Final code

Team I’d -PNT2022TMID13631

#include <WiFi.h> #include <WiFiClient.h> #include <PubSubClient.h> #include <ArduinoJson.h> #include<TinyGPS++.h> #define RXD2 16 #define TXD2 17#include <WiFi.h> #include <WiFiClient.h> #include <PubSubClient.h> #include <ArduinoJson.h> #include<TinyGPS++.h> #define RXD2 16 #define TXD2 17 HardwareSerial neogps(1); TinyGPSPlus gps; char arr[100]; const char\* ssid = "Redmi"; const char\* password = ""; #defineMadhu@16 ID "17cmwk" #define DEVICE\_TYPE "Tracker" #define DEVICE\_ID "gps1" #define TOKEN "childtracker1" char server[] = ID ".messaging.internetofthings.ibmcloud.com"; char publish\_Topic1[] = "iot-2/evt/Data1/fmt/json"; char publish\_Topic2[] = "iot-2/evt/Data2/fmt/json"; char authMethod[] = "use-token-auth"; char token[] = TOKEN; char clientId[] = "d:" ID ":" DEVICE\_TYPE ":" DEVICE\_ID; WiFiClient wifiClient; PubSubClient client(server, 1883, NULL, wifiClient); void setup() { Serial.begin(115200); Serial.println(); wifi\_init(); } long previous\_message = 0; void loop() { client.loop(); String payload = getLocationPayload(); if(payload=="{}"){ return; } Serial.print("Sending payload: "); Serial.println(payload); if (client.publish(publish\_Topic1, arr)) { Serial.println("Published successfully"); } else { Serial.println("Failed"); } delay(2000); } void wifi\_init(){ WiFi.begin(ssid, password); neogps.begin(9600,SERIAL\_8N1,RXD2,TXD2); while (WiFi.status() != WL\_CONNECTED) { delay(500); Serial.print("."); } Serial.println(""); 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("Connected TO IBM IoT cloud!"); } } String getLocationPayload(){ boolean newData = false; for(unsigned long start = millis();millis()-start<1000;){ while(neogps.available()){ if(gps.encode(neogps.read())){ newData = true; } } } String payload; if(newData == true){ newData = false; payload = locationPayloadGenerator();

} else{ Serial.println("No data"); payload ="{}"; } return payload; } String locationPayloadGenerator(){ String payload = "{}"; if(gps.location.isValid()){ float lat = gps.location.lat(); float lon = gps.location.lng(); payload = "{\"latitude\" : "+String(lat)+",\"longitude\" : "+String(lon)+"}"; create\_json(lat,lon); } return payload; } void create\_json(float lat,float lon){ StaticJsonDocument<100> doc; JsonObject root = doc.to<JsonObject>(); root["name"]="Child"; root["latitude"] = lat; root["longitude"] = lon; serializeJsonPretty(doc,arr); } Footer HardwareSerial neogps(1); TinyGPSPlus gps; char arr[100]; const char\* ssid = "Redmi"; const char\* password = "krish@08"; #define ID "17cmwk" #define DEVICE\_TYPE "Tracker" #define DEVICE\_ID "gps1" #define TOKEN "childtracker1" char server[] = ID ".messaging.internetofthings.ibmcloud.com"; char publish\_Topic1[] = "iot-2/evt/Data1/fmt/json"; char publish\_Topic2[] = "iot-2/evt/Data2/fmt/json"; char authMethod[] = "use-token-auth"; char token[] = TOKEN; char clientId[] = "d:" ID ":" DEVICE\_TYPE ":" DEVICE\_ID; WiFiClient wifiClient; PubSubClient client(server, 1883, NULL, wifiClient); void setup() { Serial.begin(115200); Serial.println(); wifi\_init(); } long previous\_message = 0; void loop() { client.loop(); String payload = getLocationPayload(); if(payload=="{}"){ return; } Serial.print("Sending payload: "); Serial.println(payload); if (client.publish(publish\_Topic1, arr)) { Serial.println("Published successfully"); } else { Serial.println("Failed"); } delay(2000); } void wifi\_init(){ WiFi.begin(ssid, password); neogps.begin(9600,SERIAL\_8N1,RXD2,TXD2); while (WiFi.status() != WL\_CONNECTED) { delay(500); Serial.print("."); } Serial.println("");

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("Connected TO IBM IoT cloud!"); } } String getLocationPayload(){ boolean newData = false; for(unsigned long start = millis();millis()-start<1000;){ while(neogps.available()){ if(gps.encode(neogps.read())){ newData = true; } } } String payload; if(newData == true){ newData = false; payload = locationPayloadGenerator();

} else{ Serial.println("No data"); payload ="{}"; } return payload; } String locationPayloadGenerator(){ String payload = "{}"; if(gps.location.isValid()){ float lat = gps.location.lat(); float lon = gps.location.lng(); payload = "{\"latitude\" : "+String(lat)+",\"longitude\" : "+String(lon)+"}"; create\_json(lat,lon); } return payload; } void create\_json(float lat,float lon){ StaticJsonDocument<100> doc; JsonObject root = doc.to<JsonObject>(); root["name"]="Child"; root["latitude"] = lat; root["longitude"] = lon; serializeJsonPretty(doc,arr); } Footer Foote