

Objectifs

Ce TD a les objectifs suivants :

- générer des signaux périodiques (sonores) de fréquence variable à l'aide d'un timer ;
- utilisation des fonctions natives de l'Arduino : gestion du temps, gestion des ports ;
- pilotage d'un clavier matriciel ;
- développer un système embarqué autonome.

1 Présentation du problème proposé

Un synthétiseur midi permet de jouer de la musique, à l'aide de sons synthétisés, c'est-à-dire produits sans échantillons sonores enregistrés. Dans notre cas, le morceau de musique à jouer est décrit de manière analogue au protocole *midi*, mais de manière simplifiée : il est composé d'une série de notes jouées consécutivement, caractérisée chacune par une *fréquence* (sa « hauteur »), et une *durée*. Cette série est rentrée sous forme de 2 tableaux, dans le fichier *music.ino*.

Le son sera obtenu en branchant un capteur/buzzer piezo sur l'Arduino (broches 11 et GND) : ce composant passif externe a la faculté de transformer un signal électrique en son. L'idéal est de lui appliquer un signal sinusoïdal, mais un signal carré est en pratique suffisant.

Il faudra dresser l'oreille, car le volume sonore du buzzer est extrêmement faible.

Consigne : *Testez vos fonctions aussitôt développées !*

2 Complétion du fichier *music.ino* fourni

Question 1

La fonction `initT2music` configure le timer 2 de manière à lui faire générer un signal carré (sonore) sur la broche 11 de l'Arduino.

Comme dans le TD précédent (`chrono`), il s'agit de le programmer de façon à lui faire accomplir des cycles de comptage, en permanence, de la durée/fréquence souhaitée. Il y a cependant quelques différences :

- il n'y a pas de fonction à déclencher à la fin de chaque cycle : la finalité n'est pas ici de générer des interruptions ;
- le timer doit générer un signal sur sa broche de sortie *OC2A* (c'est la broche 11, cf. brochage de l'*ATmega28*) : la finalité est en effet de produire un signal carré. Pour cela, en mode CTC, le timer peut être configuré de manière à inverser l'état de sa sortie *OC2A* à chaque fin de cycle de comptage.

Par défaut, le signal audio généré *devra avoir une fréquence d'approximativement 261.6Hz* (la note « Do »).

Pour simplifier, le prédiviseur sera *fixé une fois pour toutes*. Vous choisirez le plus petit qui permette de couvrir toutes les fréquences du morceau : grosso modo entre 250Hz et 750Hz. Plus le prédiviseur est petit, plus la résolution temporelle (= période de comptage)

est petite, et plus les approximations des périodes (et fréquences) seront faibles.

Écrire cette fonction `initT2music`, après avoir lu les ajouts de cours traitant de la génération de signaux carrés par le timer 0 (cliquez [ici](#)), qui sont aussi valables pour le timer 2, et en vous inspirant de la fonction `init_T2` du TD précédent.

En particulier, il ne faut pas oublier :

- d'activer le mécanisme de génération du signal carré du mode CTC, en spécifiant les 2 bits de poids forts du registre `TCCR2A` (cf. description du registre correspondant dans le timer 0 : `TCCR0A`).
- de bien spécifier la direction de la broche utilisée (la 11), avec `pinMode()` ou `DDRB` : sans cela, le signal sera produit, mais ne sortira pas du microcontrôleur ;
- de bien désactiver l'interruption de comparaison A du timer 2.

Approchez votre oreille du buzzer pour espérer entendre le « Do » (ou appliquez-le sur un support capable d'entrer en vibration avec lui).

Cette question ne concerne que cette 1ère fonction : ãne modifiez pas les autres.

Question 2

Complétez la fonction `joueNote`. C'est cette fonction qui détermine la hauteur (fréquence) de la note jouée par le timer 2. Vous pouvez jouer une note quelconque du morceau pour vérifier le changement de hauteur.

Question 3

Complétez la fonction `silence`. Il suffit pour cela de jouer une note de la fréquence la plus grande possible (à prédiviseur constant). Dans notre cas, elle ne sera pas entendue, car le buzzer piezo n'a pas la capacité de vibrer à une fréquence aussi extrême.

Question 4

Complétez la fonction `resetMorceau`. Elle permet de relancer le morceau au début.

3 Étude du fichier *clavier.ino* fourni

Question 5

Qu'est-ce qu'une variable `static` en C/C++ ?

Question 6

Étudiez la fonction `litTouche` : qu'apporte-t'elle par rapport à `detecteTouche` ?

4 Étude et complétion du fichier principal, également fourni

Question 7

Étudiez le code de la fonction `loop`. En particulier :

- Que fait-elle ? (les questions précédentes sont censées avoir été traitées avec succès).
- Quel est le rôle du `if` ?
- Y a-t-il des interruptions ?

Question 8

Sur le modèle de la procédure d'enchaînement des notes du morceau de la fonction `Loop`, mettre en place un mécanisme d'interrogation du clavier matriciel *toutes les 50ms*.

À des fins de tests, vous afficherez la touche nouvellement pressée à chaque détection.

Question 9

Mettez en place le « reset » du morceau, sur appui de la touche **1**.

Question 10

Gérez la lecture/pause de la mélodie : c'est la même touche du clavier matriciel qui sera utilisée pour cela : la touche **2**. Suite à la pause, le buzzer devra rester silencieux, jusqu'au prochain ordre de lecture.

5 Questions subsidiaires pour les plus avancés

Question 11

Proposez la possibilité d'un changement de la vitesse d'exécution du morceau, sur appui de la touche **3**. Le tempo devra être spécifié sur 3 chiffres, exprimant un « *nombre de notes noires par minute* ». Un tempo de 60 correspond ainsi à 60 notes « *noires* » (= de durée 1) par minute.

Question 12

Mettez en place un mécanisme permettant d'entendre « *l'attaque* » des notes. En effet, en l'état actuel, 2 notes consécutives de même fréquence ne peuvent pas être discernées à l'oreille, car elles sont jouées immédiatement l'une après l'autre, sans même une pause de durée minimale : *l'attaque* n'est pas perceptible, le changement de note ne s'accompagnant pas d'un accroissement du volume sonore.

Vous intercalerez donc un silence de durée `DUREE_SILENCE_FIN_NOTE_MS` (constante égale à 50ms) à la fin de chaque note (même en cas de changement de fréquence), et vous déduirez cette durée de celle de la note (de manière à ne - quasiment - pas modifier la durée totale du morceau).

Question 13

Proposez un ou plusieurs autres morceaux, qui pourront être sélectionnés au clavier matriciel (commande **4**).