

Arduino UNO avec l'environnement logiciel TINKERCAD

Programme d'acquisition d'un signal audio sur une des entrées analogiques avec reconnaissance des notes par FFT.

Organisation :

Vous allez travailler en binôme. Chaque binôme doit être constitué par des étudiants appartenant à **un même groupe de TP**.

Vous utiliserez l'environnement TINKERCAD où vous trouverez facilement des tutoriels sur le net.

Questionnement :

Certaines questions peuvent être faites indépendamment des autres...

1. Indiquer les fréquences des touches d'un piano.
2. Préciser notamment celle du LA de l'octave qui se situe au milieu du clavier (octave numérotée 3 en général).

3. Acquisition d'un signal sinusoïdal sur une entrée analogique de la carte Arduino UNO :

- a. Créer un nouveau Circuit sous TINKERCAD et insérer une carte Arduino UNO.
- b. Repérer les entrées analogiques sur la carte Arduino UNO.
- c. Indiquer la plage de tensions admissibles en entrée (cf. R2.06).
- d. Indiquer sur combien de bits le CAN travaille (cf. R2.06).
- e. Récupérer le programme "Acquisition_v1.ino" dans Moodle ou ci-dessous et **comprenez** ce qu'il fait :

```
#define CHANNEL A0
const double samplingFrequency = 2000; //Hz, must be less than 10000 due to ADC
```

```
unsigned int sampling_period_us;
unsigned long microseconds;
double valeur;
```

```
void setup()
{
  sampling_period_us = round(1000000*(1.0/samplingFrequency));
  Serial.begin(115200);
}
```

```
void loop()
{
  microseconds = micros();
  valeur = analogRead(CHANNEL);
  Serial.println(valeur);
  while(micros() < (microseconds + sampling_period_us)){
    //empty loop
  }
}
```

- f. Ajouter dans le schéma un GBF et un oscilloscope.
- g. Régler le GBF pour obtenir un signal sinusoïdal de fréquence **20 Hz**, d'amplitude allant de **1V à 4V**.
- h. Vérifier avec l'oscilloscope.
- i. Raccordez le GBF sur l'entrée analogique A0 de la carte Arduino UNO.
- j. Utiliser le traceur série (rubrique Code) pour visualiser la valeur numérique du signal analogique en entrée A0. Conclure.
- k. Utiliser le moniteur série et trouver les valeurs numériques min et max (faites des photos en réduisant la période...) ou utiliser un générateur de tension continue à la place du GBF 😊.
- l. Vérifier que ces valeurs numériques sont en adéquation avec les valeurs analogiques du signal sinusoïdal.
- m. Expliquer, côté microcontrôleur, l'impact du choix de la valeur de « samplingFrequency = 2000 ».

- n. Avons-nous vu une autre façon de faire l'échantillonnage (cf. R2.06) ? **Argumenter.**

4. Bibliothèque « arduinoFFT » avec carte Arduino UNO:

- a. Créer un nouveau Circuit sous TINKERCAD.
- b. Installer la bibliothèque FFT de Enrique Condes **version 1.6.2** depuis l'environnement IDE d'Arduino. Vous la trouverez aussi par ce lien :

<https://github.com/kosme/arduinoFFT/tree/v1.6.2>

Sous TINKERCAD, cette librairie n'est pas disponible... Trouvez alors sur le net comment pouvoir l'exploiter 😊

- c. Intéressez-vous aux programmes d'exemple « fft_xx.ino » et au site suivant :
<https://www.norwegiancreations.com/2017/08/what-is-fft-and-how-can-you-implement-it-on-an-arduino/>
- d. Récupérer celui qui pourrait reconnaître la fréquence d'une touche de piano (le LA par exemple) et modifier-le pour qu'il en soit ainsi.
- e. Vérifier en injectant le signal du GBF - **sinusoïde à 440 Hz d'amplitude crête à crête (cc) de 3V et de tension continue de 1.5V** - et répondre aux questions suivantes :
- Combien d'échantillons sont utilisés pour calculer la FFT ?
 - Quelle est la fréquence pas déterminée par le programme ? Vérifier par la formule.
 - En déduire ou donner toutes les fréquences prises en compte par la FFT ? Préciser la valeur minimale et maximale.
 - Pourquoi n'avons-nous pas la valeur exacte de 440 Hz affichée sur la console série ?
 - Proposer d'autres « réglages » au niveau du programme pour améliorer la détection de la fréquence du LA. Argumenter.
 - Noter dans le tableau de la question 6.b. les fréquences détectées pour chacune des notes.

5. Afin de pouvoir faire correctement les parties suivantes, posez-vous la question suivante :
 Quel est l'intérêt de la FFT (Fast Fourier Transform) dans ce projet ?
Indication : garder à l'esprit ce qu'on veut en faire et comment l'exploiter (nombre de points pour le calcul de la FFT, lien avec la fréquence d'échantillonnage, etc.).

6. Reconnaître les notes par une LED :

- a. Continuer avec le même circuit de la question 4 en prenant soin de faire une copie !
- b. Brancher des LEDs et résistances sur les sorties numériques 2 à 10. Elles représenteront les touches d'un clavier de piano. Voici le tableau de correspondance :

Touche	Fréquence	Fréquence détectée	Broche
Ré2	147		2
Mi2	165		3
Fa2	175		4
Sol2	196		5
La2	220		6
Si2	247		7
Do3	262		8
Ré3	294		9
Mi3	330		10

- c. Modifier le programme précédent de la question 4 afin que les LEDs s'allument selon la note jouée en vérifiant avec le GBF.

Validation Intermédiaire [VI3.1]

7. Jouer un morceau de musique avec une carte Arduino UNO :

- a. Récupérer le circuit sur le lien suivant :

<https://www.tinkercad.com/things/bXuK4iUc6im>

- b. Que permet de faire ce montage R-2R ? Donner ses caractéristiques principales.

- c. A l'aide du programme incomplet ci-dessous, modifier-le afin qu'il permette de générer/jouer la note LA (rôle de la fréquence f_1):

```
#define pi 3.14159265359
#define PIN_ADC A2
char start_analog = 0;
float fe, te;
float a11, a21, b01, f1, w1, Ap1;
float y1[3];

// Fonction de traitement IT n°12 = Timer 1 CTC
ISR(TIMER1_COMPA_vect){
  start_analog=1;
}

void setup(){
  // Echantillonnage
  fe = ???;
  te = 1/fe;
  // Sinus 1
  Ap1 = 1;
  f1 = 440;
  w1 = 2*pi*f1/fe;
  a11 = 2*cos(w1);
  a21 = 1;
  b01 = Ap1*sin(w1);
  y1[2] = 0;
  y1[1] = b01;

  // Configuration Timer 1
  TCCR1A=0;
  TCCR1B=B1010; // Mode CTC (Clear Timer On Compare) et Prescaler 8 (Clock/8)
  OCR1A= ???; // Registre de comparaison
  TIMSK1=B010; // IT Timer1 Quand TCNT1=OCR1A
  DDRD = B11111111; //PortB en sortie !
  sei(); // activation des IT (SREG.7=1)
}

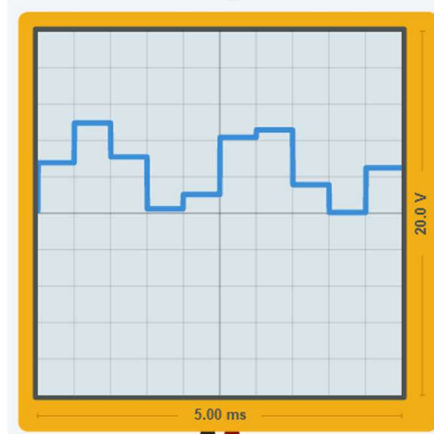
void loop() {
  if (start_analog)
  {
    //Equation aux différences d'un sinus (cf. R2.13 ☺)
    y1[0]=a11*y1[1]-a21*y1[2];
    //Décalage des échantillons retardés
    y1[2] = ???;
    y1[1] = ???;

    PORTD = ???;

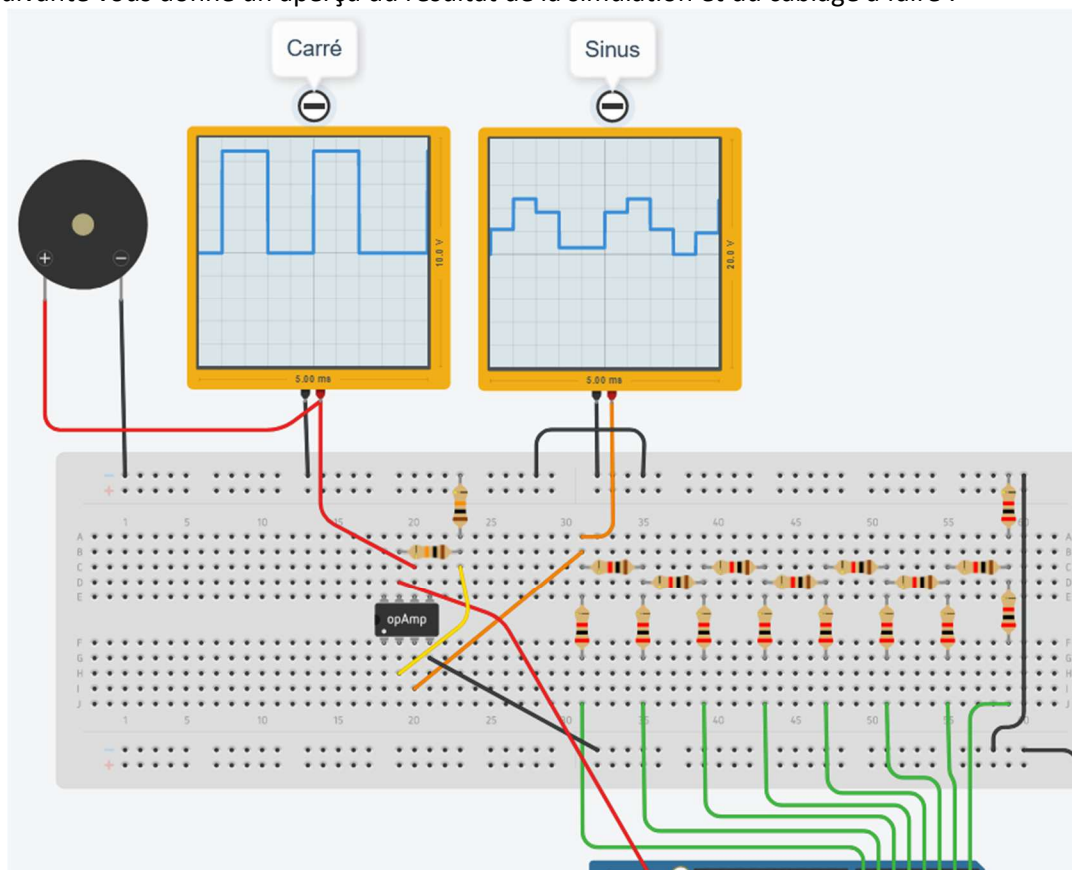
    start_analog=0;
  }
}
```

Indications :

- Choisir f_e en fonction des fréquences des notes de la partition à jouer (cf. fin du sujet) et en lien avec la question 3.m.
- Calculer OCRA1 comme vu en R2.06 avec f_e choisi précédemment. Rien ne vous empêche de faire le calcul dans le programme...
- PORTD doit fournir une valeur entière sur 8 bits à partir de la variable $y1[0]$ qui est déclarée en type « float ».
- Voici ce que vous pouvez obtenir en sortie du CNA avec $f_1=440$ Hz et $f_e= ???$ Hz [Astuce ☺ : le tracé temporel vous permet de trouver T_e , donc f_e] :

**Validation Intermédiaire [VI1.1]**

- d. Si vous souhaitez entendre la note, il suffirait de rajouter un amplificateur avec un HP. Tinkercad ne le propose pas... Néanmoins, vous pouvez brancher un buzzer. Comme il fonctionne avec un signal carré, il faut rajouter un comparateur à base d'un « opAmp » (Amplificateur Opérationnel) dont le basculement se fait à la tension moyenne du signal sinusoïdal généré. La capture d'écran suivante vous donne un aperçu du résultat de la simulation et du câblage à faire :



- e. Expliquer le rôle des 2 résistances de 10 K Ω et calculer le potentiel présent sur le fil jaune.
- f. Comprendre comment se fait le basculement en s'inspirant du montage « comparateur simple ».
- g. A partir de ce circuit (prenez soin d'en faire une copie !), écrire un programme qui permet de jouer un morceau de musique (cf. dernière page).

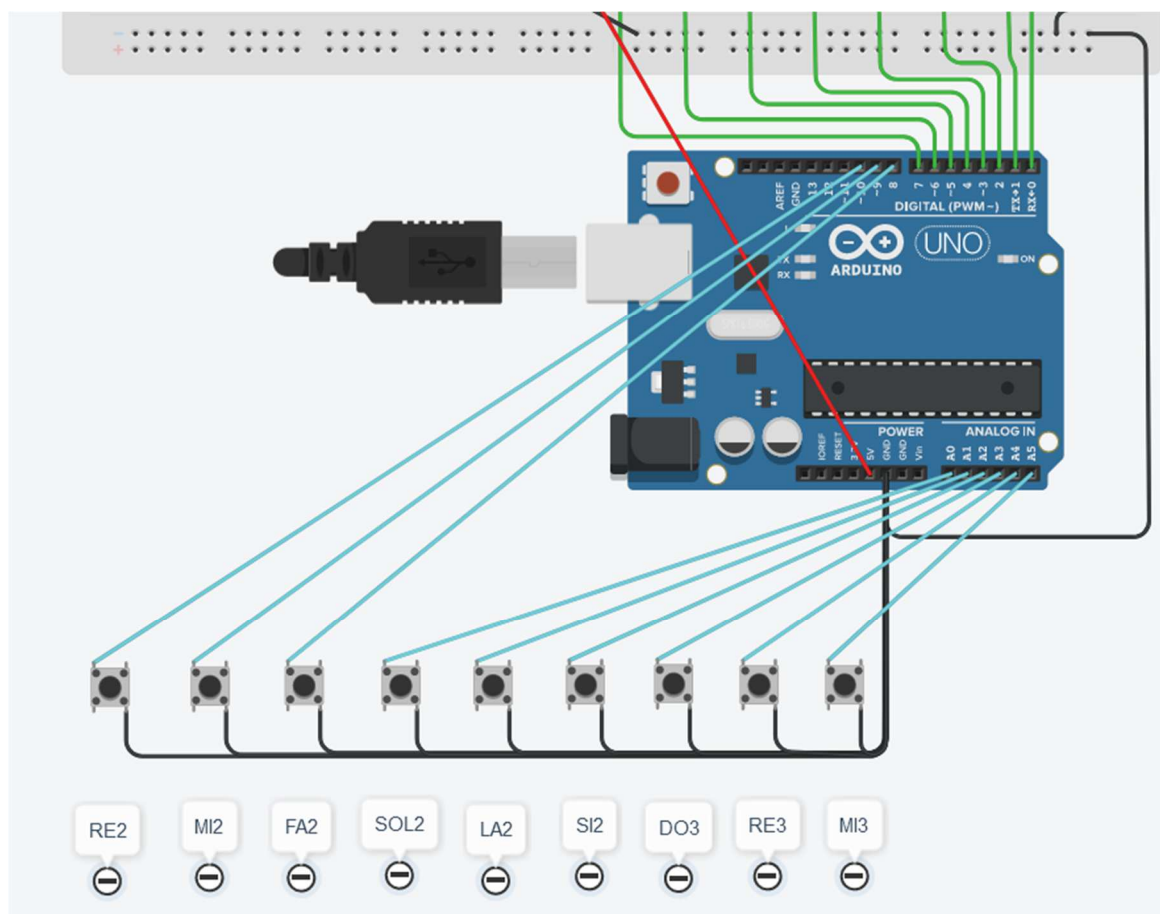
Indications :

- Il va falloir revoir sa copie concernant la mise en place de l'interruption timer par le drapeau « start_analog » pour que vous puissiez, dans le programme principal, utiliser la fonction delay() sans qu'elle soit bloquante pour l'interruption ET qu'elle permette de donner la durée de la note !
- Il serait opportun de créer une fonction qui permette de générer/initialiser les coefficients de l'équation aux différences selon la fréquence à jouer...

FAIRE VALIDER PAR L'ENSEIGNANT [V1]

8. Jouer au piano avec une carte Arduino UNO :

- a. Faire une copie du circuit de la question 7 et éditer ce circuit.
- b. Réaliser le câblage suivant où chaque Bouton Poussoir [BP] simulera une touche du piano :



- c. Modifier le programme pour que l'appui sur un BP puisse générer la note et ainsi jouer un morceau de musique.

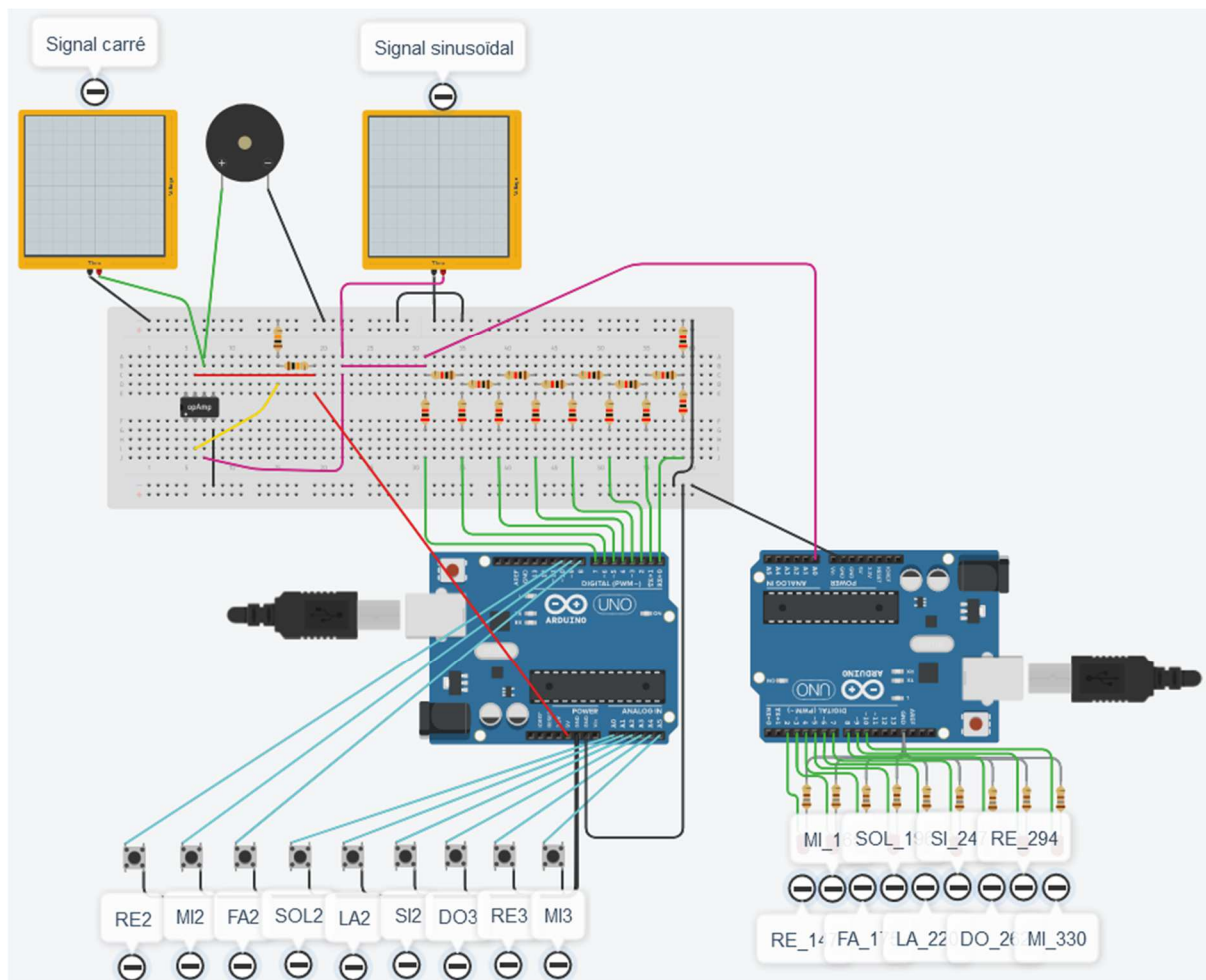
Indications :

- Chaque BP utilise la résistance interne au ATMEGA328P montée en PULL-UP.

FAIRE VALIDER PAR L'ENSEIGNANT [V2]

9. Finalité :

- Faire une copie du circuit de la question 6 et éditer ce circuit.
- Editer le circuit de la question 7 et faire un copier/coller de son schéma dans le schéma de la copie précédente. Voici ce que vous devez obtenir après avoir relié la sortie du CNA sur l'entrée analogique A0 de l'autre carte Arduino UNO :



- Valider le fonctionnement des LEDs avec selon la note jouée grâce à son BP dédié.

Indications :

- Il faudra certainement refaire les fréquences détectées du tableau 6.b. car le signal sinusoïdal n'est plus parfait comme le faisait le GBF... :

Touche	Fréquence	Fréquence détectée	Broche
Ré2	147		2
Mi2	165		3
Fa2	175		4
Sol2	196		5
La2	220		6
Si2	247		7
Do3	262		8
Ré3	294		9
Mi3	330		10

FAIRE VALIDER PAR L'ENSEIGNANT [V3]

d. Allumer la Built-in LED lorsque la **séquence** de notes suivantes est reconnue :

DO3 puis MI3 puis RE3.

FAIRE VALIDER PAR L'ENSEIGNANT [V4]

Barème pouvant être modifié : Note sur 30 répartie comme suit :

V1	V2	V3	V4
10 points dont 5 pour VI1.1	5 points	10 points dont 5 pour VI3.1	5 points

1
Au clair de la lu - ne, 1 3 mon a - mi Pier-rot. 1
Prê-te-moi ta plu - me,

4
2 3 2 1 5
pour é - crire un mot. Ma chan-delle est mor - te, je n'ai plus de feu.

7
1 1 3
Ou - vre moi ta por - te, pour l'a - mour de Dieu.