

Pour compiler le programme :

Le programme a été écrit en C pour un environnement de type Linux. Pour le compiler il faudra alors utiliser une distribution Linux disposant des compilateurs GCC et G++, et de toutes les librairies de base nécessaire à la compilation.

-> Pré-requis :

Dans Ubuntu il faudra installer le package « build-essential » qui fournit les librairies de base nécessaires à la compilation. Pour se faire on pourra utiliser le gestionnaire de paquets « Synaptic », dans un shell : `sudo apt-get install build-essential`

• Installation de librairies :

Le programme utilise plusieurs librairies qu'il est indispensable d'installer avant de le compiler, et qui seront nécessaires pour l'exécuter.

- **LIBSNDFILE** : Librairie de manipulation de fichiers audio. Disponible :

<http://www.mega-nerd.com/libsndfile/#Download>

Dans les distributions de type Debian, comme Ubuntu avec Synaptic :

`sudo apt-get install libsndfile1 libsndfile1-dev`

- **FFTW3** : Librairie de transformée de Fourier rapides. Disponible :

<http://www.fftw.org/download.html>

Dans les distributions de type Debian, comme Ubuntu avec Synaptic :

`sudo apt-get install fftw3 fftw3-dev`

- **GTK+** : Librairie de création et de gestion d'interfaces graphiques. Disponible:

<http://www.gtk.org/download.html>

Dans les distributions de type Debian, comme Ubuntu avec Synaptic :

`sudo apt-get install libgtk2.0-dev libgtk2.0-0`

• Installation de CodeBlocks :

Le programme a été développé et organisé dans un fichier projet de l'IDE CodeBlocks. Afin de proprement le compiler il sera nécessaire de télé charger et d'installer cet IDE. Il est disponible pour de nombreuses plateformes.

(Disponible: <http://www.codeblocks.org/downloads>)

Une fois ces librairies installées il faudra ouvrir le fichier projet, avec extension « .cbp » contenu dans la racine de l'archive, l'ouvrir avec l'IDE CodeBlocks , lui linker ces librairies (Settings -> Compiler and debugger...->Linker Settings), et le compiler : touche « F9 » ou « Ctrl-F9 ».

Bien que l' usage de Code::Blocks ne sera pas indispensable pour compiler ce projet, je ne dispose pas pour le moment de la commande complète permettant de le compiler en ligne de commande. Elle sera communiquée ultérieurement et attachée à l'archive du programme.

Pour utiliser le programme :

Il faut se placer dans un shell, de type bash, dans le dossier contenant l'exécutable, le fichier binaire, du programme et rentrer la commande:

`./audio <ARGUMENTS> <OPTIONS>`

« audio » est le nom de l'exécutable. Il faudra si nécessaire l'adapter.

<ARGUMENTS> est une liste d'arguments, les arguments sont précédés du symbole « - »

<OPTIONS> est une liste d'options, les options sont précédées du symbole « - »

Liste et explication des arguments :

Aucun argument ou option n'est obligatoire. Par défaut le programme se lancera en mode CLI (interface ligne de commande). Si aucun arguments ou options ne sont fournis le programme commencera alors par les demander à l'utilisateur.

Pour avoir de l'aide il faut utiliser l'argument « -help » ou « -HELP ».

Ces arguments définissent des paramètres importants du programme. Ils peuvent être entrés dans n'importe quel ordre en minuscules ou en majuscules.

-help ou -HELP: affiche le message d'aide.

-m (ou -M): Suivi de 'p' ou 'f'. Définit le mode de traitement et d'analyse.

Deux modes sont disponibles:

'p' ou 'P' définit le mode de traitement et d'analyse sur la puissance moyenne

'f' ou 'F' définit le mode d'analyse et de traitement sur la fréquence. Ex: -m P

-i (ou -I): Suivi du chemin relatif ou absolu d'un fichier. Définit le chemin du fichier à traiter/analyser, il doit être accessible en lecture à l'utilisateur courant. Si on ne spécifie rien il sera demandé par après. Ex: -i /home/root/test.wav

-o (ou -O): Suivi du chemin relatif ou absolu d'un dossier. Définit le chemin du dossier de destination dans lequel seront enregistrés tous les résultats. Ce dossier doit être accessible en lecture/écriture pour l'utilisateur courant. Si on ne spécifie rien il sera demandé par après. Ex: /home/root/results

-g (ou -G): Suivi de 'g' ou 'c'. Définit le type d'interface désirée. Le mode CLI est activé par défaut si il n'est pas spécifié

Deux interfaces sont disponibles:

'c' ou 'C' définit le mode CLI

'g' ou 'G' définit le mode par interface graphique. Ex: -g G

-l (ou -L): Suivi du chemin relatif ou absolu d'un dossier. Définit le chemin du dossier dans lequel sera écrit le fichier log. Ce dossier doit exister. Si il n'est pas spécifié, ou que le chemin n'est pas valide, par défaut il sera créé dans le dossier de destination avec le nom du fichier d'entrée suivi de l'extension .txt. Ex: -l /home/results/

-v (ou -V): Suivi de 'v' ou 'n'. Définit le mode verbeux ou non-verbeux. Ce mode permet d'afficher à l'écran, sur la console, des informations sur le traitement en cours, à des fins de diagnostic.

'v' ou 'V' définit le mode verbeux

'n' ou 'N' définit le mode non-verbeux

Options :

Aucun de ces options n'est obligatoire. Ces options définissent les paramètres des traitements. Ils doivent obligatoirement être en majuscule

-B: Suivi d'un nombre entier. Il est utilisé pour définir la taille des blocs du traitement et de l'analyse. Cette valeur divisera la fréquence d'échantillonnage. Elle doit être comprise entre 100 et 250. Si la valeur fournie est invalide, ou aucune valeur n'est fournie, la valeur de 100 sera prise par défaut. Ex: -B 100

-F: Suivi d'un nombre entier. Définit la fréquence à détecter. Il faut entrer une valeur comprise entre 1 et la moitié de la fréquence d'échantillonnage du fichier. Si la valeur fournie est invalide, ou aucune valeur n'est fournie, elle sera redemandée lors de l'étape du traitement. Ex: -F 440

-T: Suivi d'un nombre entier. Définit la marge de tolérance pour la détection de la fréquence. Ainsi le programme pourra détecter des fréquences supérieures ou inférieures (de ce nombre) à la fréquence à détecter. Cette valeur ne pourra être supérieure à la moitié de la valeur de la fréquence à détecter. Si la valeur fournie est invalide, ou aucune valeur n'est fournie, elle sera redemandée lors de l'étape du traitement. Ex: -T 50

-N: Suivi d'un nombre entier. Définit le paramètre de NFFT du calcul de la FFT. Cette valeur devrait être une puissance de 2 supérieure à 256. Si la valeur fournie est invalide, ou aucune valeur n'est fournie, la valeur de 512 sera prise par défaut. Ex: -N 512

Résultats:

-> Fichiers résultats :

Le traitement produira en résultat des fichiers WAVE, portant le nom du fichier d'entrée avec un « _# » où « # » est le numéro de l'évènement extrait. Ils contiendront les « signaux intéressants » extraits sur la base des différents critères définis.

-> Fichier Log :

Un fichier log, portant le nom du fichier d'entrée et une extension.txt, reprenant une synthèse des résultats (temps de l'évènement, puissance, fréquence, fichier résultant) sera écrit dans le dossier de destination ou le dossier précisé en paramètres (argument -l).