**Assignment-4**

|  |  |
| --- | --- |
| **Student Name** | TAMIL VANAN S |
| **Roll Number** | 211719106087 |
| **Team ID** | PNT2022TMID26547 |
| **Date** | 30-October 2022 |
| **Project Name** | **Project -Smart farmer-IOT enabled smart Farming Application** |

**Question:**

Write code and connections in wokwi for ultrasonic sensor.

Whenever distance is less than 100cm send "alert" to IBM cloud and display in device recent events.

Upload document with wokwi share link and images of IBM cloud.

**CODE:**

#include <WiFi.h>

#include <WiFiClient.h>

#include <PubSubClient.h>

#define ORG "17lsro"

#define DEVICE\_TYPE "MyDeviceType"

#define DEVICE\_ID "12345"

#define TOKEN "GkatKdiUS?UVHKvnAD"

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

char pubTopic1[] = "iot-2/evt/TAMIL VANAN S /fmt/json";

char pubTopic2[] = "iot-2/evt/status2/fmt/json";

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

char token[] = TOKEN;

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

const int DHT\_PIN = 15;

WiFiClient wifiClient;

PubSubClient client(server, 1883, NULL, wifiClient);

#define ECHO\_PIN 12

#define TRIG\_PIN 13

float readDistanceCM() ;

void setup() {

**Serial**.begin(115200);

  pinMode(15, OUTPUT);

  pinMode(TRIG\_PIN, OUTPUT);

  pinMode(ECHO\_PIN, INPUT);

**Serial**.println();

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

    WiFi.begin("Wokwi-GUEST", "", 6);

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

      delay(50);

**Serial**.print(".");

    }

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

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

**Serial**.println(WiFi.localIP());

    if (!client.connected()) {

**Serial**.print("Reconnecting client to ");

**Serial**.println(server);

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

**Serial**.print(".");

            delay(500);

        }

**Serial**.println("Bluemix connected");

}

}

float readDistanceCM()

{

  digitalWrite(TRIG\_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG\_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG\_PIN, LOW);

  int duration = pulseIn(ECHO\_PIN, HIGH);

  return duration \* 0.034 / 2;

}

long lastMsg = 0;

void loop() {

  float distance = readDistanceCM();

  bool isNearby = distance < 100;//checking  whether the distance is less than 100

  digitalWrite(15, isNearby);

**Serial**.print("Measured distance: ");

**Serial**.println(readDistanceCM());

  delay(100);

  if(isNearby)//Whenever the distance is less than 100 cms send an "alert" to the IBM cloud

  {

   client.loop();

    long now = millis();

    if (now - lastMsg > 3000) {

        lastMsg = now;

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

       payload += distance;

       payload += "}";

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

**Serial**.println(payload);

        if (client.publish(pubTopic1, (char\*) payload.c\_str())) {

**Serial**.println("Publish ok");

        } else {

**Serial**.println("Publish failed");

        }

    }

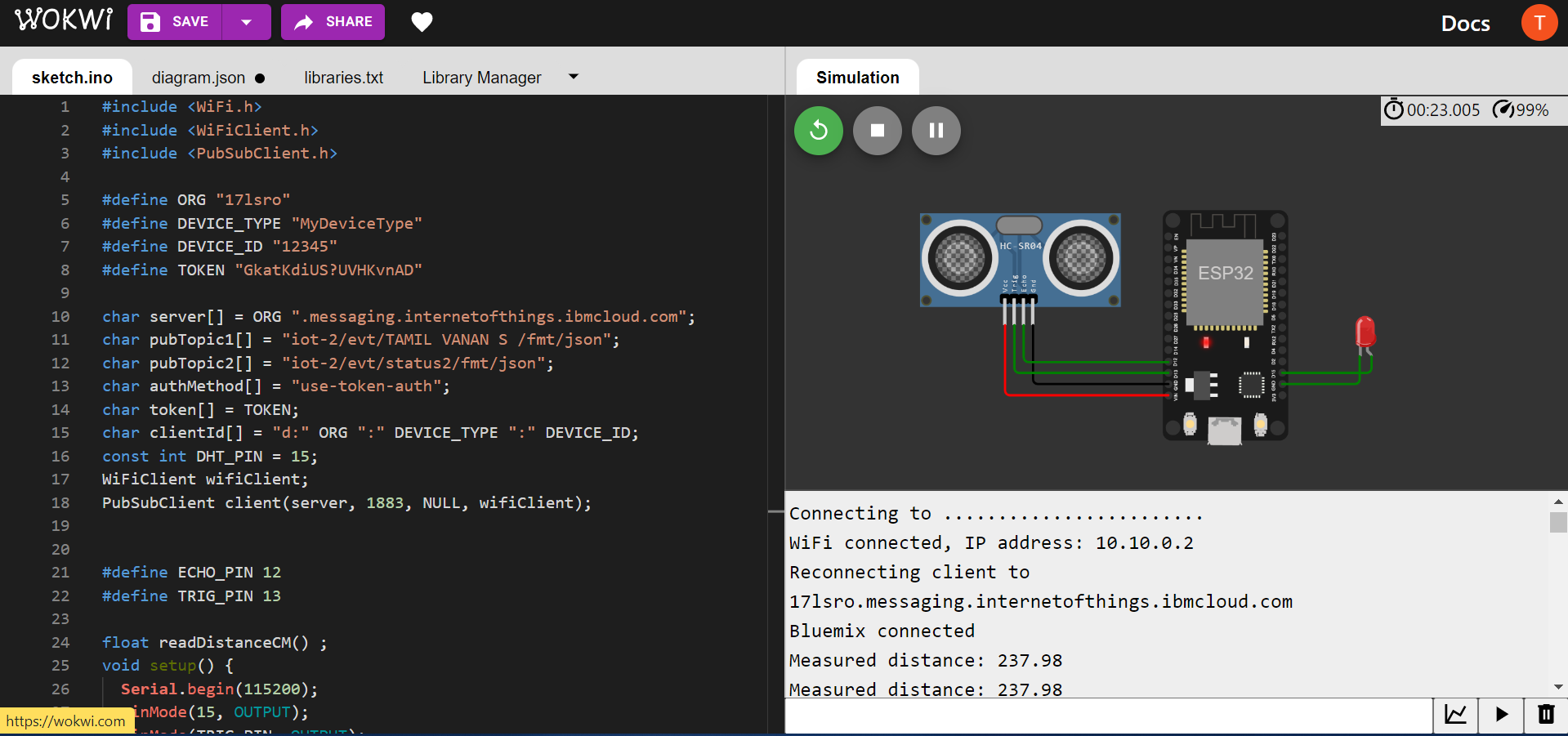
}

}

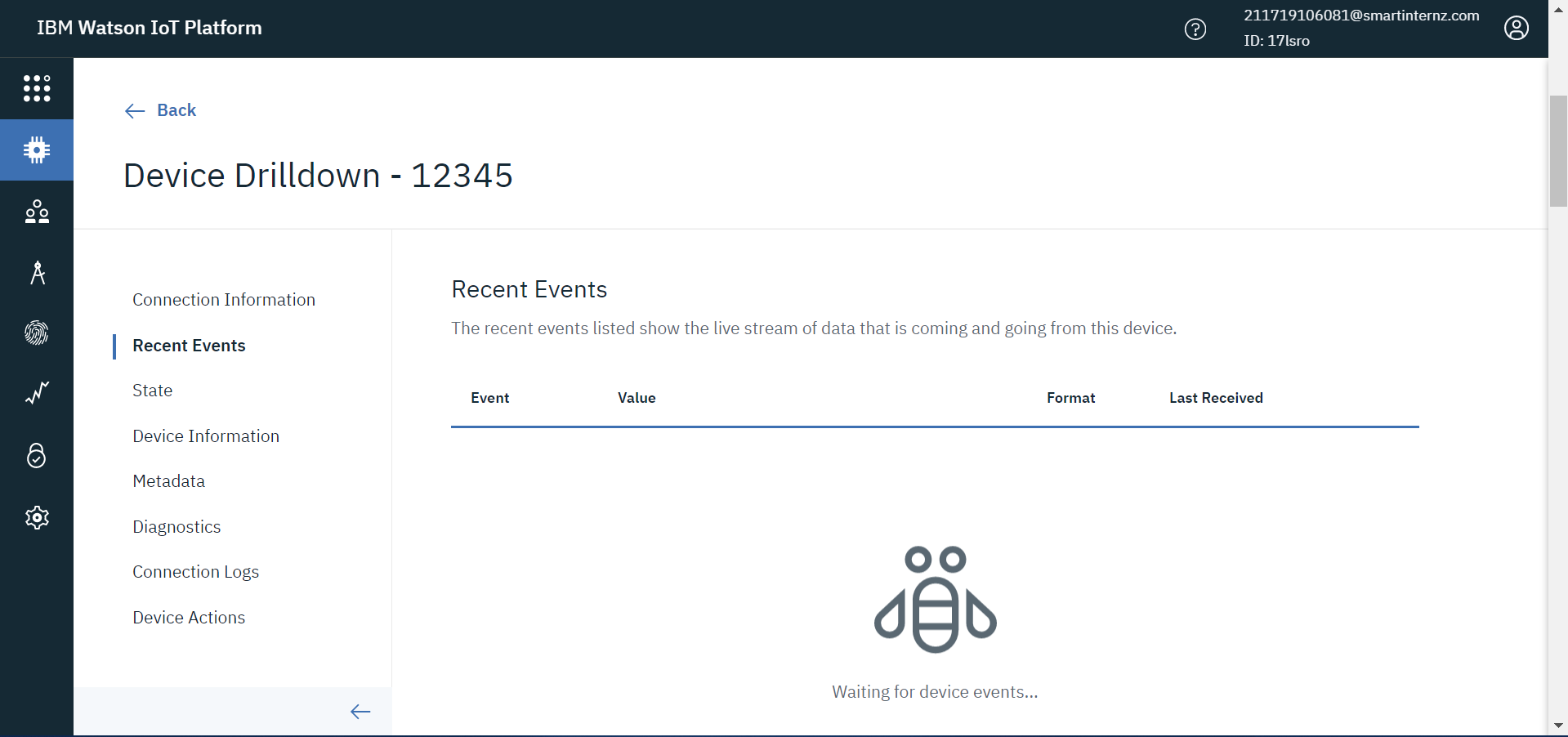
**OUTPUT:**

**Case: 1**

When Distance Is Above 100 Cm

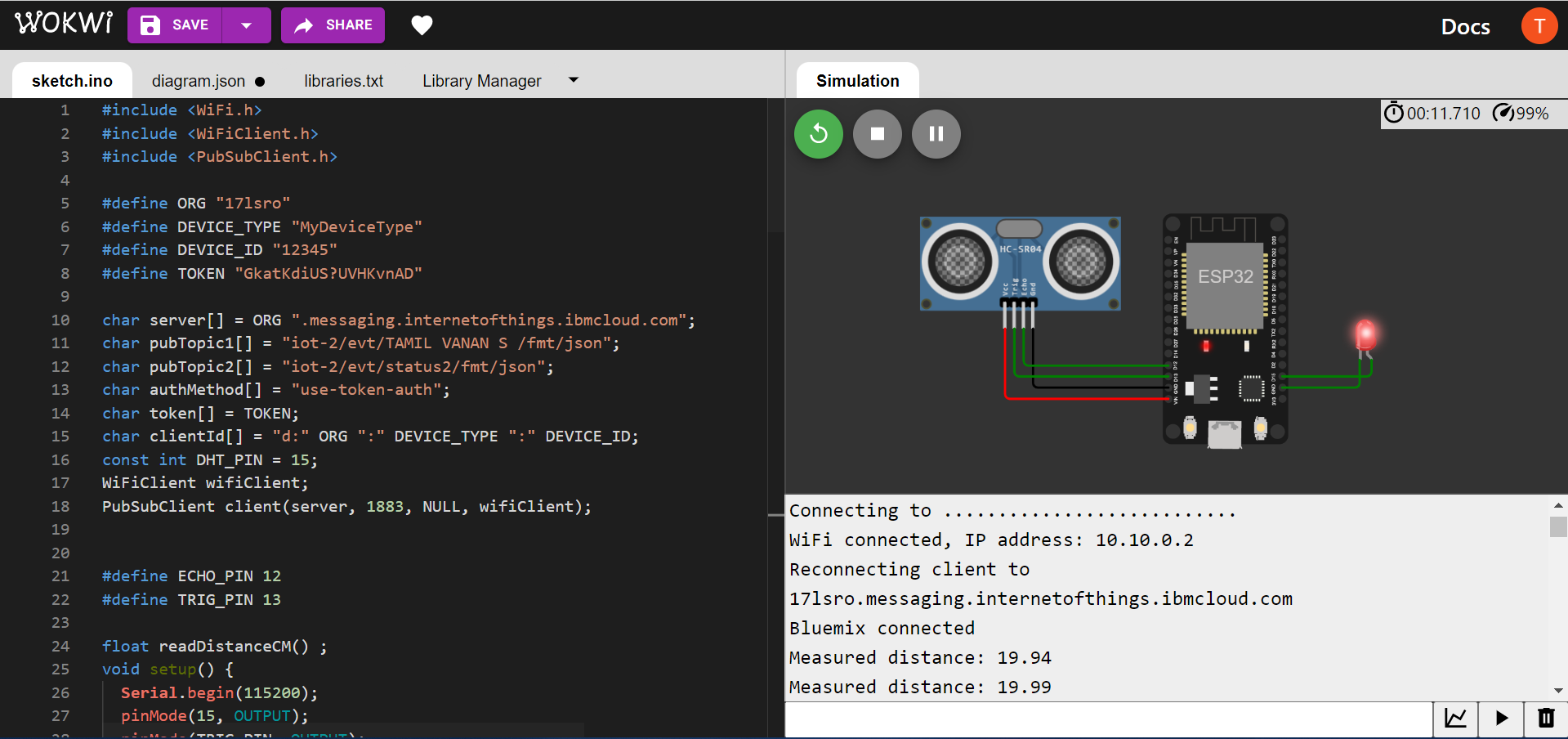


Data Is Not Send to IBM IOT PLATFORM If Distance Is Above 100 Cm

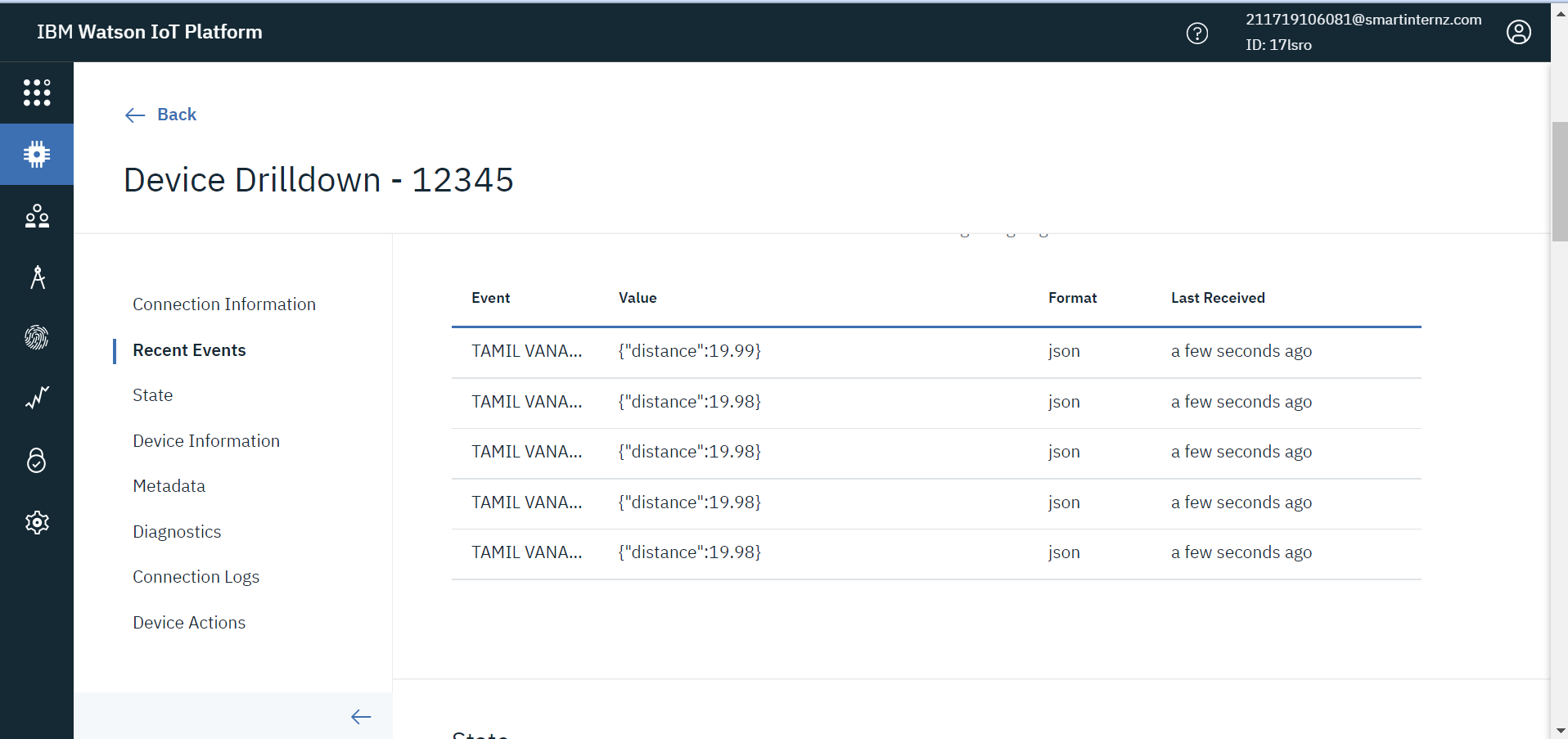


**Case:2**

When Distance Is Below 100 Cm

****

When The Distance Is Below 100Cm Data Is Sent To IBM Iot Platform

****