Project Delivery Sprint - 1

Date	20 Oct 2022
Team ID	PNT2022TMID04699
Project Name	Smart Farmer-IOT Enabled Smart Farming Application

Arduino using Python code To Connect Sensors

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "i3869j"
deviceType = "abcd"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"


# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led 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 "hello" with value "world" into the cloud as an 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)
        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
           print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "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()
```

Arduino using C++ code To Connect Sensors

```
#include "Arduino.h" #include
"dht.h"
#include "SoilMoisture.h"
#define dht_apin A0
const int sensor_pin = A1; //soil moisture int pin_out = 9;
dht DHT; int c=0; void setup()
{
pinMode(2, INPUT); //Pin 2 as INPUT pinMode(3, OUTPUT); //PIN
3 as OUTPUT pinMode(9, OUTPUT);//output for pump
} void
loop()
{
if (digitalRead(2) == HIGH)
{
digitalWrite(3, HIGH); // turn the LED/Buzz ON
delay(10000); // wait for 100 msecond digitalWrite(3, LOW); // turn the
LED/Buzz OFF delay(100);
}
Serial.begin(9600);
delay(1000);
DHT.read11(dht_apin); //temprature 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)//pump
{ while(m<40)
{
digitalWrite(pin_out,HIGH); //open pump sensor_analog =
analogRead(sensor_pin);
moisture_percentage = ( 100 - ( (sensor_analog/1023.00) * 100 )
); m=moisture_percentage; delay(1000);
}
digitalWrite(pin_out,LOW); //closepump
} if(c>=0)
{
mySerial.begin(9600); delay(15000);
Serial.begin(9600); delay(1000);
Serial.print("\r"); delay(1000);
Serial.print((String)"update-
>"+(String)"Temprature="+t+(String)"Humidity="+h+(String
)"Moisture="+m); delay(1000);
}
}
```

Circuit Diagram: