

Projet d'une Web Radio portable

Objectif :

Il s'agit de pouvoir écouter en stéréo par casque ou par haut-parleurs des stations de radio internet codées en mp3 ou AAC. Vous contrôlerez le choix des stations, le volume, la tonalité et l'effet de spatialisation du son depuis votre téléphone.

Voici 2 liens pour obtenir les url :

<http://fluxradios.blogspot.com/>
<https://fmstream.org/index.php?c=FT>

Séances programmées de la SAE :

Temps alloué encadré : CM : 1.5 heure TD : 1*1.5h = 1.5 heures TP : 2*3.5h = 7 heures TOTAL : 10 heures	Temps alloué en autonomie : TP : 3*3.5h = 10.5 heures
---	--

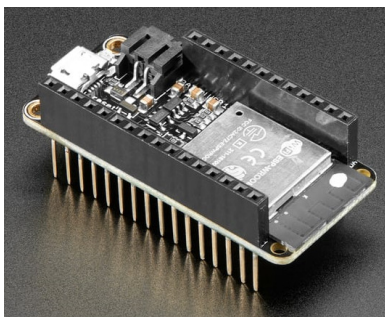
Temps global : 20.5 heures.

SOMMAIRE

1.	Partie 1 : Environnement matériel Adafruit	1
2.	Partie 2 : Environnement logiciel IDE Arduino	2
3.	Partie 3 : Le CoDec VS1053	3
4.	Partie 4 : La carte Adafruit Music Maker FeatherWing	5
5.	Partie 5 : Premier exemple d'une WebRadio	5
6.	Partie 6 : Améliorations au niveau du son du programme de la partie 5	6
7.	Partie 7 : Amélioration au niveau de la gestion du WiFi du programme de la partie 6	7
8.	Partie 8 : Amélioration au niveau des « url » du programme de la partie 7	7
9.	Partie 9 : Protocole de communication MQTT	7
10.	Partie 10 : Projet final	9
11.	Extensions possibles	11
12.	Capture de ma dernière version compilée indiquant les bibliothèques utilisées et leur version	11

1. Partie 1 : Environnement matériel Adafruit

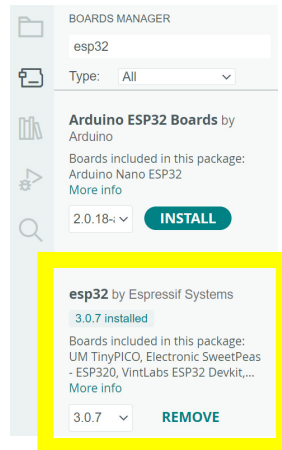
Vous utiliserez une carte Adafruit qui contient un ESP32 : la carte Adafruit HUZZAH32 ESP32 Feather :



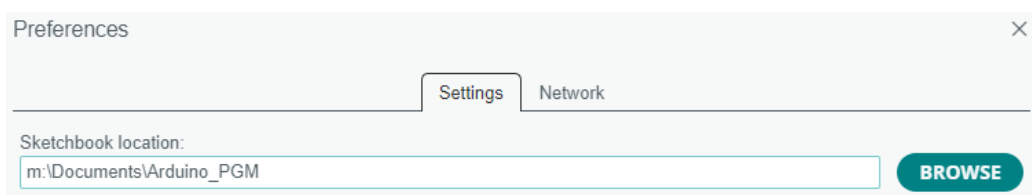
Product ID : 3619

Vous pouvez vous référer à leur site plutôt bien construit pour obtenir les informations nécessaires :

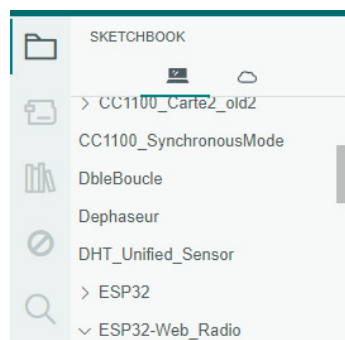
<https://learn.adafruit.com/adafruit-huzzah32-esp32-feather>



Il est conseillé de mettre l'ensemble de vos programmes dans un dossier, par exemple « Arduino_PGM » et de l'indiquer au logiciel en allant depuis le menu dans « File ➔ Preference » :

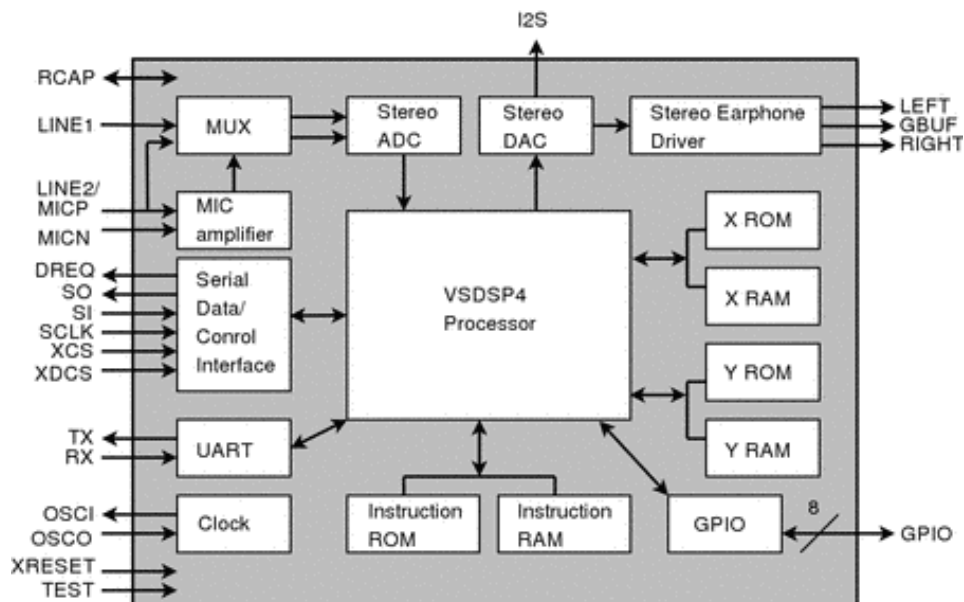


Ainsi, vous aurez la possibilité de les exploiter depuis la rubrique « SKETCHBOOK » :



3. Partie 3 : Le CoDec VS1053

Il s'agit de comprendre comment utiliser le CoDec VS1053 de VLSI dont vous trouverez le diagramme fonctionnel ci-dessus :



- Q7 : Donner les 6 broches qui constituent le SPI pour le VS1053.
- Q8 : Que signifie SDI et SCI ? Indiquer les broches qui sont dédiées au SDI et au SCI.

4. **Partie 4 : La carte Adafruit Music Maker FeatherWing**

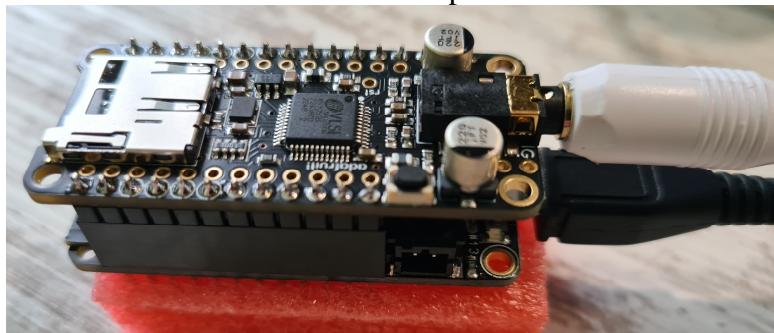
Vous trouverez les informations de cette carte sur le lien suivant :

<https://learn.adafruit.com/adafruit-music-maker-featherwing>

Des exemples sont proposés pour écouter de la musique depuis une carte SD par exemple. Vous avez aussi leur propre bibliothèque « Adafruit_VS1053 » qu'on n'utilisera pas au profit d'une autre (Baldrum notamment)...

On s'intéresse maintenant aux broches de cette carte et notamment celles du SPI (SDI + SCI).

- Q9 : Repérer les 6 broches du SPI de cette carte Adafruit Music Maker FeatherWing et indiquer avec quelles broches de la carte Adafruit HUZZAH32 elles sont connectées, lorsque les deux cartes sont l'une sur l'autre comme le montre la photo suivante :



Carte Adafruit Music Maker FeatherWing	Carte Adafruit HUZZAH32
MISO	
MOSI	
SCK	
MP3CS	
DREQ	
XDCS	

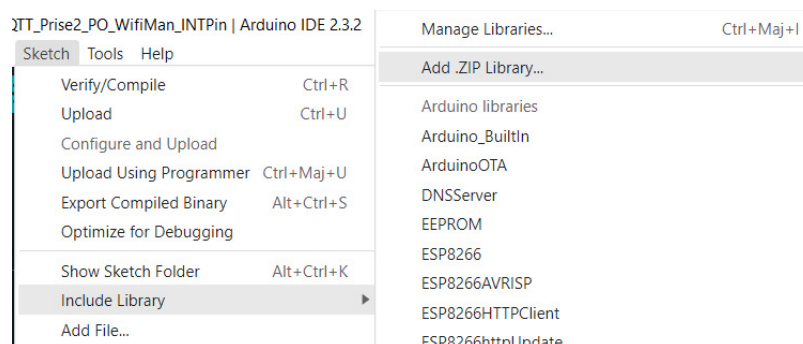
5. **Partie 5 : Premier exemple d'une WebRadio**

Pour mener à terme ce projet, je suis parti du site de Yves PELLETIER, qui lui-même s'est fortement inspiré du travail fait par Vince GELLAR, qui lui aussi a récupéré l'excellente contribution de Marcin Szalowski sous le pseudo de Baldrum :

Bibliothèque Baldrum : https://github.com/baldrum/ESP_VS1053_Library

Site Yves PELLETIER : <http://electroniqueamateur.blogspot.com/2021/03/esp32-et-vs1053-ecouter-la-radio-sur.html>

La bibliothèque de Baldrum n'est pas reconnue par l'IDE Arduino. Il vous faudra donc la rajouter à la « main » de la manière suivante :



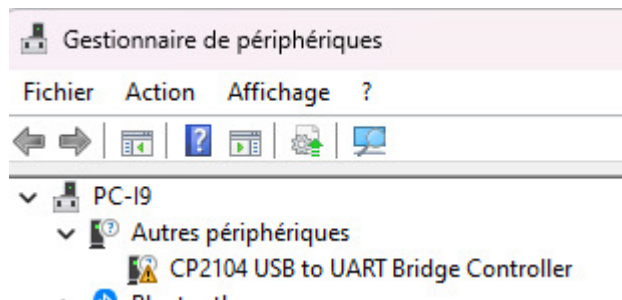
- Q10 : Reprendre le programme proposé par Yves PELLETIER et modifier-le en tenant compte, notamment, des cartes que vous avez.

[V_PGM_1] Faire valider le fonctionnement par l'enseignant !

- Q11 : Modifier les « url » des stations selon vos préférences...

Aides... :

- Penser à sélectionner la carte « Adafruit ESP32 Feather » ainsi que le COM actif proposé.
- S'il y a une erreur de téléversement, c'est certainement lié au fait que la carte n'est pas totalement reconnue. Dans ce cas, ouvrez le Gestionnaire de périphériques :



Ainsi, il faut installer le driver à « la main » depuis le Gestionnaire de périphériques en vous aidant du lien suivant :

<https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/using-with-arduino-ide>

Pour finir, il faudra certainement (re)sélectionner le bon port.

6. Partie 6 : Améliorations au niveau du son du programme de la partie 5

Vous allez rajouter les fonctionnalités suivantes qui seront disponibles depuis le clavier :

Contrôles:

```
n[+] / v[-]: synthoniser une autre chaine
y[+] / b[-]: contrôle du volume
g[+] / f[-]: contrôle des basses
j[+] / h[-]: contrôle des aigus
d: tonalité par défaut
s: spatialisation
```

Le chapitre 9 du datasheet et autres recherches sur le net vous permettront de répondre aux questions suivantes :

- Q12 : Pour le réglage des basses et des aigus il faudra utiliser la fonction disponible de la bibliothèque de Baldram.

[V_PGM_2] Faire valider le fonctionnement par l'enseignant !

- Q13 : Pour les 4 modes de spatialisation et comme il n'y a pas de fonction toute faite, il faudra la créer en passant par les registres appropriés du VS1053 (cf. diaporama TD R2.06).

[V_PGM_3] Faire valider le fonctionnement par l'enseignant !

7. **Partie 7 : Amélioration au niveau de la gestion du WiFi du programme de la partie 6**

L'idée est de pouvoir connecter votre lecteur sur n'importe quel réseau WiFi disponible dont vous connaissez le mot de passe (mdp).

Ainsi, il existe un outil via la bibliothèque « WiFiManager » de tzapu qui permet de vous connecter à votre réseau de prédilection sans pour autant mettre les identifiants et mdp dans le programme ! Le site suivant vous explique son fonctionnement :

<https://dronebotworkshop.com/wifimanager/>

- Q14 : Suivez les instructions du site.
- Q15 : Modifier le programme de la partie 6 pour que vous puissiez définir le réseau auquel vous connecter.

[V_PGM_4] Faire valider le fonctionnement par l'enseignant !

8. **Partie 8 : Amélioration au niveau des « url » du programme de la partie 7**

L'idée est de pouvoir donner l'url complète sans devoir la partitionner en 3 parties (host, path et port). A cet effet, installer la bibliothèque « ESP32_VS1053_Stream » de Cellie depuis l'IDE Arduino ou à la main si elle ne veut pas depuis l'IDE Arduino :

https://github.com/CelliesProjects/ESP32_VS1053_Stream

- Q16 : Avec quelle version des cartes ESP32 cette librairie est-elle compatible ?
- Q17 : Depuis l'IDE Arduino, faire fonctionner le programme d'exemple « simple » de cette nouvelle bibliothèque.
- Q18 : Modifier le programme de la partie 7 en s'appuyant sur le programme testé précédemment pour répondre au cahier des charges de cette partie.

NB :

Le réglage de la tonalité et de la spatialisation ne fonctionneront certainement plus avec cette nouvelle bibliothèque. Néanmoins, comme elle dépend de celle de Baldram, vous pourrez apporter les modifications nécessaires pour que ces réglages fonctionnent de nouveau !

[V_PGM_5.1] Faire valider le fonctionnement par l'enseignant [uniquement les url] !

[V_PGM_5.2] Faire valider le fonctionnement par l'enseignant [url + tonalité] !

[V_PGM_5.3] Faire valider le fonctionnement par l'enseignant [url + tonalité + spatialisation] !

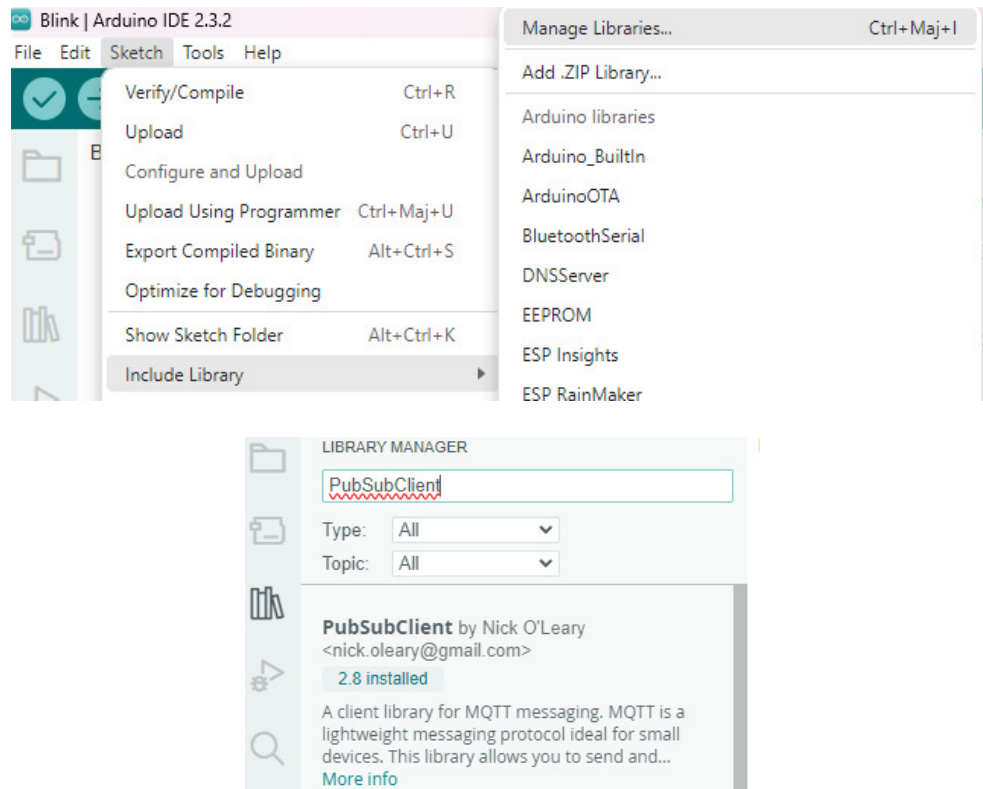
9. **Partie 9 : Protocole de communication MQTT**

Il s'agit de comprendre le protocole MQTT et de savoir l'utiliser en répondant aux questions suivantes à travers vos recherches sur le net :

- Q19 : Donnez les caractéristiques principales et le principe de fonctionnement du MQTT.
- Q20 : Citez quelques brokers publics.
- Q21 : Quels logiciels gratuits permettent d'utiliser le MQTT sous Windows, Android et IOS ?
- Q22 : Faites des essais avec ces logiciels...
- Q23 : Déduisez-en l'avantage principal du MQTT.

Depuis « Manage Libraries », installer la bibliothèque « PubSubClient » de Nick O'LEARY :

<https://pubsubclient.knolleary.net/api>



Ouvrir le programme d'exemple « mqtt_esp8266 » de la bibliothèque « PubSubClient ».

Modifier les quelques lignes suivantes pour que le programme puissent fonctionner :

- Changer la bibliothèque de gestion du Wifi d'un ESP8266/ESP12 à un ESP32 :

```
21  // #include <ESP8266WiFi.h>
22  #include <WiFi.h>
23  #include <PubSubClient.h>
```

- Remplacer le « %ld » de la ligne suivante par « %d » :

```
125  | | snprintf (msg, MSG_BUFFER_SIZE, "hello world #%d", value);
```

- Q24 : Faire **fonctionner** le programme tout en comprenant ce qu'il fait !

[V_MQTT] Faire valider le fonctionnement par l'enseignant !

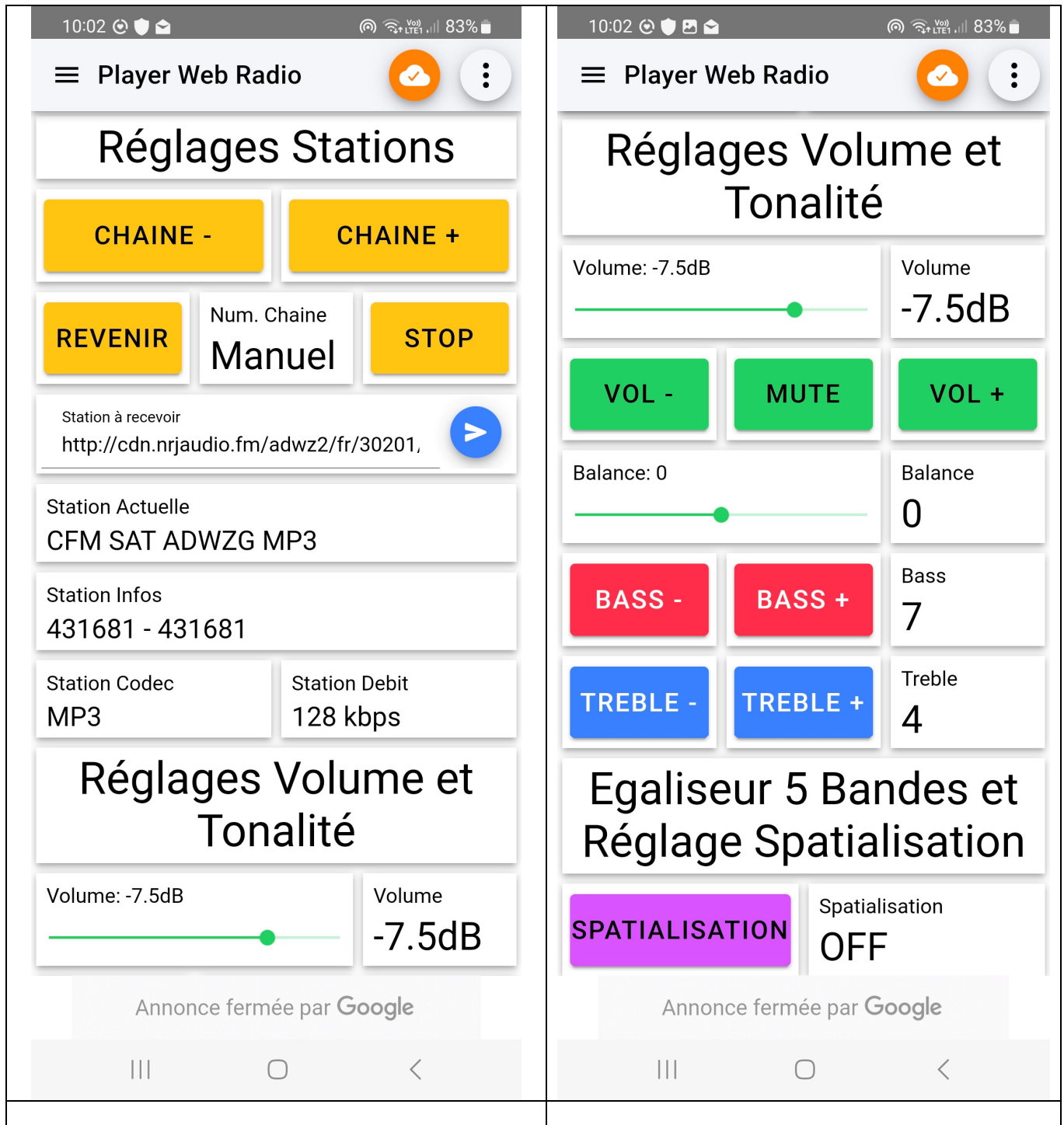
Quelques points importants à faire pour assurer le bon fonctionnement :

- Configurer votre téléphone en mode « Point d'accès mobile » ou équivalent. Récupérez « le nom du réseau » et « le mot de passe » pour les mentionner dans le programme.
- Enlever la possibilité d'avoir le « Point d'accès mobile » en 5 GHz, seule la bande des fréquences en 2.4 GHz doit être autorisée.
- Respecter les règles d'utilisation liées au MQTT (identité unique, topic différent entre les binômes, etc.).

10. Partie 10 : Projet final

Si vous êtes arrivés à cette partie c'est que votre projet permet, depuis le clavier, de gérer le volume, la tonalité, la spatialisation et le changement de chaînes/stations/urls préprogrammées.

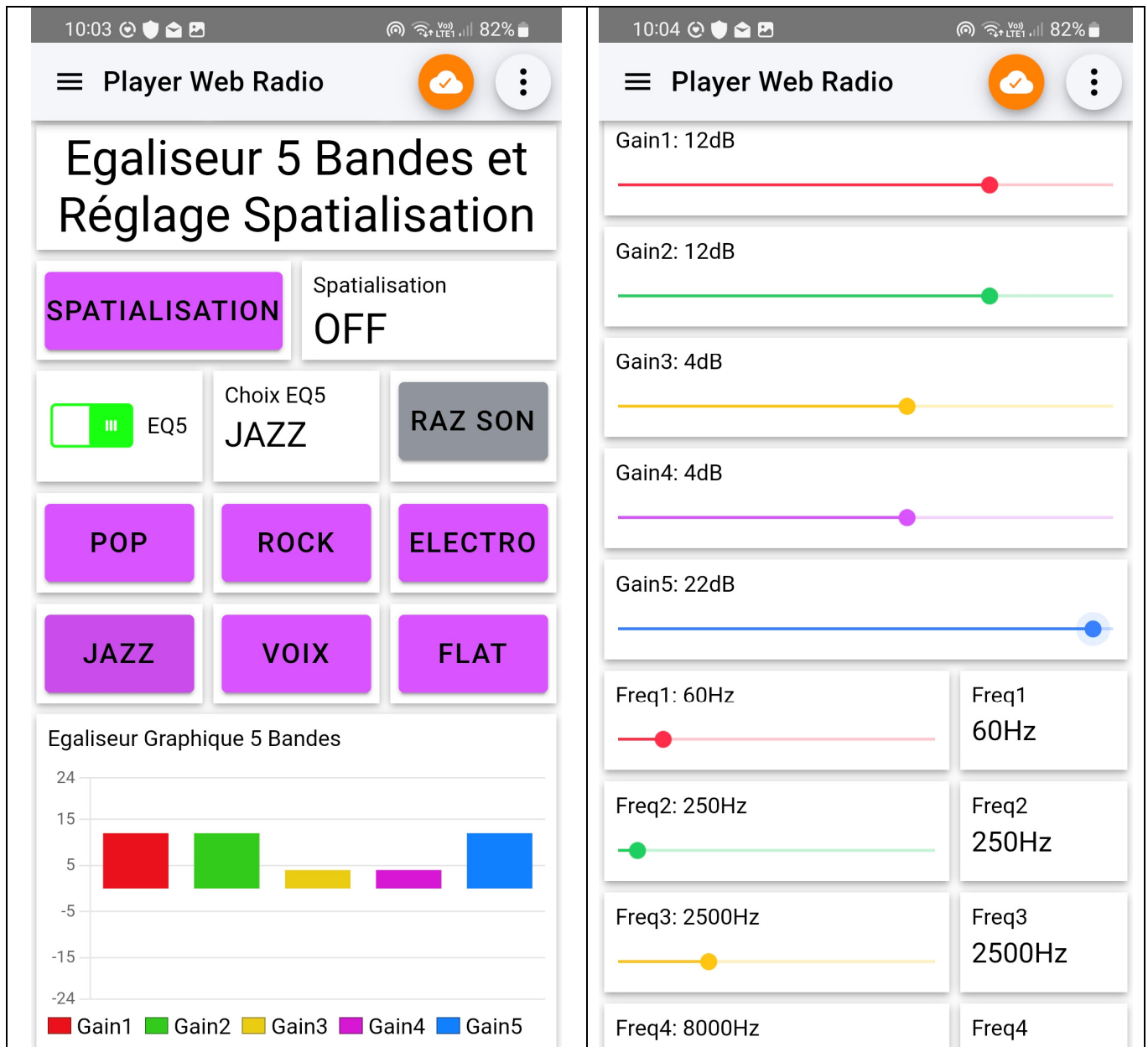
Il serait maintenant intéressant d'avoir toutes ses fonctionnalités depuis votre téléphone. Cela est possible en utilisant le MQTT et l'application Android « IoT MQTT Panel ». Voici ce à quoi on peut s'attendre au niveau de l'interface graphique avec « IoT MQTT Panel » :



Et si jamais vous utilisez le PLUGIN de l'égaliseur graphique 5 bandes [VS1053b EQ5 Plugin] du lien suivant... :

<https://www.vlsi.fi/en/support/software/vs10xxplugins.html>

... vous pouvez obtenir :



- Q25 : Modifier le programme de la partie 8 afin de pouvoir interagir sur le lecteur depuis cette interface Android.

[V_PGM_6] Faire valider le fonctionnement par l'enseignant !

- Q26 : Rajouter la possibilité de pouvoir écouter une chaîne en mettant l'url dans la cellule « Station à recevoir » et de « Revenir (à la station préprogrammée) » :

REVENIR

Num. Chaîne
Manuel

STOP

Station à recevoir
[http://cdn.nrjaudio.fm/adwz2/fr/30201,](http://cdn.nrjaudio.fm/adwz2/fr/30201)

>

[V_PGM_7] Faire valider le fonctionnement par l'enseignant !

- Q26 : faire en sorte que lorsque vous gérez votre lecteur depuis le clavier, les informations soient modifiées de manière dynamique sur votre téléphone !

[V_PGM_8] Faire valider le fonctionnement par l'enseignant !

11. Extensions possibles

Si vous avez fini, voici quelques pistes d'amélioration sans ordre particulier :

- Faire une vraie application sous Android ou App Store.
- Faire une page web en s'inspirant d'un player comme Squeezelite sous LMS ou équivalent (Volumio, etc.).
- Rajouter des boutons et/ou potentiomètres pour le réglage du volume, de la tonalité, de la balance, des stations pré-enregistrées, etc.
- Rajouter un afficheur LCD ou OLED (cf. solution TTGO ou équivalent).
- Implémenter l'égaliseur graphique 5 bandes « EQ5 » proposé dans le sujet.
- Pouvoir lire des morceaux de musique depuis une carte SD.

12. Capture de ma dernière version compilée indiquant les bibliothèques utilisées et leur version

```
esptool.py v4.8.1
Wrote 0x400000 bytes to file C:\Users\compte.local\AppData\Local\arduino\
Utilisation de la bibliothèque VS1053 Library for ESP8266/ESP32 version 1
Utilisation de la bibliothèque SPI version 3.2.0 dans le dossier: C:\Users\
Utilisation de la bibliothèque PubSubClient version 2.8 dans le dossier: C
Utilisation de la bibliothèque WiFiManager version 2.0.17 dans le dossier
Utilisation de la bibliothèque WiFi version 3.2.0 dans le dossier: C:\User
Utilisation de la bibliothèque Networking version 3.2.0 dans le dossier: C
Utilisation de la bibliothèque Update version 3.2.0 dans le dossier: C:\Us
Utilisation de la bibliothèque WebServer version 3.2.0 dans le dossier: C
Utilisation de la bibliothèque FS version 3.2.0 dans le dossier: C:\Users\
Utilisation de la bibliothèque DNSServer version 3.2.0 dans le dossier: C
Utilisation de la bibliothèque ESP32 Async UDP version 3.2.0 dans le dossi
Utilisation de la bibliothèque ESP32_VS1053_Stream version 2.1.2 dans le c
Utilisation de la bibliothèque HTTPClient version 3.2.0 dans le dossier: C
Utilisation de la bibliothèque NetworkClientSecure version 3.2.0 dans le c
"C:\\Users\\compte.local\\AppData\\Local\\Arduino15\\packages\\esp32\\too
Le croquis utilise 1196250 octets (91%) de l'espace de stockage de program
Les variables globales utilisent 53312 octets (16%) de mémoire dynamique,
```

Barème sujet à modifications :

Les validations ne seront acceptées que si vous êtes capable d'expliquer ce que vous avez fait !

VALIDATION GROUPEE 1			
V_PGM_1	V_PGM_2	V_PGM_3	V_PGM_4
PELLETIER SPI	V_PGM_1 + tonalité	V_PGM_2 + spatialisation	V_PGM_3 + WiFiManager
1	2	2	1

VALIDATION GROUPEE 2			
V_PGM_5.1	V_PGM_5.2	V_PGM_5.3	V_PGM_6
V_PGM_4 + url	V_PGM_5.1 + tonalité	V_PGM_5.2 + spatialisation	V_PGM_5.3 + MQTT
1	2	2	5

VALIDATION GROUPEE 2 (suite)			
V_PGM_7	V_PGM_8	V_MQTT	
V_PGM_6 + url depuis application	V_PGM_7 + retour		
2	1	1	