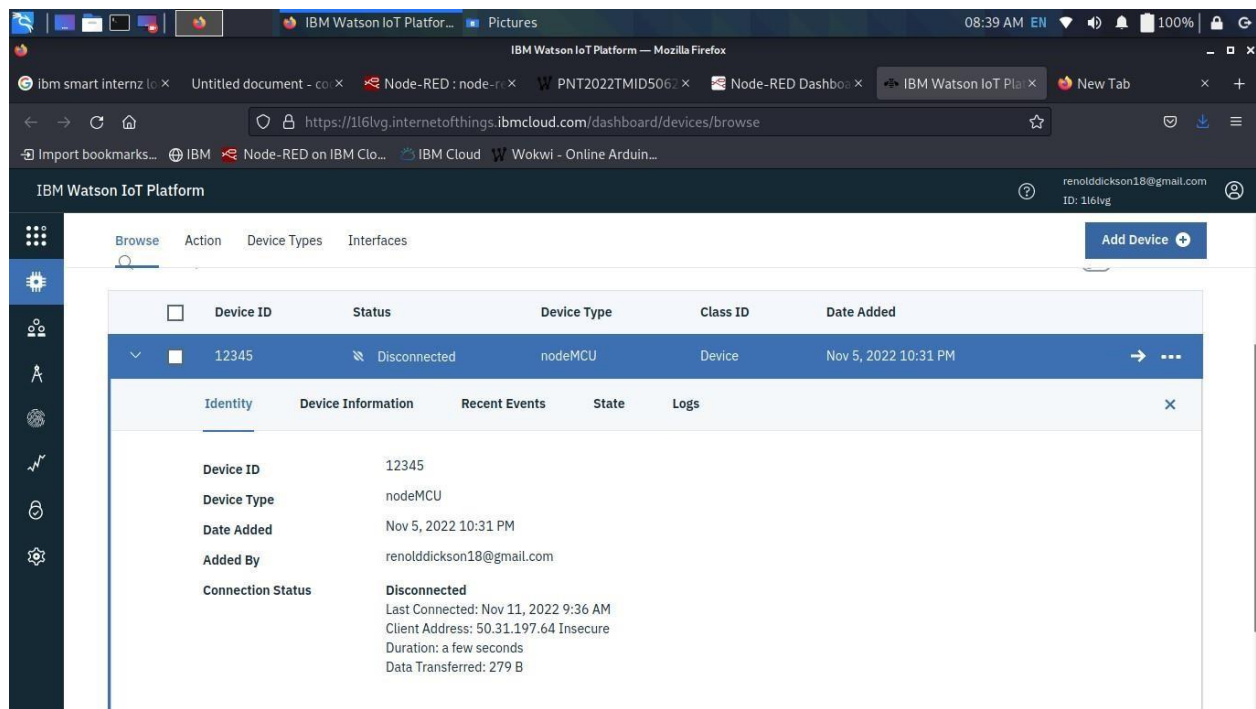


SPRINT - 1

| | |
|--------------|---|
| Team ID | PNT2022TMID50941 |
| Project Name | Personal Assistance for Seniors Who are self-reliant |
| Date | 15November2022 |

Created IOT device config in IBM IOT watson



Connected IBM IOT watson cloud with wokwi(nodemcu)

Wokwi url: <https://wokwi.com/projects/347406552336958036>

Code:

```
#include<WiFi.h>//library for wifi module #include<PubSubClient.h>//library for
MQTT void callback(char* subscribetopic, byte* payload,unsigned int
payloadlength);
//-----credentials of IBM Account-----
#define ORG "twoqgm"// IBM ORGANIZATION ID
#define DEVICE_TYPE "nodemcu"//DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
#define DEVICE_ID "11111"//DEVICE ID MENTIONED IN IOT WATSON PLATEFORM
```

```

#define TOKEN "XsgsvH8+*w4p2Y_XQO"//Token
String data3;
float dist;
//-----customize the above value-----char server[]=ORG
".messaging.internetofthings.ibmcloud.com";//server name char
publishtopic[]="ultrasonic/evt/Data/fmt/json";//topic name and type of event perform and format
in which data to be send*/
char subscribetopic[]="ultrasonic/cmd/test/fmt/String";//cmd REPRESENT Command tupe and
COMMAND IS TEST OF FORMAT STRING*/char authMethod[]="use-token-
auth";//authentication method char token[]=TOKEN; char clientid[]="d:" ORG ":"
DEVICE_TYPE":" DEVICE_ID;//CLIENT ID
//-----
WiFiClient wifiClient;// creating an instance for wificlient
PubSubClient client(server, 1883 , wifiClient); void setup()
{
Serial.begin(115200);
wificonnect(); mqttconnect();
}
void loop()//recursive function
{
if (!client.loop()){
mqttconnect();
}
}
/* .....retriving to cloud.....*/ void
mqttconnect(){ if(!client.connected()){
Serial.print("Reconnecting client to ");
Serial.println(server);
while(!client.connect(clientid,authMethod,
token)){ Serial.print("."); delay(500); }
initManagedDevice();
Serial.println();
} else{
Serial.println("Connected :)");
} } void wificonnect()//function defenition for wificonnects
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "",6);//PASSING THE WIFI CREDIDENTIALS TO ESTABLISH
CONNECTION
while (WiFi.status() !=WL_CONNECTED){delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

```

The screenshot displays the Wokwi IoT platform interface. The top navigation bar includes the Wokwi logo and various links. The main workspace is divided into two sections. The left section, titled 'sketch.ino', contains C++ code for an IoT device. The code includes headers for WiFi and MQTT, defines constants for the IBM Watson IoT Platform (ORG, DEVICE_TYPE, DEVICE_ID, TOKEN), and implements a setup function that initializes the serial port, connects to the WiFi, and subscribes to a specific MQTT topic. The right section, titled 'Simulation', shows a visual representation of the ESP32 module and a console output. The console output indicates that the device is connecting to the IBM Watson IoT Platform, successfully connects to the WiFi, and receives a message from the subscribed topic.

Url: <https://pnt2022tmid50622.atlassian.net/jira/software/projects/PAFSCII/boards/1>

[illegible]