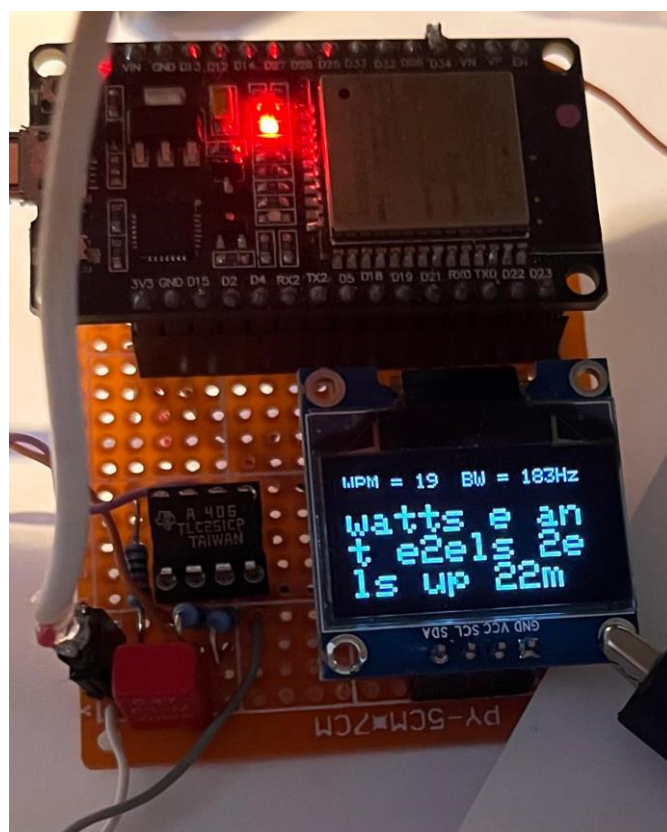


Décodeur Morse avec affichage OLED

Alain Bertout F6KFA

Mai 2025

Ce décodeur Morse est basé sur une première solution technique proposée par Hjalmar Skovholm Hansen OZ1JHM. <https://github.com/G6EJD/ESP32-Morse-Decoder>. D'utiles modifications ont été apportées pour l'optimisation de l'affichage, l'amplification réglable en entrée, l'offset avant la conversion analogique/numérique et la mise en œuvre d'un tableau `morseCode` d'accès plus rapide.



Montage sur une plaque à trous

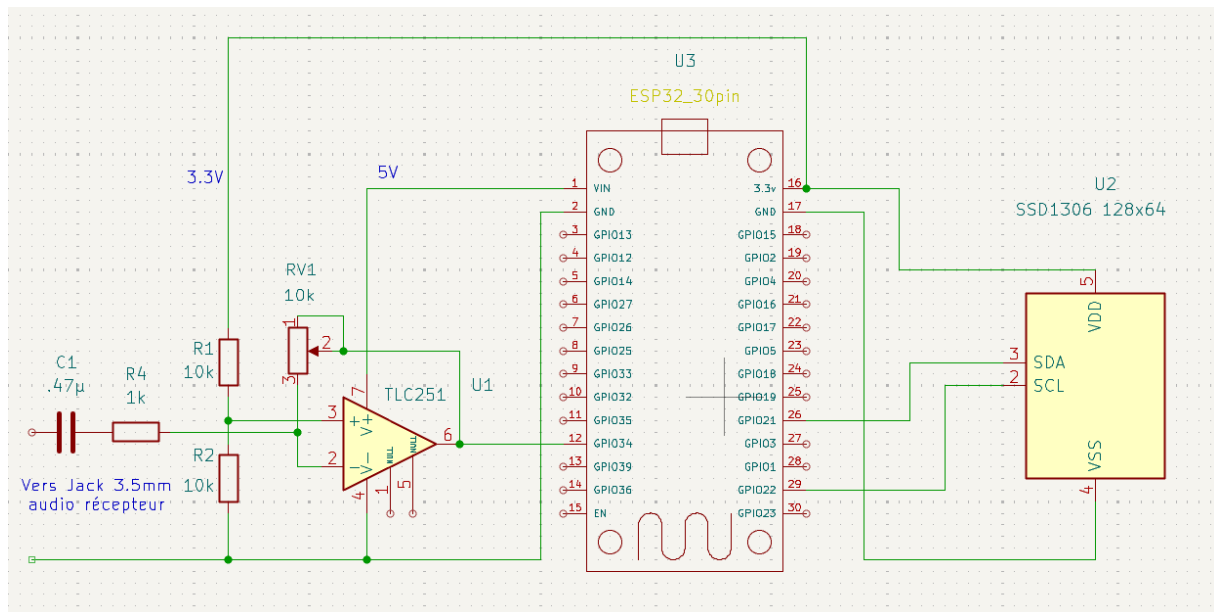


Schéma électrique du décodeurMorse

Le schéma électrique

Le montage électronique utilise un microcontrôleur ESP32 Dev Kit alimenté en 5V par l'interface USB. Une liaison I2C raccorde l'ESP32 à un afficheur 128x64 de type SS1306 par ses liaisons SDA et SCL (adresse 0x3C). Les signaux Morse du récepteur radio peuvent provenir d'une prise écouteur 3.5mm ou d'un micro électret positionné devant l'hautparleur. Un pont diviseur assure un offset de 2.65V destiné à centrer le signal analogique dans la plage de fonctionnement de l'ADC (0-3.3V). L'ampli opérationnel utilisé est du type TLC251 conçu pour être alimenté et fonctionner entre 0 et 5V. Un potentiomètre permet de régler l'amplification jusqu'à un gain de 10. Un condensateur bloque toute composante continue avec le récepteur radio.

Le code ESP32

Ce code constitue un décodeur Morse autonome basé sur l'emploi d'un ESP32. Il utilise un algorithme de Goertzel pour détecter la présence ou non d'une tonalité (600 Hz) typique du Morse audio. Il affiche en temps réel le texte décodé ainsi que la vitesse de transmission en mots / minutes (WPM).

Bibliothèques utilisées :

- `Adafruit\_GFX` et `Adafruit\_SSD1306` : Pour l'affichage OLED.
- `Wire` : Pour la communication I2C.

Définition des constantes

- Taille de l'écran OLED : 128x64 pixels.
- `NUM\_MORSE\_CODES` : 36 codes Morse (A-Z + 0-9).
- `num\_chars` : Longueur maximale de la ligne affichée.

Tableau Morse

Un tableau `morseCode` associant directement les symboles Morse à leurs caractères alphabétiques ou numériques.

Variables globales

- ``display`` : Objet de type ``Adafruit_SSD1306``.
- ``magnitude`` : Amplitude du signal détecté par l'algorithme de Goertzel.
- ``realstate`` / ``filteredstate`` : Etats binaires du signal (HIGH ou LOW) filtrés.
- ``sampling_freq`` : Fréquence d'échantillonnage (~24 kHz).
- ``target_freq`` : Fréquence cible (600Hz).
- ``n`` : Taille de la fenêtre Goertzel (128 échantillons).
- Buffers : ``testData[]``, ``CodeBuffer``, ``DisplayLine``.

setup()

- Configure l'ADC, le bus I2C, l'écran OLED et les constantes pour l'algorithme de Goertzel.
- Calcule les constantes ``sine``, ``cosine``, ``coeff`` nécessaires à l'algorithme.

loop()

Acquisition

- Echantillonnage de ``n`` valeurs depuis la broche GPIO34.
- Calcul de la moyenne numérique (``mean``) pour supprimer l'offset DC (1,65V).

Algorithme de Goertzel

- Appliqué sur le signal numérique provenant de l'ADC – la valeur moyenne (``mean``).
- Calcule la magnitude du signal autour de 600Hz (valeur efficace)

Seuil dynamique

- ``magnitudelimit`` s'adapte à l'intensité du signal.

Filtrage d'état

- Détection des transitions basée sur la magnitude.
- Anti-rebond logiciel avec temporisation (``nbtime``) sur 6ms.

Mesure des durées HIGH et LOW (son / silence)

- Calcul de la durée des signaux actifs et inactifs.
- Moyenne glissante sur ``hightimesavg``.

Décodage Morse

- Point : signal court (``.``).
- Trait : signal long (``-``).
- Fin de caractère : silence moyen.
- Fin de mot : long silence.

Affichage OLED

- Mise à jour tous les nouveaux caractères.
- Affiche le WPM, la bande passante BW et la ligne de texte décodée.

``CodeToChar()``

- Compare le contenu de ``CodeBuffer`` avec le tableau Morse.
- Si correspondance, ajoute le caractère nouveau à l'affichage.

``AddCharacter(char)``

- Décale la ligne affichée vers la gauche et ajoute le nouveau caractère.
- Rafraichit l'écran OLED.

Analyse des performances

Vitesse d'échantillonnage

Une boucle *'for'* de lecture des 128 échantillons tourne au maximum à 24 kHz (durée = 5.4 ms) ; elle est insérée dans chaque tour du programme *'loop'* qui dure 6 ms. Le signal Morse de fréquence 600Hz est donc mesuré sur une durée de 3 périodes. La durée d'un signal morse (point) est de 60ms pour un WPM de 20 mots/minute, ce qui correspond à 15 boucles *'loop'*. Cela donne le temps de faire les détections de seuils, détections d'états, calcul des durées (traits et points, silences caractère et silences mots), décodage morse et affichage à la sortie de chaque nouveau caractère.

La détection à la fréquence cible (600 Hz) nécessitera d'ajuster un peu le tuner du récepteur pour centrer la réception du signal proche de cette fréquence.

Les réglages possibles :

- N = nombre d'échantillons (= 128)
- fc = Fréquence cible (= 600Hz)
- Fs = fréquence d'échantillonnage (24 kHz)

Cela joue sur la largeur de bande du filtre passe bande de Goertzel

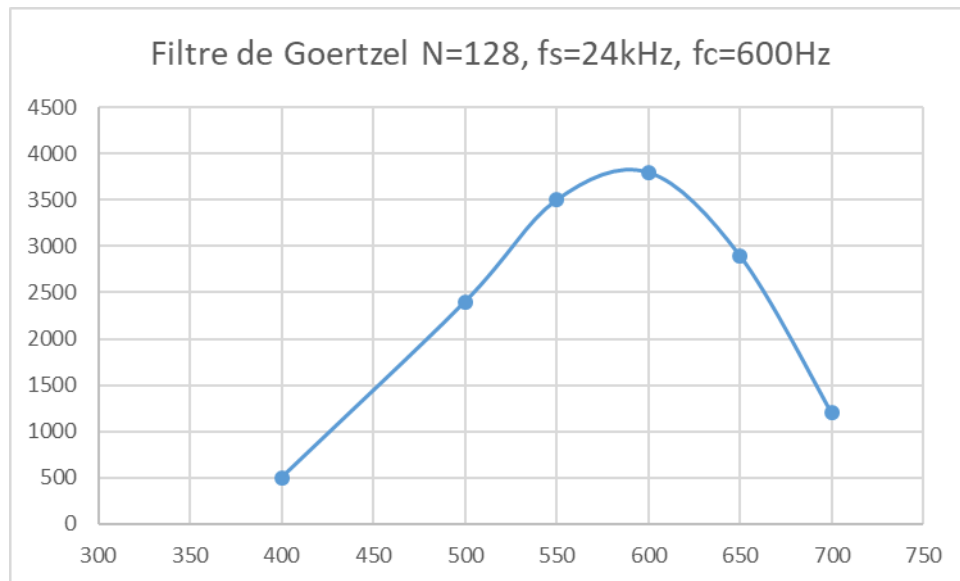
Bande passante

Le filtre de Goertzel est équivalent à un filtre passe-bande très étroit, centré sur une fréquence précise, il fournit en sortie la magnitude du signal (enveloppe). La largeur de bande du filtre dépend de la durée d'intégration. Le bénéfice de ce filtre passe bande est de réduire les bruits de fond de la réception radio. Il demeure toutefois utile d'utiliser les autres filtres disponibles dans le récepteur (squelch, notch, dnr).

La largeur de bande approximative (à -3 dB) est donnée par :

$$\Delta f \approx \frac{1.5}{T} = \frac{1.5 \cdot f_s}{N}$$

Dans notre cas : $\Delta f = 270$ Hz (soit de 330 Hz à 870 Hz) ce qui couvre une bonne largeur de la bande habituelle du morse (en SSB, le ton de battement se trouve typiquement entre 400 et 800 Hz).



Mesure de la bande passante = 200Hz à 3dB

Formule de base du Morse

Le Morse standard (modèle PARIS) définit qu'un mot contient 50 unités de temps :

- 1 point = 1 unité (t_{point})
- 1 trait = 3 unités
- espace entre éléments (d'un même caractère) = 1 unité
- espace entre caractères = 3 unités
- espace entre mots = 7 unités

Mots par minutes

La solution s'adapte à la vitesse de manipulation Morse des opérateurs (WPM entre 15 et 35)

Le nombre de mots par minute (WPM) se calcule comme suit :

$$\text{WPM} = \frac{60}{t_{\text{dot}} \times 50}$$

Application : pour $t_{\text{point}} = 60 \text{ ms}$, WPM = 20 mots / minutes

Note :

Quelle est la différence entre la DFT (transformé de Fourier discrète) et le filtre de Goertzel ?

L'algorithme de Goertzel est un algorithme utilisé en traitement du signal pour détecter la présence d'une fréquence dans une séquence d'échantillons. Il fut publié par le physicien américain, Gerald Goertzel (université de New York), en 1958

Comme pour la DFT, l'algorithme itératif de Goertzel analyse la composante fréquentielle d'un signal discret. Contrairement aux calculs dans la DFT, l'algorithme de Goertzel est plus simple car il applique un seul coefficient réel à chaque itération, en utilisant uniquement ses calculs sur les grandeurs d'entrée réelles.

Pseudocode Goertzel :

N nombre d'échantillons (128), Fs Fréquence d'échantillonnage (24kHz), fc fréquence cible (600Hz)

$K = f_c / F_s$, $\omega = 2 \times \pi \times K / N$, coeff = $2 \times \cos(\omega)$

Calcul par itération :

$Q_{n-1} = 0$, $Q_{n-2} = 0$, $x[n]$ = valeur de l'échantillon n

Pour chaque n de 0 à N-1 faire :

$$Q_n = x[n] + \text{coeff} \times Q_{n-1} - Q_{n-2}$$

$$Q_{n-2} = Q_{n-1}$$

$$Q_{n-1} = Q_n$$

fin

$$\text{magnitude}^2 := Q_{n-1} \times Q_{n-1} + Q_{n-2} \times Q_{n-2} - (\text{coeff} \times Q_{n-1} \times Q_{n-2})$$

Mise en œuvre

1. Raccorder l'appareil sur une prise USB 5V par le câble USB-A,
2. Faire un Reset en appuyant sur le bouton poussoir pour effacer l'écran OLED 0.96 pouce et le code en mémoire,
3. Raccorder le câble audio male-male 3.5mm sur la sortie audio du récepteur Morse,
4. Sélectionner la fréquence audio de réception Morse et ajuster le signal autour de 500Hz – 700Hz,
5. Ajuster le niveau de réception autour de 50 % avec le potentiomètre,
6. Utiliser le filtrage DNR si disponible.