IOT BASED SAFETY GADGET FOR CHILD SAFETY MONITORING AND NOTIFICATION

Team ID : PNT2022TMID23803

Team Members : 4

Team Leader : AKSHAYA J

Team member 1: DEEPIKA G

Team member 2 : DHARANI K

Team member 3 : EZHILARASI S

FINAL CODE :

```
#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 = "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);
}
```