**SOURCE CODE:**

PYTHON CODE FOR CONNECTING WITH IBM CLOUD:

import wiotp.sdk.device

import time

import random

import requests,json

import datetime

import time

myConfig = {

"identity": {

"orgId": "8pqusj",

"typeId": "tamilss",

"deviceId":"tamilsst"

},

"auth": {

"token": "V?quzXKKZ5P6(u\*\*L6"

}

}

BASE\_URL = "https://api.openweathermap.org/data/2.5/weather?"

CITY = "Coimbatore"

API\_KEY = "46faa4ab6fede1d9ae549b90d91253f2"

URL = BASE\_URL + "q=" + CITY + "&appid=" + API\_KEY

response = requests.get(URL)

if response.status\_code == 200:

data = response.json()

main = data['main']

temp = round(main['temp'] - 273,2)

humy = main['humidity']

pres = main['pressure']

rept = data['weather']

report = rept[0]['description']

# #time = datetime.datetime.now()

# morning = time.replace(hour=11, minute=59, second=0, microsecond=0)

# if time <= morning:

# me = '8.30 AM - 9.30 AM'

# else:

# me = '3.45 PM - 5.00 PM'

def myCommandCallback(cmd):

print("Message received from IBM IoT Platform: %s" % cmd.data['command'])

m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

client.connect()

while True:

temp1=temp - 273

if(temp1 < 14):

temp\_alert = "snow may occur"

hum=humy

if(hum>95):

hum\_alert2 = "rain will occur for sure"

elif (hum > 90):

hum\_alert1 = "chances for raining is high"

pre = pres

wea = report

if(wea == "clear sky"):

wea\_alt1 = "possibility of raining is low"

elif(wea == "few clouds"):

wea\_alt2 = "drive safely!"

elif(wea == "shower rain"):

wea\_alt3 = "rain is showering in less amount"

elif(wea == "rain"):

wea\_alt4 = "its raining"

elif(wea == "thunderstorm"):

wea\_alt5 = "thunderstorm alert, don't panic, drive slow and safe!"

elif (wea == "snow"):

wea\_alt6 = "snow occurs, chance of skidding is more,drive safely!!"

elif(wea == "mist"):

wea\_alt7 = "mist is formed, switch on the foglamp"

else:

good\_alt = "good day!! Drive safely!!"

myData={'location':CITY,'temperature':temp, 'humidity':hum, 'pressure':pre, 'weather\_report':wea}

client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

print("Published data Successfully: %s", myData)

client.commandCallback = myCommandCallback

time.sleep(2)

client.disconnect()

**CODE FOR INTERFACING GSM AND SENSORS WITH ARDUINO**

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

SoftwareSerialGPS\_SoftSerial(4, 3);

TinyGPSPlusgps;

volatile float minutes, seconds;

volatile int degree, secs, mins;

void setup() {

Serial.begin(9600);

GPS\_SoftSerial.begin(9600);

}

void loop() {

smartDelay(1000);

unsigned long start;

double lat\_val, lng\_val, alt\_m\_val;

bool loc\_valid, alt\_valid;

lat\_val = gps.location.lat();

loc\_valid = gps.location.isValid();

lng\_val = gps.location.lng();

alt\_m\_val = gps.altitude.meters();

alt\_valid = gps.altitude.isValid();

if (!loc\_valid)

{

Serial.print("Latitude : ");

Serial.println("\*\*\*");

Serial.print("Longitude : ");

Serial.println("\*\*\*");

delay(4000);

}

else

{

Serial.println("GPS READING: ");

DegMinSec(lat\_val);

Serial.print("Latitude in Decimal Degrees : ");

Serial.println(lat\_val, 6);

DegMinSec(lng\_val);

Serial.print("Longitude in Decimal Degrees : ");

Serial.println(lng\_val, 6);

delay(4000);

}

}

static void smartDelay(unsigned long ms)

{

unsigned long start = millis();

do

{

while (GPS\_SoftSerial.available())

gps.encode(GPS\_SoftSerial.read());

} while (millis() - start <ms);

}

void DegMinSec( doubletot\_val)

{

degree = (int)tot\_val;

minutes = tot\_val - degree;

seconds = 60 \* minutes;

minutes = (int)seconds;

mins = (int)minutes;

seconds = seconds - minutes;

seconds = 60 \* seconds;

secs = (int)seconds;

}

**Code for interfacing sensors with Arduino**

#include "dht.h"

#define dht\_apin A0

dht DHT;

int vib\_pin=7;

int led\_pin=13;

void setup() {

Serial.begin(9600);

Serial.println("DHT11 Humidity & temperature Sensor\n\n");

delay(1000);

pinMode(vib\_pin,INPUT);

pinMode(led\_pin,OUTPUT);

}

void loop() {

DHT.read11(dht\_apin);

Serial.print("Current humidity = ");

Serial.print(DHT.humidity);

Serial.print("% ");

Serial.print("temperature = ");

Serial.print(DHT.temperature);

Serial.println("C ");

int val;

val=digitalRead(vib\_pin);

if(val==1)

{

digitalWrite(led\_pin,LOW);

}

else

digitalWrite(led\_pin,HIGH);

}