

```

import requests
from google.transit import gtfs_realtime_pb2
from google.protobuf.json_format import MessageToDict
import time
import board
import neopixel
from math import sqrt

# --- CONFIGURATION MATÃRIELLE ---
OFFSET = 10
# Ruban 1 : Ligne principale (Pin physique 12 / GPIO 18)
pixels_main = neopixel.NeoPixel(board.D18, 60, brightness=0.1,
auto_write=True)
# Ruban 2 : Branche Gouesnou (Pin physique 40 / GPIO 21)
pixels_gouesnou = neopixel.NeoPixel(board.D21, 3, brightness=0.1,
auto_write=True)

URL_BASE = "https://proxy.transport.data.gouv.fr/resource/bibus-brest-
gtfs-rt-vehicle-position"

# --- STATIONS DATA (CoordonnÃ©es corrigÃ©es en dÃ©cimal pur) ---
STATIONS_DATA = {
"Porte de PlouzanÃ©": {"led": 0, "lat": 48.3610, "lon": -4.5645, "term":
True, "branch": "MAIN"}, "Fort Montbarey": {"led": 1, "lat": 48.3698,
"lon": -4.5492, "term": False, "branch": "MAIN"}, "KÃ©ranroux": {"led": 2,
"lat": 48.3734, "lon": -4.5422, "term": False, "branch": "MAIN"}, "Coat
Tan": {"led": 3, "lat": 48.3854, "lon": -4.5332, "term": False, "branch":
"MAIN"}, "Val Hir": {"led": 4, "lat": 48.3870, "lon": -4.5258, "term":
False, "branch": "MAIN"}, "Polygone": {"led": 5, "lat": 48.3881, "lon": -
4.5176, "term": False, "branch": "MAIN"}, "Dupuy de Lome": {"led": 6,
"lat": 48.3904, "lon": -4.5104, "term": False, "branch": "MAIN"}, "Les
Capucins": {"led": 7, "lat": 48.3907, "lon": -4.5051, "term": False,
"branch": "MAIN"}, "Saint ExupÃ©ry": {"led": 8, "lat": 48.3881, "lon": -
4.5042, "term": False, "branch": "MAIN"}, "Mac Orlan": {"led": 9, "lat":
48.3850, "lon": -4.5023, "term": False, "branch": "MAIN"}, "Recouvrance":
{"led": 10, "lat": 48.3843, "lon": -4.4985, "term": False, "branch":
"MAIN"}, "Chateau": {"led": 11, "lat": 48.3858, "lon": -4.4922, "term":
False, "branch": "MAIN"}, "Siam": {"led": 12, "lat": 48.3883, "lon": -
4.4889, "term": False, "branch": "MAIN"}, "LibertÃ©": {"led": 13, "lat":
48.3905, "lon": -4.4855, "term": False, "branch": "MAIN"}, "Jean Jaures":
{"led": 14, "lat": 48.3924, "lon": -4.4826, "term": False, "branch":
"MAIN"}, "Saint Martin": {"led": 15, "lat": 48.3946, "lon": -4.4793,
"term": False, "branch": "MAIN"}, "Octroi": {"led": 16, "lat": 48.3967,
"lon": -4.4761, "term": False, "branch": "MAIN"}, "Pillier Rouge": {"led":
17, "lat": 48.3992, "lon": -4.4722, "term": False, "branch": "MAIN"},
"Place de Strasbourg": {"led": 18, "lat": 48.4028, "lon": -4.4687, "term":
False, "branch": "MAIN"}, "Menez Paul": {"led": 19, "lat": 48.4073, "lon":
-4.4679, "term": False, "branch": "MAIN"}, "Europe": {"led": 20, "lat":
48.4112, "lon": -4.4582, "term": False, "branch": "MAIN"}, "PontanÃ©zen":
{"led": 21, "lat": 48.4161, "lon": -4.4687, "term": False, "branch":
"MAIN"}, "Eau Blanche": {"led": 22, "lat": 48.4173, "lon": -4.4597,
"term": False, "branch": "MAIN"}, "Kerlaurent": {"led": 23, "lat":
48.4187, "lon": -4.4521, "term": False, "branch": "MAIN"}, "Porte de
Guipavas": {"led": 24, "lat": 48.4194, "lon": -4.4462, "term": True,
"branch": "MAIN"}, # --- BRANCHE GOUESNOU (Ruban 2) ---
"Mesmerrien": {"led": 0, "lat": 48.4201, "lon": -4.4674, "term": False,
"branch": "GOUES"}, "Kergaradec": {"led": 1, "lat": 48.4252, "lon": -

```

```
4.4669, "term": False, "branch": "GOUES"}, "Porte de Gouesnou": {"led": 2,
"lat": 48.4295, "lon": -4.4676, "term": True, "branch": "GOUES"}
}
```

```
while True:
```

```
try:
    url_fresh = f"{URL_BASE}?t={int(time.time())}"
    response = requests.get(
url_fresh, headers={'Cache-Control': 'no-cache'}, timeout=10)
```

```
    feed = gtfs_realtime_pb2.FeedMessage()
    feed.ParseFromString(response.content)
    data_dict = MessageToDict(feed)
```

```
# 1. RESET des bandeaux
```

```
pixels_main.fill((0, 0, 0))
pixels_gouesnou.fill((0, 0, 0))
```

```
# Initialisation Terminus en Bleu
```

```
for name, info in STATIONS_DATA.items():
    if info['term']:
        if info['branch'] == "MAIN":
            pixels_main[info['led'] + OFFSET] = (0, 0, 50)
        else:
            pixels_gouesnou[info['led']] = (0, 0, 50)
```

```
print(f"\n--- MAJ : {time.strftime('%H:%M:%S')} ---")
```

```
if 'entity' in data_dict:
    for entity in data_dict['entity']:
        if 'vehicle' in entity:
            v = entity['vehicle']
            if v.get('trip', {}).get('routeId', '') in ['A', '1', 'a']:
                v_lat =
float(v['position']['latitude'])
                v_lon = float(v['position']['longitude'])
                v_id = v.get('vehicle', {}).get('id', '????')[-4:]
```

```
# Trouver la station la plus proche
```

```
nearest_name, min_dist = None, 999
for name, info in STATIONS_DATA.items():
    dist = sqrt((v_lat - info['lat'])**2 + (v_lon - info['lon'])**2)
    if dist < min_dist:
        min_dist, nearest_name = dist, name
```

```
if nearest_name:
```

```
    info = STATIONS_DATA[nearest_name]
    dist_m = int(min_dist * 111139)
```

```
# Seuil strict : 0.001 (~110m)
```

```
seuil = 0.004 if info['term'] else 0.001
```

```
if min_dist < seuil:
    color = (255, 100, 145) if info['term'] else (255,
0, 0)
```

```
if info['branch'] == "MAIN":
    pixels_main[info['led'] + OFFSET] = color
else:
    pixels_gouesnou[info['led']] = color
    print(f"TRAM {v_id} -> [€ QUAI] {nearest_name}")
else:
```

```
print(f"TRAM {v_id} -> [EN ROUTE] vers {nearest_name} ({dist_m}m)")  
  
except Exception as e:  
    print(f"Erreur : {e}")  
    time.sleep(30)
```