**SMART FARMING USING IOT**

**SPRINT DELIVERY-1**

**TEAM ID:**  **PNT2022TMID49187**

**CODES:**

import time

import sys import

ibmiotf.application import

ibmiotf.device import

random

#details of IBM Watson Device Credentials

organization = "157uf3" deviceType = "abcd"

deviceId = "7654321" authMethod = "token"

authToken = "87654321"

def myCommandCallback(cmd): print("Command

received: %s" % cmd.data['command'])

status=cmd.data['command'] if status=="motoron":

print ("motor is on") elif status == "motoroff": print

("motor is off") else :

print ("please send proper command")

try:

deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,

"auth-method": authMethod, "auth-token": authToken}

deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:

print("Caught exception connecting device: %s" % str(e))

sys.exit()

event of type "greeting" 10 times deviceCli.connect()

while True:

#Get Sensor Data from DHT11

temp=random.randint(90,110)

Humid=random.randint(60,100)

Mois=random.randint(20,120)

data = { 'temp' : temp, 'Humid': Humid, 'Mois' :Mois}

myOnPublishCallback():

print ("Published Temperature

= %s C" % temp, "Humidity = %s

%%" % Humid, "Moisture =%s

deg c" %Mois, "to IBM

Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,

on\_publish=myOnPublishCallback) if not success: print("Not connected

to IoTF") time.sleep(10)

deviceCli.commandCallback = myCommandCallback

**AURDINO CODE IN C:**

#include <dht.h>

#include <SoftwareSerial.h>

#define dht\_apin A0

SoftwareSerial mySerial(7,8);

const int sensor\_pin = A1;

int pin\_out = 9;

dht DHT;

int c=0;

void setup()

{

pinMode(2, INPUT);

pinMode(3, OUTPUT);

pinMode(9, OUTPUT);

}

void loop()

{

if (digitalRead(2) == HIGH)

{

digitalWrite(3, HIGH);

delay(10000);

digitalWrite(3, LOW);

delay(100);

}

Serial.begin(9600);

delay(1000);

DHT.read11(dht\_apin);

float h=DHT.humidity;

float t=DHT.temperature;

delay(5000);

Serial.begin(9600);

float moisture\_percentage;

int sensor\_analog;

sensor\_analog = analogRead(sensor\_pin);

moisture\_percentage = ( 100 - ( (sensor\_analog/1023.00) \* 100 ) );

float m=moisture\_percentage;

delay(1000);

if(m<40)

{

while(m<40)

{

digitalWrite(pin\_out,HIGH);

sensor\_analog = analogRead(sensor\_pin);

moisture\_percentage = ( 100 - ( (sensor\_analog/1023.00) \* 100 ) );

m=moisture\_percentage;

delay(1000);

}

digitalWrite(pin\_out,LOW);

}

if(c>=0)

{

mySerial.begin(9600);

delay(15000);

Serial.begin(9600);

delay(1000);

Serial.print("\r");

delay(1000);

Serial.print("AT+CMGF=1\r");

delay(1000);

Serial.print("AT+CMGS=\"+XXXXXXXXXX\"\r");

delay(1000);

Serial.print((String)"update>"+(String)"Temprature="+t+(String)"Humidity="+h+(String)"Moisture="+m);

delay(1000);

Serial.write(0x1A);

delay(1000);

mySerial.println("AT+CMGF=1");

delay(1000);

mySerial.println("AT+CMGS=\"+XXXXXXXXXX\"\r");

mobile number

delay(1000);

mySerial.println((String)"update-

>"+(String)"Temprature="+t+(String)"Humidity="+h+(String)"Moisture="+m);

mySerial.println();

delay(100);

Serial.write(0x1A);

delay(1000);

c++;

}

}

