

```

#include <Adafruit_NeoPixel.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <WiFiClientSecure.h>

#define broche 33
#define nbpixels 78
#define delai 30000
#define taille_buffer 16384
#define nd_arret 28

// ----- LED -----
Adafruit_NeoPixel pixels(nbpixels, broche, NEO_GRB + NEO_KHZ800);

// ----- WIFI -----
const char* ssid = "Point";
const char* password = "Point...";

struct arret { // structure arrêt
  const char* nom;
  float lat_max;
  float lat_min;
  float long_max;
  float long_min;
  int led_nb;
};

arret Liste_arret[nd_arret] = {}; // liste d'arret

int tram_count;
unsigned char buffer[taille_buffer]; // buffer pour stocker les données de l'api
int buffer_length = 0;

void downloadData() // fonction pour télécharger les données
{
  buffer_length = 0;

  // Pas de vérification HTTPS : moins sécurisé, mais ça évite de charger un
  // certificat en mémoire
  WiFiClientSecure client;
  client.setInsecure();

  // Connexion au serveur BIBUS
  Serial.println("Connexion au serveur BIBUS...");
  HTTPClient http;

```

```

if (http.begin(client, "https://proxy.transport.data.gouv.fr/resource/bibus-brest-
gtfs-rt-vehicle-position")) {
    int httpCode = http.GET();

    if (httpCode == HTTP_CODE_OK) {
        int len = http.getSize(); // Récupère la taille du fichier GTFS-RT

        if ((len > 0)&&(len < 16384)) {

            WiFiClient* stream = http.getStreamPtr();
            if (stream) {

                // Téléchargement par blocs pour remplir le buffer
                while (http.connected() && (buffer_length < len)) {
                    size_t size = stream->available();
                    if (size) {
                        int c = stream->readBytes(&buffer[buffer_length%taille_buffer], size);
                        buffer_length += c;

                        Serial.print(".");
                    }
                }

                Serial.println();
                Serial.printf("Téléchargement terminé : %d octets\n", buffer_length);

            } else {
                Serial.println("Erreur pour récupérer le stream HTTPs");
            }
        } else {
            Serial.println("Erreur : buffer trop petit !");
        }
    } else {
        Serial.printf("Erreur HTTP : %s\n", http.errorToString(httpCode).c_str());
    }

    http.end(); // Fin de la connexion HTTPs
}

// Fonction pour convertir 4 octets (Little Endian) en float
float bytesToFloat(const uint8_t* p) {
    float f;
    memcpy(&f, p, 4);
    return f;
}

void analyserTrameGTFS() { //fonction pour analyser les données téléchargées
    printf("--- DEBUT DE L'ANALYSE ---\n");
    tram_count = 0;
}

```

```

//eteindre les leds
for (int k = 0; k < nd_arret; k++) {
pixels.setPixelColor(Liste_arret[k].led_nb, pixels.Color(0, 0, 0));
pixels.show();}

// On parcourt le buffer à la recherche des entités (Tag 0x12)
for (int i = 0; i < buffer_length; i++) {

    // Detection d'une entité : Tag 0x12 (Entity)
    if (buffer[i] == 0x12) {
        int entity_len = buffer[i+1];
        int entity_start = i + 2;

        // Recherche du Vehicule ID (Tag 0x0A dans l'entité)
        char current_v_id[32] = "Inconnu";
        for (int j = entity_start; j < entity_start + 20; j++) {
            if (buffer[j] == 0x76 && buffer[j+1] == 0x65) { // "ve..."
                int id_len = buffer[j-1];
                snprintf(current_v_id, (id_len < 31 ? id_len + 1 : 31),
(char*)&buffer[j]);
                break;
            }
        }

        bool is_tram = false;
        char type_vehicule[10] = "Bus";
        for (int j = entity_start; j < entity_start + entity_len; j++) {
            if (buffer[j] == 0x2A) { // Tag route_id
                if (buffer[j+1] == 0x01 && buffer[j+2] == 0x41) { // 0x41 = 'A'
                    strcpy(type_vehicule, "TRAM A");
                    is_tram = true;
                }
                break;
            }
        }

        // Extraction des coordonnées GPS
        float lat = 0, lon = 0;
        for (int j = entity_start; j < entity_start + entity_len; j++) {
            if (buffer[j] == 0x0D) { // Latitude
                lat = bytesToFloat(&buffer[j+1]);
            }
            if (buffer[j] == 0x15) { // Longitude
                lon = bytesToFloat(&buffer[j+1]);
            }
        }

        // Affichage si on a trouvé des coordonnées valides
        if (is_tram && lat != 0) {
            tram_count++;
        }
    }
}

```

```

    for (int j = 0; j < nd_arret; j++) {
        // Vérifier si le tram est sur l'arrêt
        if ((lat < Liste_arret[j].lat_max && lat > Liste_arret[j].lat_min) &&
            (lon < Liste_arret[j].long_max && lon > Liste_arret[j].long_min)) {
            printf("[%s] ID: %s | arret: %s \n",
                type_vehicule, current_v_id, Liste_arret[j].nom);

            // Allumer la LED correspondant à l'arrêt
            pixels.setPixelColor(Liste_arret[j].led_nb, pixels.Color(0, 150, 0));
            pixels.show();
        }
    }
    i += entity_len;
}

printf("--- FIN DE L'ANALYSE ---\n");
}

```

```

void setup() {

```

```

    Serial.begin(115200);

```

```

    pixels.begin();
    pixels.setBrightness(50); // initialisation des leds
    pixels.clear();
    pixels.show();

```

```

    WiFi.begin(ssid, password); // initialisation de la connection wifi
    Serial.print("Connexion au WiFi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.print("\nConnected to Wi-Fi network with IP Address: ");
    Serial.println(WiFi.localIP());

```

```

Liste_arret[0] = { // liste de tous les arrêts du tram, ainsi que leur coordonnées
    "porte plouzane",
    48.48375994,
    48.377961,
    -4.547852,

```

```
-4.554199,  
0  
};
```

```
Liste_arret[1] = {  
    "fort montbarey",  
    48.381144,  
    48.376875,  
    -4.541294,  
    -4.548565,  
    1  
};
```

```
Liste_arret[2] = {  
    "kérannroux",  
    48.384528,  
    48.380512,  
    -4.536135,  
    -4.532103,  
    2  
};
```

```
Liste_arret[3] = {  
    "coat tan",  
    48.386820,  
    48.384133,  
    -4.529541,  
    -4.536938,  
    3  
};
```

```
Liste_arret[4] = {  
    "val hir",  
    48.387873,  
    48.386271,  
    -4.521774,  
    -4.529920,  
    4  
};
```

```
Liste_arret[5] = {  
    "polygone",  
    48.389576,  
    48.386816,  
    -4.513365,  
    -4.521971,  
    5  
};
```

```
Liste_arret[6] = {  
    "dupuy de lome",
```

```
48.391591,  
48.389291,  
-4.506991,  
-4.513809,  
6  
};  
  
Liste_arret[7] = {  
    "les capucins",  
48.392151,  
48.389433,  
-4.503496,  
-4.506872,  
7  
};  
  
Liste_arret[8] = {  
    "saint exupéry",  
48.389901,  
48.386394,  
-4.503791,  
-4.504754,  
8  
};  
  
Liste_arret[9] = {  
    "mac orlan",  
48.386212,  
48.383869,  
-4.500667,  
-4.504032,  
9  
};  
  
Liste_arret[10] = {  
    "recouvrance",  
48.384992,  
48.383621,  
-4.495307,  
-4.501744,  
10  
};  
  
Liste_arret[11] = {  
    "chateau",  
48.387521,  
48.384250,  
-4.489854,  
-4.494623,  
11  
};
```

```
Liste_arret[12] = {  
    "siam",  
    48.389871,  
    48.386909,  
    -4.487087,  
    -4.490748,  
    12  
};
```

```
Liste_arret[13] = {  
    "liberté",  
    48.391644,  
    48.389503,  
    -4.483785,  
    -4.487211,  
    13  
};
```

```
Liste_arret[14] = {  
    "jean jaures",  
    48.393494,  
    48.391413,  
    -4.480993,  
    -4.484316,  
    14  
};
```

```
Liste_arret[15] = {  
    "saint martin",  
    48.395846,  
    48.393355,  
    -4.477499,  
    -4.481214,  
    15  
};
```

```
Liste_arret[16] = {  
    "octroi",  
    48.397918,  
    48.395494,  
    -4.474355,  
    -4.477971,  
    16  
};
```

```
Liste_arret[17] = {  
    "pillier rouge",  
    48.400799,  
    48.397694,  
    -4.469849,
```

```
-4.474585,  
17  
};
```

```
Liste_arret[18] = {  
    "place de strasbourg",  
    48.405118,  
    48.400639,  
    -4.467327,  
    -4.470112,  
    18  
};
```

```
Liste_arret[19] = {  
    "menez paul",  
    48.410148,  
    48.404597,  
    -4.467368,  
    -4.468463,  
    19  
};
```

```
Liste_arret[20] = {  
    "europe",  
    48.413350,  
    48.409120,  
    -4.469341,  
    -4.4471942,  
    20  
};
```

```
Liste_arret[21] = {  
    "pontanézen",  
    48.418446,  
    48.413871,  
    -4.465310,  
    -4.472115,  
    21  
};
```

```
Liste_arret[22] = {  
    "eau blanche",  
    48.418402,  
    48.416256,  
    -4.455644,  
    -4.463926,  
    22  
};
```

```
Liste_arret[23] = {  
    "kerlaurent",
```

```
48.419780,  
48.417615,  
-4.448016,  
-4.456213,  
23  
};
```

```
Liste_arret[24] = {  
    "porte de guipavas",  
    48.419887,  
    48.419033,  
    -4.443616,  
    -4.448905,  
    24  
};
```

```
Liste_arret[25] = {  
    "mesmerrien",  
    48422180.,  
    48.418049,  
    -4.466353,  
    -4.468526,  
    25  
};
```

```
Liste_arret[26] = {  
    "kergaradec",  
    48.427632,  
    48.422944,  
    -4.466457,  
    -4.467400,  
    26  
};
```

```
Liste_arret[27] = {  
    "porte de guesnou",  
    48.431615,  
    48.427470,  
    -4.467107,  
    -4.468236,  
    27  
};  
}
```

```
void loop() { //boucle principal  
    downloadData();  
    analyserTrameGTFS();  
    printf("nb de trame détecter : %u \n", tram_count);  
    delay(delai);  
}
```

