Frédéric Bouthot, Paul Hachey-Perron et Edward Richer

Communications numériques sans-fil

243-510-MA

Groupe : 0002

**Mise en fonction : Système de notification externe pour réseaux sociaux**

Travail présenté à

Messieurs Frédéric Daigle et David Legris Tremblay

Collège Maisonneuve

Département des Technologies du Génie Électrique

Le 14 Décembre 2020

# Réalisation Du Projet :

Afin d’accomplir le projet, il faut ajouter la libraire arduino-instagram-stats[[1]](#footnote-1) et json-streaming-parser[[2]](#footnote-2) au compilateur Arduino. Après avoir ajouter ces deux librairies, il faut remplacer le nom du réseaux Wi-Fi (ssid[ ]) et son mot de passe (password[ ]) ainsi que le nom d’utilisateur Instagram (userName). De plus, il faut brancher une DEL sur la broche 13 du ESP32. Après avoir modifié ces paramètres, téléverser le programme Test-Instagram.ino (contenu dans le même GitHub que le présent document) dans la carte ESP32 Feather. Après avoir mis le programme dans la carte, il faut ouvrir le serial monitor du ESP32 afin de pouvoir voir le nombre d’abonné de la page Instagram. À chaque fois que le nombre d’abonné change, il y aura une DEL qui allumera. Il faut faire attention de téléverser le programme le moins de fois possible dans la carte ESP32 (maximum 3 fois) sinon la carte ESP32 sera banni des serveurs Instagram.

# Réalisation Du Circuit Audio :

Afin de pouvoir lire des fichiers audio mp3 sur l’esp32, il a fallu tout d’abord réaliser le circuit présenté dans le dossier GitHub. Si vous portez une attention particulière à la fiche technique du DAC (CS4334-DSZ) vous vous rendrez vite compte que nous n’avons pas tenu compte des spécifications de ceux-ci. La raison est fort simple, le circuit audio que nous avions créé avec l’amplificateur audio, changeait la valeur des paramètres requis dû aux condensateurs nécessaires à l’entrée de l’amplificateur. La fiche technique donnait des informations au sujet du DAC, toutefois, ses développeurs n’avaient pas les connaissances du reste du circuit audio ce qui explique les différences notables entre nos valeurs de filtres et les leurs. Un autre aspect qui sera important lors de la mise en route du projet sera de faire attention au master clock présent dans l’interface I2s entre le DAC et l’ESP32. Celle-ci doit être absolument à 5.6448 MHz, soit 128 X fois la fréquence d’échantillonnage (44.1KHz). Votre DAC se fit sur cette fréquence afin de déterminer la fréquence d’échantillonnage choisi par l’esp32, une fréquence moindrement trop petite ou trop grande empêchera le fonctionnement du circuit audio. Afin de faire un test du circuit audio, un programme vous est fourni ainsi que les librairies utilisées dans le GitHub.

À titre informatif, les haut-parleurs sont des haut-parleurs 4 Ω et 15 W. Cela devrait pouvoir mieux vous guider dans votre choix de haut-parleur. Vous remarquerez que les gains ont été branchés en valeur 00 ce qui donne un gain de 20 dB, il vous est toutefois possible de monter ce gain si vous ne considérez pas le son comme étant assez fort ou de faire des diviseurs de tensions aux entrées afin de réduire l’amplitude du signal et ainsi permettre au son d’être moins fort. Finalement, en cas de problème avec la lecture de la carte SD, vous pouvez toujours réviser la matière du cours d’interface puisque la lecture de cette carte se fait via SPI. Lors de la tentative de mise en fonction du programme final, nous avons connu des problèmes avec l’API Instagram, ce qui explique que le programme de lecture soit séparé. Il est important de noter que la librairie utilisée est une des seules vous permettant aussi de faire la conversion de fichier .mp3 en .wav. La raison pour cela est qu’il est impossible de lire un fichier mp3 directement. Étant un fichier compressé, il ne contient pas les niveaux numériques nécessaire pour faire fonctionner directement un DAC. Sans la libraire du projet, il vous faudrait en trouver une autre pour lire des fichiers mp3. Finalement, en cas de nécessité d’utilisation de nouveaux ports de la carte esp32 feather, vous n’aurez qu’à changer les define en début de programme et celui-ci devrait continuer à fonctionner comme si de rien n’était.

1. <https://github.com/witnessmenow/arduino-instagram-stats> [↑](#footnote-ref-1)
2. <https://github.com/squix78/json-streaming-parser> [↑](#footnote-ref-2)