Assignment-6:

#include <WiFi.h>

#include <PubSubClient.h> // to connect to MQTT client in Iot platform

int trigpin=5;

int echopin=4;

float distance;

int ldr;

const char\* ssid = "wifi name";//your wifi ssid

const char\* password = "password";//your password

// CHANGE TO YOUR DEVICE CREDENTIALS AS PER IN IBM BLUMIX

#define ORG "p0ql3m"

#define DEVICE\_TYPE "Esp32"

#define DEVICE\_ID "12345"

#define TOKEN "12345678" // Authentication Token OF THE DEVICE

// PIN DECLARATIONS

//-------- Customise the above values --------

const char publishTopic[] = "iot-2/evt/Data/fmt/json";

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

//char topic[] = "iot-2/cmd/home/fmt/String";// cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883, wifiClient);

int publishInterval = 5000; // 30 seconds

long lastPublishMillis;

void publishData();

void setup() {

Serial.begin(115200);

pinMode(trigpin,OUTPUT);

pinMode(echopin,INPUT);

Serial.println();

Serial.print("Started..........");

wifiConnect();

mqttConnect();

}

void loop() {

if (millis() - lastPublishMillis > publishInterval)

{

publishData();

lastPublishMillis = millis();

}

if (!client.loop()) {

mqttConnect();

}

}

void wifiConnect() {

Serial.print("Connecting to "); Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.print("WiFi connected, IP address: ");

Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected())

{

Serial.print("Reconnecting MQTT client to "); Serial.println(server);

while (!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

}

}

void publishData()

{

digitalWrite(trigpin,HIGH);

delay(1000);

digitalWrite(trigpin,LOW);

int duration=pulseIn(echopin,HIGH);

distance=duration\*0.343/2; //velocity of air=0.343

Serial.println("the distance is");

Serial.print(distance);

Serial.print("cm");

delay(1000);

int LDRValue=analogRead(32);

double dV=LDRValue;

Serial.println(LDRValue);

Serial.println("ldr value is");

Serial.print(ldr);

delay(1000);

String payload = "{\"d\":{\"distance\":";

payload += distance;

payload += ",""\"ldr\":";

payload += ldr;

payload += "}}";

Serial.print("\n");

Serial.print("Sending payload: "); Serial.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) {

Serial.println("Publish OK");

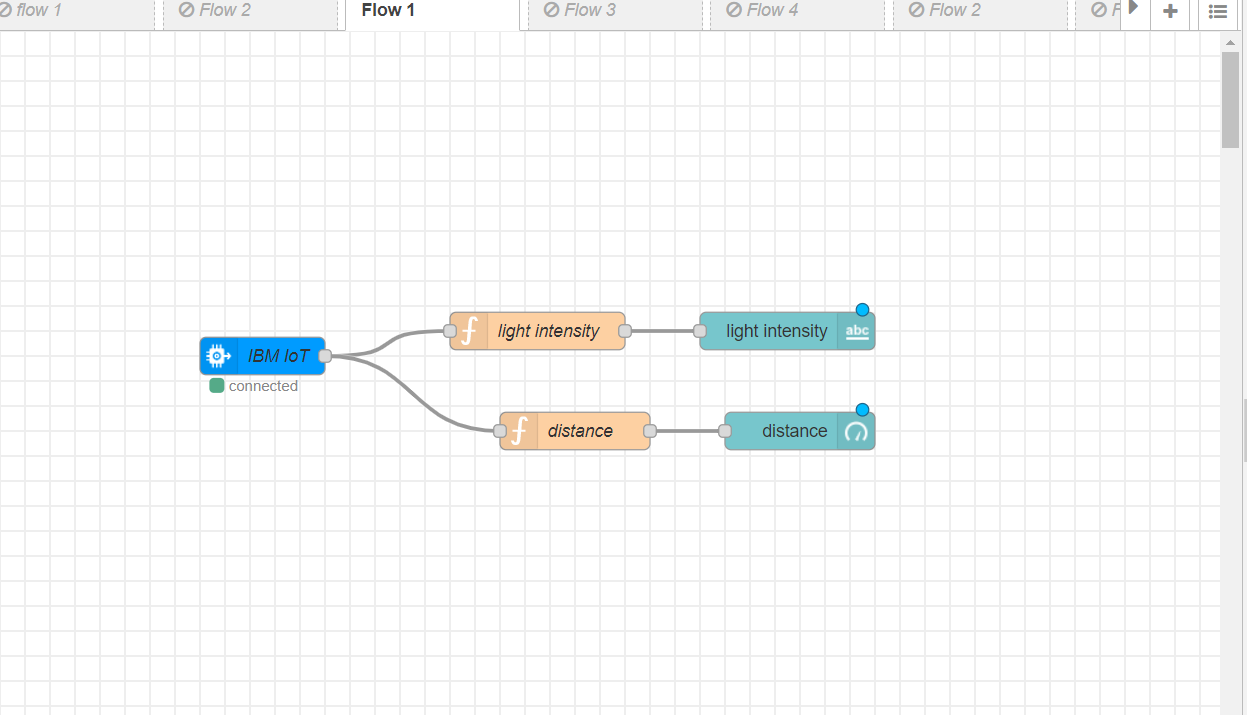
} else {

Serial.println("Publish FAILED");

}

}

Node-Red Alignment of nodes:



OUTPUT:

