

Compte rendu séance 3 :

Il y a eu une grosse mise à jour du projet !

Tout d'abord voici la nouvelle organisation des fichiers :

code_fred_testons	effets_bluetooth.h	fonctions_basiques.h	melodies.h	notes.h	other_left_piano.h	principal_right_piano.h
-------------------	--------------------	----------------------	------------	---------	--------------------	-------------------------

code_fred_testons.ino est le fichier principal, où tous les fichiers *.h* seront importer (petite référence à Fred Testo pour le nom d'ailleurs, nom qui n'a pas d'importance car l'essentiel du code n'est pas dans ce fichier).

effets_bluetooth.h regroupera tous les effets que l'on pourra appliquer aux notes du piano principal droite.

En effet, maintenant le projet contient deux pianos indépendants (un « principal droite » et un autre « secondaire gauche ») avec un buzzer pour chaque. Par ailleurs, l'indépendance des deux buzzers a été un problème fort contraignant, problème réglé grâce à l'importation de la bibliothèque *ToneLibrary*.

ToneLibrary

by Daniel Centore Version **1.7.1** **INSTALLED**

This is an Arduino Library to produce a tone on any Arduino pin. Produces a square-wave of the specified frequency (and 50% duty cycle) on any Arduino pin.

[More info](#)

Toutes les touches sont maintenant chacune reliées à une entrée **numérique** d'une carte Arduino **MEGA**.

Je pense que les noms des fichiers *.h* sont assez explicites par rapport à leur utilité principal, je ne détaillerai donc pas plus que les captures d'écran suivantes :

fonctions_basiques.h :

code_fred_testons	effets_bluetooth.h	fonctions_basiques.h	melodies.h	notes.h	other_left_piano.h	principal_right_piano.h
-------------------	--------------------	----------------------	------------	---------	--------------------	-------------------------

```

/*****
*      Voici toutes les fonctions qui permettront le bon fonctionnement du variophuino.
*      Tous les effets non essentiels seront traités ailleurs.
*****/

```

melodies.h :

code_fred_testons	effets_bluetooth.h	fonctions_basiques.h	melodies.h	notes.h	other_left_piano.h	principal_right_piano.h
-------------------	--------------------	----------------------	------------	---------	--------------------	-------------------------

```
/*
 *      On retrouve ici des séquences de notes
 *      déjà enregistrées et donc jouables.
 */
```

notes.h :

code_fred_testons	effets_bluetooth.h	fonctions_basiques.h	melodies.h	notes.h	other_left_piano.h	principal_right_piano.h
-------------------	--------------------	----------------------	------------	---------	--------------------	-------------------------

```
/*
 * Absolument toutes les notes d'un piano classique !
 */

#define NOTE_A0 27
#define NOTE_AS0 29
#define NOTE_B0 31

/*-----*/

/*      début octave n°1      */
#define NOTE_C1 33 //première note jouable -> Do0
#define NOTE_CS1 35

      ●
      ●
      ●

/*
 *      Définissons maintenant le reste des variables
 *      et trions ce qui a été fait ci-dessus.
 */

int notes[85] = {
    NOTE_C1,
      ●
      ●
      ●
```

other_left_piano.h :

code_fred_testons	effets_bluetooth.h	fonctions_basiques.h	melodies.h	notes.h	other_left_piano.h	principal_right_piano.h
-------------------	--------------------	----------------------	------------	---------	--------------------	-------------------------

```
/*
 *      On retrouve ici tout ce qui concerne en
 *      particulier le second piano (à gauche).
 */
```

principal_right_piano.h :

code_fred_testons	effets_bluetooth.h	fonctions_basiques.h	melodies.h	notes.h	other_left_piano.h	principal_right_piano.h
-------------------	--------------------	----------------------	------------	---------	--------------------	-------------------------

```
/*
 *      On retrouve ici tout ce qui concerne en
 *      particulier le piano principal (à droite).
 */
```

Voici le code de *code_fred_testons.ino* au passage :

```
#include <Tone.h> //Aka "the library that saved us"

#include "notes.h"
#include "fonctions_basiques.h"

#include "principal_right_piano.h"
#include "other_left_piano.h"

#include "melodies.h"

void setup() {
  first_initializing();
  left_piano_initializing();
  right_piano_initializing();
  cch_short(left_tone);
}

void loop() {
  commands(Serial.read());
  actual_frequencies_update();
  left_piano_checking();
  //right_piano_checking();
}
```

cch_short() est la fonction permettant de faire quelques sons quand la carte Arduino démarre.

left_piano_checking() et *right_piano_checking()* ne sont pas faites pour fonctionner simultanément, une fonction sera créée ultérieurement pour cela.

effets_bluetooth.h est importé dans *fonctions_basiques.h* (car pratique).