

Faire de la musique numérique

OBSERVATIONS

Les sons musicaux sont depuis des millénaires créés à l'aide d'instruments, qu'ils soient plutôt simples comme les percussions ou les flûtes ou beaucoup plus complexes comme les pianos ou les violons. Mais depuis plusieurs décennies, l'avancée des techniques numériques a permis l'essor de musiques dites électroniques où les sons sont produits par des hauts parleurs auxquels on envoie des signaux de caractéristiques (fréquences, amplitude, durée) contrôlées par un programme.

OBJECTIFS

- Créer un son à l'aide d'un système numérique de type Arduino.
- Respecter les notes et le rythme d'une musique avec ce système.

Matériels et produits disponibles

- Microcontrôleur Arduino
- Émetteur « grove speaker »
- Câble de connexion

ALLER À LA PAGE SUIVANTE

Faire de la musique numérique

Q1. Réaliser le protocole suivant :



Protocole 1.

- À l'aide du petit câble fourni (avec les deux extrémités blanches) connecter le module « grove speaker » (cf. image ci-dessous) au port D3 du microcontrôleur Arduino.



- Télécharger le fichier « programme_speaker.ino » (cf. site) et le mettre dans votre dossier « Documents ».
- Ouvrir ce fichier en double cliquant dessus. Il devrait s'ouvrir directement avec le bon logiciel.
- Compiler le programme en cliquant sur la touche « check » (en haut à gauche).
- Si tout est bon, relier la plaquette Arduino à l'ordinateur à l'aide du câble USB fourni.
- Téléverser (upload) le programme sur la plaquette Arduino en cliquant sur la touche « flèche » (en haut à gauche aussi).
- Une petite fenêtre s'ouvre, choisir « COM (Arduino Uno) » (En cas d'erreur, aller dans l'onglet « Outils » puis « Port » pour choisir « COM (Arduino Uno) »).
- Que se passe-t-il alors ?
- Lire le programme et essayer de déterminer quelle est la note qui est jouée et combien de temps dure le son.

Document 1.

Fréquences des notes pour différentes octaves :

Fréquences des hauteurs (en Hertz)

Note\octave	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

APPEL N°1 :

Compétences évaluées : App App

ALLER À LA PAGE SUIVANTE

Faire de la musique numérique

Q2. Modifier le programme pour que le système joue le début de l'Hymne à la joie de Beethoven (qui est aussi l'hymne européen) dont les premières notes sont :

$\text{si}_3 - \text{si}_3 - \text{do}_4 - \text{ré}_4 - \text{ré}_4 - \text{do}_4 - \text{si}_3 - \text{la}_3 - \text{sol}_3 - \text{sol}_3 - \text{la}_3 - \text{si}_3 - \text{si}_3 - \text{la}_3 - \text{la}_3$

Document 2.

Dans le programme Arduino (.ino), la fonction **tone()** permet de générer un signal carré, périodique, avec comme paramètres, la borne utilisée sur la carte, la fréquence (entre 31 Hz et 65 535 Hz pour les cartes Arduino) et la durée (en ms) :

tone(borne, fréquence, durée).

Q3. Si l'on écoute bien l'air originel, on entend que le dernier si_3 dure une 1,5 fois le temps des premières notes, l'avant dernier la_3 dure 0,5 fois le temps des premières notes et le dernier la_3 dure 2 fois le temps des premières notes. **Modifier** le programme pour qu'il prenne en compte ce rythme.

Document 3.

La fonction **tone()** est non bloquante, c'est-à-dire qu'elle n'arrête pas le programme pendant l'exécution de la fonction. Entre deux sons successifs différents, utiliser **delay()** avec en paramètre la durée (en ms) avant l'émission d'un nouveau son.

Q4. Modifier le programme pour qu'il répète ce petit air indéfiniment. Vous pourrez ajouter un délai plus long pour séparer chaque itération (répétition).

Document 4.

La fonction **void loop() { }** permet de répéter indéfiniment la partie du programme qui est entre les accolades. C'est ce qu'on appelle une boucle.

APPEL N°2 : **Compétences évaluées : Réa Réa Réa**

Q5. Chercher une partition ou le nom des notes d'un air qui vous plaît sur internet (version sans accords juste une suite de notes simples) et **modifier** le programme pour que le système puisse le jouer. Le document ci-dessous peut vous permettre de traduire certaines partitions.



APPEL N°3 : **Compétences évaluées : Réa**

NE PAS ALLER À LA PAGE SUIVANTE, IL N'Y EN A PLUS !