

Utilisation des microcontrôleurs en Physique-Chimie au lycée



## Initiation à mBlock 3 avec une carte UNO



L'Environnement de Développement Intégré (EDI) **mBlock** est un logiciel qui regroupe tous les outils nécessaires pour programmer la carte UNO et communiquer avec elle.

Il possède un éditeur à blocs de type "Scratch" (logiciel de programmation utilisé au collège en mathématiques) ; son utilisation au collège est courante en technologie et se développe en Physique-Chimie. Il est donc particulièrement adapté aux élèves de seconde (en attendant qu'ils soient aguerris avec les langages de programmation).

Le protocole est proposé ici avec la version 3 de mBlock (à adapter pour la version 5)

### Exemple : produire un son avec un buzzer passif (ou un HP)

#### Objectif et outils :

Il s'agit de produire un son d'une certaine fréquence à l'aide d'une carte type UNO et de mBlock 3.

#### Lien avec le programme de physique-chimie du lycée :

NIVEAU	Seconde générale
Partie du programme officiel	Ondes et signaux
Capacité expérimentale ou numérique	Utiliser un dispositif comportant un microcontrôleur pour produire un signal sonore.

**Matériel :** carte type UNO + buzzer passif (seul ou module) + 2 fils M/F ou 1 cordon Grove-4F + mBlock

Les sorties numériques de la carte UNO peuvent monter à 5V mais ne supportent **pas plus de 40mA**. Conséquences :



- Il ne faut pas brancher un haut-parleur de moins de 150Ω directement sur une sortie numérique.
- L'intensité nominale d'un buzzer allant usuellement de 10 à 50mA selon le modèle, il faut éviter de brancher directement sur une sortie numérique un buzzer de forte intensité nominale, ou il faut programmer des émissions sonores très brèves.

### 1) Préparation du montage sur carte type UNO



D9 GND

L'USB de la carte UNO étant **NON connecté**, connecter les deux bornes du buzzer ou du haut-parleur aux ports numérique 9 et GND de la carte UNO : - soit avec deux fils Dupont M/F

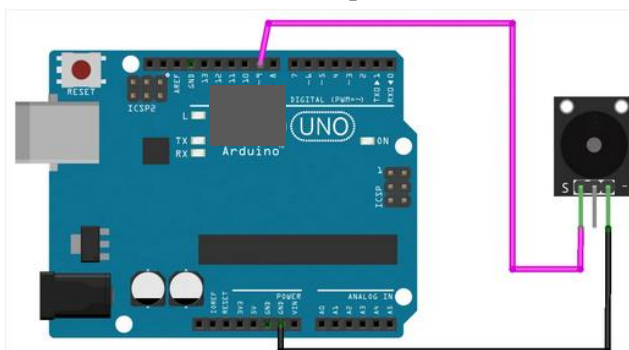
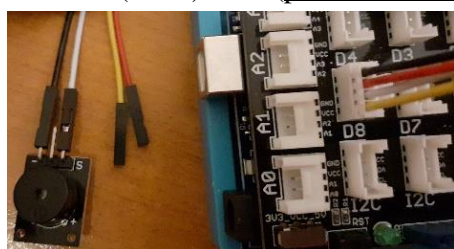
- soit avec un cordon Grove-4F sur le port D8 du shield Grove et relié aux broches du buzzer par ses fils blanc (D9) et noir (GND)

Illustration du câblage avec un module buzzer passif : (tribroche mais la broche du milieu ne sert pas)

la broche S du module est connectée au port D9 de la carte UNO, et la broche – à un port GND,

- soit avec deux fils Dupont M/F (schéma ci-contre),

- soit avec un câble Grove sur le port D8 de l'interface Grove et relié au buzzer par ses fils blanc (D9) sur S et noir (GND) sur – (photo ci-dessous)



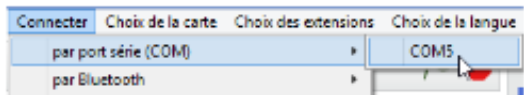
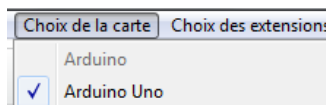
**Précaution d'usage : avant de connecter la carte UNO pour la programmer, déconnecter la broche D8 (ou fil blanc du câble Grove). Ne reconnecter qu'après avoir téléversé le programme (cf. ci-après).**

## 2) Connexion USB et préparation de la carte

- La session Windows étant ouverte, connecter la carte UNO sur le PC via le câble USB.

- Ouvrir *mBlock3*. Sélectionner la langue dans le menu *Choix de la langue*

- Dans le menu *Choix de la carte*, sélectionner *Arduino Uno*



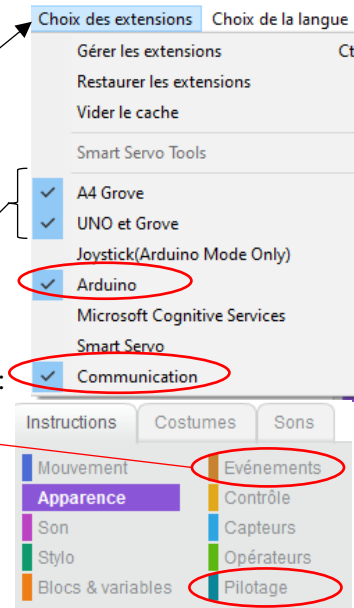
- Dans le menu *Connecter*, sélectionner le *port COM* qui apparaît

En cas de doute, regarder sur quel port COM est branchée la carte UNO dans le menu *périphériques* des *paramètres* de windows.

## 3) Réalisation du code avec mBlock et pilotage en mode direct :

- Dans le menu *Choix des extensions*, sélectionner *Arduino* pour pouvoir programmer la carte UNO et *Communication* pour permettre son pilotage en mode direct :

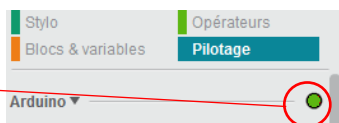
Remarque : les extensions *A4 Grove* et/ou *UNO et Grove* ne sont pas utiles ici (elles sont nécessaires pour les modules Grove, ce qui n'est pas le cas ici). Elles peuvent cependant être sélectionnées si elles sont installées.



- Construire l'algorithme dans le cadre de droite en déposant les blocs du cadre du milieu :

1) commencer par l'instruction  (dans l'onglet *Evénements*)

2) les instructions de pilotage Arduino se trouvent dans l'onglet *Pilotage* (le voyant doit être vert) :

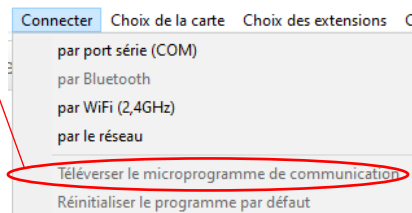


Voici par exemple l'algorithme à blocs pour produire un Do3 pendant "1/4 temps" avec le buzzer passif connecté à la sortie numérique 9 :



- Dans le menu *Connecter*, cliquer sur *Téléverser le microprogramme de communication*



- Cliquer sur le drapeau vert pour lancer le programme et sur l'octogone rouge pour le stopper.



### Correspondance des notes anglo-saxonnes pour la 3<sup>e</sup> octave par exemple :

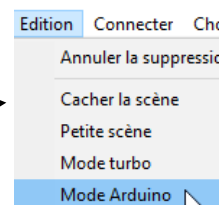
Note (notation française)	Do3	Re3	Mi3	Fa3	Sol3	La3	Si3
Note (notation anglaise)	C4	D4	E4	F4	G4	A4	B4
Fréquence (Hz)	262	294	330	349	392	440	494

## 4) Visualisation du code en langage Arduino

mBlock permet de visualiser le code dans un éditeur de texte : remplacer  par  (onglet *Pilotage*) comme ci-dessous :



puis sélectionner le *Mode Arduino* dans le menu *Edition* :



Cela ouvre l'éditeur de texte suivant :

```

Retour Téléverser dans l'Arduino Ouvrir dans l'EDI Arduino
1
2 double angle_rad = PI/180.0;
3 double angle_deg = 180.0/PI;
4
5 void setup() {
6   pinMode(9,OUTPUT);
7   tone(9,262,250); // write to buzzer
8   delay(250);
9 }
10
11 void loop() {
12   _loop();
13 }
14
15 void _delay(float seconds){
16   long endTime = millis() + seconds * 1000;
17   while(millis() < endTime)_loop();
18 }
19
20 void _loop() {
21 }
22
23
24 }

```

Ce peut être intéressant pour commencer à familiariser les élèves avec le langage Arduino, mais aussi et surtout dans le cas présent pour montrer que chaque note est caractérisée par une fréquence :

on pourra demander à l'élève de trouver l'instruction qui pilote le buzzer (*facile : il y a le commentaire !*),

les textes situés après // sont des commentaires destinés à la compréhension du code

puis de proposer une signification des 3 nombres en argument (en lui donnant le tableau de correspondance note-fréquence), ou de donner la fréquence du son (en lui donnant les significations des 3 arguments de la fonction `tone()`).

Remarques :

- mBlock génère par défaut des lignes de code qui ne servent pas systématiquement (lignes 5-6 par exemple).
- l'emploi de `tone()` est déconseillé avec les ports numériques 3 et 11 (d'après <https://www.arduino.cc>)

### 5) Pilotage d'objets sur la scène d'action

Comme Scratch, mBlock permet de piloter des objets sur une scène d'action.

- Cela permet de rendre le programme plus ludique.

Par exemple, on peut créer un *objet* HP avec deux *costumes* ON et OFF (en important des images) représentant l'état du buzzer :

**Exemple d'utilisation des costumes créés :**

**Aperçu à l'exécution**

- Appuyer sur la barre espace du clavier (HP OFF)
- Appuyer sur la barre espace du clavier (HP ON)

**Code**

```

quand est cliqué
  basculer sur le costume HP_OFF
  aller à x: 0 y: 0
  montrer
  répéter indéfiniment
    attendre jusqu'à touche espace pressée?
    basculer sur le costume HP_ON
    attendre 0,5 secondes
    répéter jusqu'à touche espace pressée?
    jouer un son sur la broche 9 : note C4 un quart temps
    attendre 0,2 secondes
  basculer sur le costume HP_OFF
  attendre 0,5 secondes

```

- Dans le cas d'une chaîne capteur-microcontrôleur-actionneur, cela permet aussi de simuler l'actionneur.

Par exemple, sur un montage d'alarme à thermistance, on pourra simuler le buzzer avec les deux costumes précédents, pour ne réaliser que le montage de la thermistance.

Cela est intéressant pour remplacer un circuit actionneur nécessitant du prototypage sur platine d'essais et risquant de compliquer un TP avec des élèves de seconde (par exemple un montage LED + résistance pourra être simulé par les dessins d'une lampe allumée et éteinte).

Remarque : certaines applications comme Tinkercad (application en ligne) permettent de coder avec des blocs mais aussi de représenter le montage et de simuler son fonctionnement.