um"])

print("Particles > 0.5um / 0.1L air:", aqdata["particles 05um"])

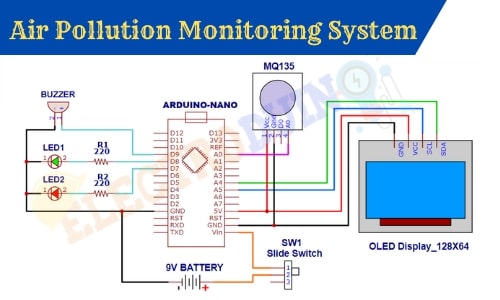
print("Particles > 1.0um / 0.1L air:", aqdata["particles 10um"])

print("Particles > 2.5um / 0.1L air:", aqdata["particles 25um"])

print("Particles > 5.0um / 0.1L air:", aqdata["particles 50um"])

print("Particles > 10 um / 0.1L air:", aqdata["particles 100um"])

print("---------------------------------------")



**Python code for Automatic flusher:**

from gpiozero import DistanceSensor

ultrasonic = DistanceSensor(echo=17, trigger=4)

while True:

print(ultrasonic.distance)

while True:

ultrasonic.wait\_for\_in\_range()

print("In range")

ultrasonic.wait\_for\_out\_of\_range()

print("Out of range")

ultrasonic = DistanceSensor(echo=17, trigger=4, threshold\_distance=0.5)

ultrasonic.threshold\_distance = 0.5

……………………………………………………

def hello():

print(“Hello”)

print(“Turn ON the flusher”)

ultrasonic.when\_in\_range = hello

…………………………………………..

def bye():

print("Bye")

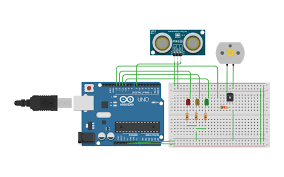
print(“Turn OFF the flusher”)

ultrasonic.when\_out\_of\_range = bye

………………………………………………………

ultrasonic = DistanceSensor(echo=17, trigger=4, max\_distance=2)

ultrasonic.max\_distance = 2



**Python code for User Availability:**

from gpiozero import DistanceSensor

ultrasonic = DistanceSensor(echo=17, trigger=4)

while True:

print(ultrasonic.distance)

while True:

ultrasonic.wait\_for\_in\_range()

print("In range")

ultrasonic.wait\_for\_out\_of\_range()

print("Out of range")

ultrasonic = DistanceSensor(echo=17, trigger=4, threshold\_distance=0.5)

ultrasonic.threshold\_distance = 0.5

……………………………………………………

def ledon ():

print(“Turn ON the LED”)

ultrasonic.when\_in\_range = ledon

…………………………………………..

def ledoff():

print(“Turn OFF the LED”)

ultrasonic.when\_out\_of\_range = ledoff

………………………………………………………

ultrasonic = DistanceSensor(echo=17, trigger=4, max\_distance=2)

ultrasonic.max\_distance = 2