





EQUIPE :	Date :
PROJET 3 : CLAVIER MUSICAL	

**AVEC QUELQUES RESISTANCES ET DES BOUTONS, VOUS ALLEZ CONSTRUIRE UN PETIT CLAVIER MUSICAL. LE BUT EST DE JOUER LES PREMIERES NOTES DE LA COMPTINE « AU CLAIR DE LA LUNE »**

Objectif : découvrir les échelles de résistances, les tableaux.

Durée : 1h30 heures


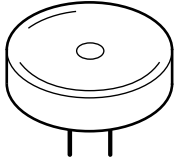



Difficulté : ■■■■□

				Score
++	++	++	++	
+	+	+	+	
-	-	-	-	
--	--	--	--	

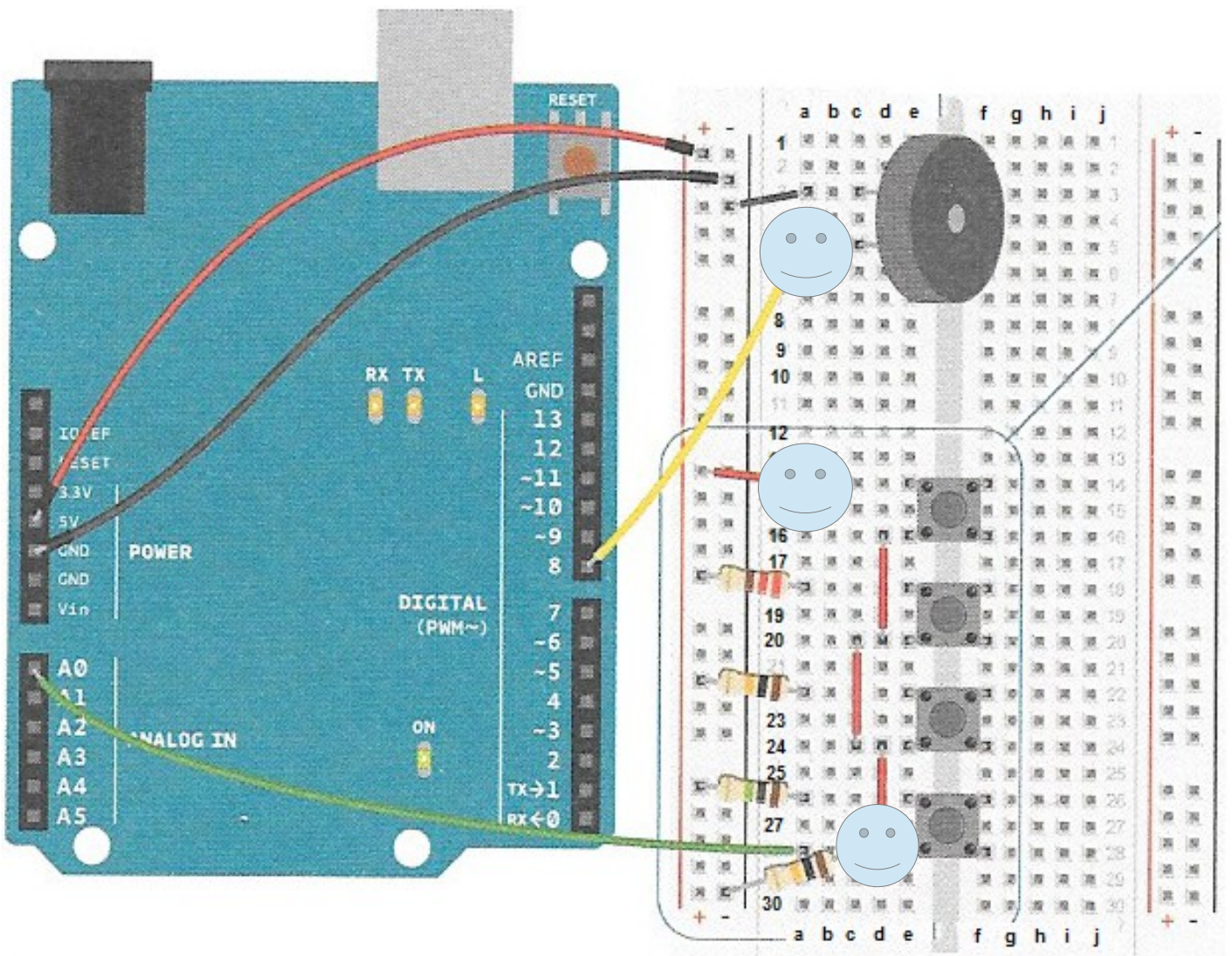
*Il est possible de simplement brancher un certain nombre de boutons poussoirs à des entrées numériques pour déclencher différents sons, mais dans ce projet vous allez construire quelque chose que l'on appelle une échelle de résistances.*

C'est une manière de lire une certaine quantité d'interrupteurs en utilisant une entrée analogique : une technique utile si vous manquez d'entrées numériques. Vous brancherez quelques boutons poussoir connectés en parallèle à l'alimentation via une résistance. En appuyant sur un bouton, une tension spécifique sera envoyée à la broche d'entrée. Si vous pressez deux boutons à la fois, vous obtiendrez un résultat unique basé sur la relation entre les deux résistances en parallèle.

## LES COMPOSANTS

				
BOUTONS POUSSOIR	PIEZO	RESISTANCE 10 kΩ	RESISTANCE 1 MΩ	RESISTANCE 220 Ω
x4	x1	x2	x1	x1

## LE MONTAGE



# LE CODE

## Le tableau

Vous aurez besoin de stocker une liste des fréquences que vous jouerez lorsque vous appuierez sur les boutons. Vous pouvez commencer avec les fréquences des do, ré, mi et fa médians (262 Hz, 294 Hz, 330 Hz et 394 Hz). Pour ce faire, vous aurez besoin d'un nouveau type de variable appelé tableau.

Un tableau est un moyen de stocker différentes valeurs ayant un lien entre elles, comme des notes sur une échelle musicale, en utilisant un seul nom. C'est un outil pratique pour accéder rapidement à des informations. Pour déclarer un tableau, faites comme avec une variable, en ajoutant une paire de crochets après le nom : []. Après le signe égal, vous inscrirez les éléments du tableau entre accolades.

Pour lire ou modifier un élément dans le tableau, vous vous adressez à eux en utilisant le nom du tableau et l'index de l'élément auquel vous voulez accéder. L'index se base sur l'ordre dans lequel l'élément apparaît lorsque le tableau est créé. Le premier élément est l'élément 0, le second est l'élément 1, etc.

## Créer un tableau de fréquences

Initialisez un tableau de quatre notes en utilisant les fréquences listées ci-dessus. Faites de ce tableau une variable globale en la déclarant avant le `setup()`.

```
int buttons[6];  
// initialisez un tableau avec 6 entiers  
  
int buttons[0] = 2;  
// donnez au premier élément du tableau la valeur 2
```

## Commencez la communication en série

Dans votre `setup()`, ouvrez la liaison série avec l'ordinateur.

```
1 int notes[] = {262, 294};  
  
2 void setup() {  
3   Serial.begin(9600);  
4 }
```

## Lisez la valeur analogique et envoyez-la au moniteur

Dans la `loop()`, déclarez une variable locale pour stocker la valeur lue sur la broche A0. Puisque chaque bouton est connecté à l'alimentation via une résistance différente, chacun sera associé à une valeur différente. Pour voir les valeurs dans le moniteur série, ajoutez la ligne `Serial.println(keyVal)`.

```
5 void loop() {  
6   int keyVal = analogRead(A0);  
7   Serial.println(keyVal);  
  
8   if(keyVal == 1023){  
9     tone(8, notes[0]);  
10  }
```

## Utilisez une instruction if() ... else pour déterminer quelle note jouer

En utilisant une instruction `if() ... else`, vous pouvez assigner à chaque valeur une note différente. Les valeurs du programme d'exemple sont les valeurs théoriques correspondant aux résistances utilisées. Comme toutes les résistances ont une tolérance, il se peut qu'elles ne marchent pas parfaitement pour vous. Utilisez les informations du moniteur série pour les ajuster si nécessaire.

```
11  else if(keyVal >= 990 && keyVal <= 1010){  
12    tone(8, notes[1]);  
13  }  
14  else if(keyVal >= 505 && keyVal <= 515){  
15    tone(8, notes[2]);  
16  }  
17  else if(keyVal >= 5 && keyVal <= 10){  
18    tone(8, notes[3]);  
19  }
```

## Jouez les notes correspondant à la valeur analogique

Appelez la fonction `tone()` après chaque instruction `if()`. Le programme cherchera dans le tableau la valeur à jouer. Vous pouvez dire à l'Arduino de jouer un son différent si la valeur lue sur l'entrée A0 ne correspond à aucune de vos instructions `if`. Il est possible que votre circuit produise des parasites et que les valeurs fluctuent un peu lorsque vous pressez les boutons. Pour ne pas être gêné par ces variations vérifiez une petite gamme de valeurs comme ci-contre. Utilisez le comparateur "`&&`" pour tester plusieurs conditions à la fois.

Si vous pressez le premier bouton, `notes[0]` sera joué. S'il s'agit du second, `notes[1]` sera joué et pour le troisième ce sera `notes[2]`. C'est ici que les tableaux deviennent intéressants.

```
20  {  
21    noTone(8);  
22  }  
23 }
```

## Arrêtez de jouer des sons lorsqu'on appuie sur rien

On ne peut jouer qu'une seule note sur une seule broche à la fois, donc si vous appuyez sur plusieurs touches vous n'entendrez qu'une note.

Pour arrêter de jouer des sons quand aucune bouton n'est pressé, appelez la fonction `noTone()`, en lui indiquant sur quelle broche arrêter de jouer.