

SPRINT - 1

Team ID	PNT2022TMID00072
Project Name	Smart Farmer-IoT Enabled Smart Farming Application

PYTHON CODE :

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
```

#Provide your IBM Watson Device Credentials

```
organization = "p2cfk6"
deviceType = "SMART"
deviceId = "15"
authMethod = "token"
authToken = "12345678"
```

Initialize GPIO

```
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()

# Connect and send a datapoint
deviceCli.connect()

while True:
    temp=random.randint(0,100) # Temperature value
    Humid=random.randint(0,100) # Humidity value
    moisture = random.randint(0,100) # Soil moisture value

    data = { 'temp' : temp, 'Humid': Humid, 'Moisture' : moisture }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %
Humid, "Soil Moisture = %s %%" % moisture, "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

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

Connecting Sensors with Arduino using C code :

```
#include "Arduino.h"

#include "dht.h"
#include "SoilMoisture.h"
#include "Pump.h"

#define DHT_PIN 2
#define SOILMOISTURE_PIN A3
#define WATERPUMP_PIN 5

dht DHT;
int c=0;

void setup()
{
    Serial.begin(9600);
    pinMode(5, OUTPUT); // Output for Pump

    delay(1000);
}

void loop()
{
    DHT.read11(DHT_PIN);
    float h=DHT.humidity;
    float t=DHT.temperature;
    delay(1000);

    float moisture_percent;
    int moisture_analog;
    moisture_analog = analogRead(SOILMOISTURE_PIN);

    moisture_percent = ( 100 - ( (moisture_analog/1023.00) *100 ) );
```

```

float moist= moisture_percent;

delay(1000);

if(moist<40)// Pump functions
{
    while(moist<40)
    {
        digitalWrite(5 ,HIGH); // Pump ON

        moisture_analog = analogRead(SOILMOISTURE_PIN);

        moisture_percent = ( 100 - ( (moisture_analog/1023.00) *100 ) );

        moist=moisture_percent;

        delay(1000);
    }

    digitalWrite(5 ,LOW);    // Pump OFF
}

if(c>=0)
{
    Serial.print("\r");

    delay(1000);

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

    delay(1000);

    c++;
}
}

```

Circuit Diagram :

