



Carte Application Shield

TP N°2

Le buzzer

Table des matières

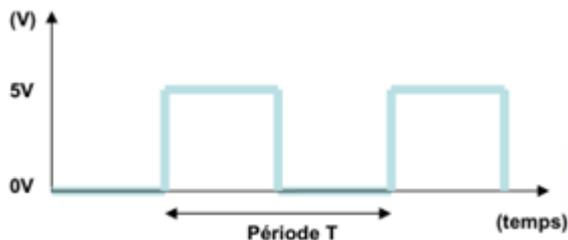
I. Le buzzer	1
a) Utilisation avec un microcontrôleur	1
b) Travail à effectuer	1
c) Mon ami Pierrot	2
d) Avec un menu	3

I. Le buzzer

a) Utilisation avec un microcontrôleur

Le buzzer se câble sur une sortie numérique en PWM (ou MLI en français) et le microcontrôleur lui envoie alors un signal périodique dont on fait varier la fréquence en fonction de la note que l'on désire jouer.

$$f = 1/T ; T=1/f ; f : \text{fréquence} ; T : \text{période}$$



Exemple : le LA est un signal d'une fréquence f de 440 Hertz soit un signal qui varie 440 fois par seconde. La fréquence du DO est 262 Hz (octave 3) etc .. (<http://jeanjacques.dialo.free.fr/frequenc.htm>)

La carte mbed possède un buzzer connecté sur une sortie PWM dont la classe est *PwmOut* :

Les principales méthodes que vous pouvez utiliser sont :

- Le constructeur de la classe *PwmOut* : `PwmOut(PinName pin)` où `pin` est la broche sur laquelle est connectée le PWM
- `void write(float value)` : définit le rapport cyclique (état haut par rapport à l'état bas) en pourcentage (1.0 représente 100%, 0.5 50% et 0.0 0% donc aucune valeur positive)
- `void period(float seconds)` : définit la période du signal PWM en seconde

De plus vous aurez besoin de la fonction `wait(int seconds)`.

b) Travail à effectuer

- Sur quelle broche est connectée le buzzer (speaker) ? regardez derrière la carte



- Déclarer un objet de classe PwmOut avec comme paramètre la broche utilisée
- Déclarer le bouton du joystick correspondant au centre
- Si la fréquence du La est de 440 Hertz, quelle est la période du signal ?
- Créez un nouveau projet vide
- Y insérez un nouveau fichier *main.cpp*
- Ecrivez un programme avec un main, une boucle while et #include "mbed.h" au début.
- Compilez, corrigez les erreurs si nécessaire (la bibliothèque mbed est manquante, appuyez sur Fix it et choisissez la toute première) .

- A l'aide d'une boucle pour, écrivez un programme qui effectue dix sonneries en LA. Un appui sur Centre permettra de recommencer. Le LA durera 400ms et le silence entre les notes 100ms.

Pour attendre l'appui sur centre : `while(!centre) {wait(0.1);}`

c) Mon ami Pierrot

Voici les fréquences de notes :

Fréquences des hauteurs (en Hertz)								
Noteloctave	0	1	2	3	4	5	6	7
Do	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01
Do#	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92
Ré	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64
Ré#	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03
Mi	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04
Fa	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65
Fa#	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91
Sol	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93
Sol#	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88
La	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00
La#	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	7458,62
Si	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13

Au clair de la lune

Partition simplifiée



Au clair de la lu - ne, mon a - mi pier - rot

- Vous travaillerez sur la gamme 3 et on choisira 400ms pour une noire (et donc 800 pour une blanche et 1200 pour une blanche pointée).
- Quelles sont les fréquences de DO, RÉ, MI ?
- Testez un programme qui joue les 6 premières notes
- Utilisez deux tableaux une pour les notes, l'autre pour la durée de chaque note. Inspirez-vous si nécessaire de https://developer.mbed.org/users/rtk/code/TP6_MarioTheme/, écrivez la totalité de la mélodie.



BTS SN – EC

Prototypage rapide avec STM32 Nucleo

TP N°2 carte « Application Shield » le buzzer



d) Avec un menu

- 🔧 Créez deux fonctions : une pour *au clair de la lune* et l'autre *Mario*
- 🔧 Créez deux menus : un qui joue aux choix l'une des deux mélodies