

## MBot foot non joueur

# SOMMAIRE

EN QUOI CONSISTE L'ACTIVITÉ ?

p 2

MATÉRIEL / APPLICATION NÉCESSAIRE

p 3

EXPLICATION DU CODE

p 4 - 16

## EN QUOI CONSISTE L'ACTIVITÉ ?

L'ACTIVITÉ CONSISTE EN UN JEU

DE LUMIÈRE  
& DE SON



 **LE BUT :** POUR LES JEUNES QUI ATTENDENT LEUR TOUR  
POUR JOUER AVEC LES MBOTS



CRÉER AVEC UN ARDUINO DES SÉQUENCES DE SON ET DE LUMIÈRE POUR CÉLÉBRER LES BUTS.

CELA LEUR PERMETTRA "D'APPRENDRE" LES BASES EN CODAGE C/C++ COMPILE EN AVR-G++

## MATÉRIEL & APPLICATION

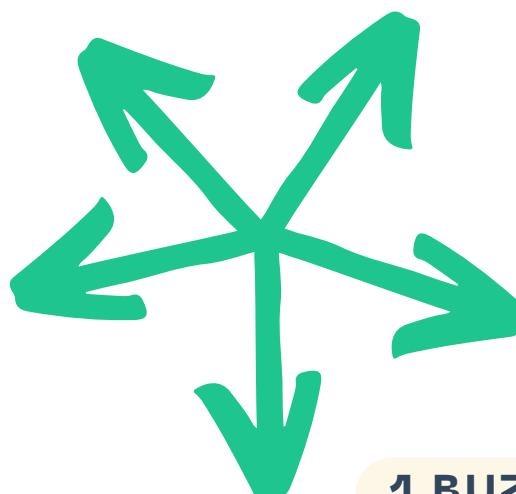
1 ARDUINO UNO



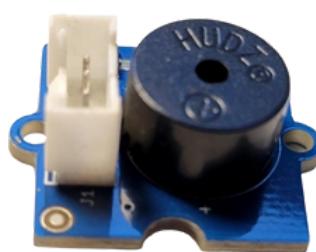
1 BOUTON POUSSOIR



3 LED ROUGES



1 BUZZER PASSIF



ARDUINO IDE



## TÉLÉCHARGEMENT



LE CODE EST SUR SHAREPOINT SOUS LE NOM DE  
"SPECTATEURV2NEW"  
**(LIEN DE TÉLÉCHARGEMENT CI-DESSOUS)**



[HTTPS://CAUXSEINE.SHAREPOINT.COM/SITES/SERVICEMEDIATIONNUMERIQUE/SERVICE\\_S%20CIVIQUES/FORMS/ALLITEMS.ASPX?ID=%2FSITES%2FSERVICEMEDIATIONNUMERIQUE%2FSERVICES%20CIVIQUES%2F01%2DATELIER%2FMBOT%2FFOOT&VIEWID=A9EF6687%2D49BF%2D48A3%2D98D4%2DC3890D48592A](https://cauxseine.sharepoint.com/sites/servicemediationnumerique/service_s%20civiques/forms/allitems.aspx?ID=%2fsites%2fservicemediationnumerique%2fservices%20civiques%2f01%2dateliers%2fmbot%2ffoot&viewid=A9EF6687%2D49BF%2D48A3%2D98D4%2DC3890D48592A)

IL REGROUPE 2 FICHERS DIFFÉRENTS :

- SPECTATEURV2NEW.INO
- PITCHES.H

UNE FOIS LE DOSSIER TÉLÉCHARGÉ, VOUS OBTIENDREZ UN DOSSIER ZIP PLACÉ AUTOMATIQUEMENT DANS VOS TÉLÉCHARGEMENTS.

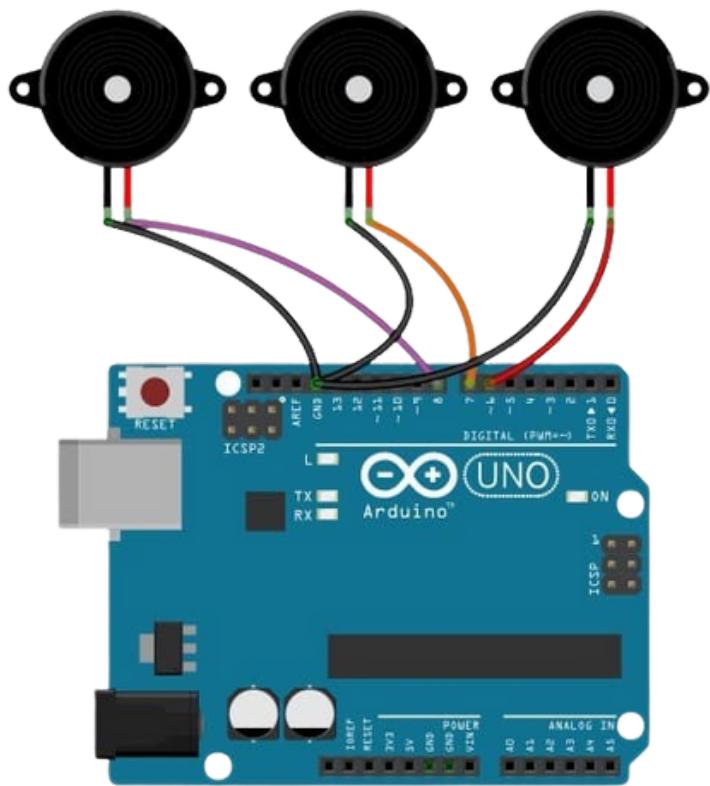
VOUS AUREZ JUSTE À FAIRE UN CLIC DROIT DESSUS, ET CLIQUEZ SUR L'OPTION "EXTRAIRE TOUT" ET CLIQUEZ ENSUITE SUR "EXTRAIRE"

LE DOSSIER OBTENU SUITE À L'EXTRACTION DOIT ÊTRE PLACÉ SUR LE BUREAU.

## GÉNÉRALITÉ

LE FICHIER SPECTATEURV2NEW EST CELUI DANS LEQUEL LES JEUNES DEVONT "CODER".

À CÔTÉ DE ÇA, PITCHES.H EST UN DOSSIER REGROUPANT LES NOTES DE MUSIQUE; SANS LUI, LE BUZZER NE POURRA PAS PRODUIRE DE SON.  
IL NE DOIT PAS ÊTRE MODIFIÉ.



## INSTALLATION ARDUINO IDE

**LES DEUX FICHIERS DEVONT ÊTRE OUVERTS À L'AIDE D'ARDUINO IDE. (LIEN DE TÉLÉCHARGEMENT CI-DESSOUS)**

[HTTPS://WWW.ARDUINO.CC/EN/SOFTWARE/](https://www.arduino.cc/en/software/)

TÉLÉCHARGEZ LA VERSION DE WINDOWS 10 EN 64 BITS (CLIQUEZ DEUX FOIS SUR LES OPTIONS "JUST DOWNLOAD")

UNE FOIS TÉLÉCHARGÉ, LANCEZ L'EXÉCUTABLE ET CLIQUEZ SUR "J'ACCEPTE" PUIS SUIVANT EN CHOISISSANT "JUSTE POUR MOI" PUIS "INSTALLER"

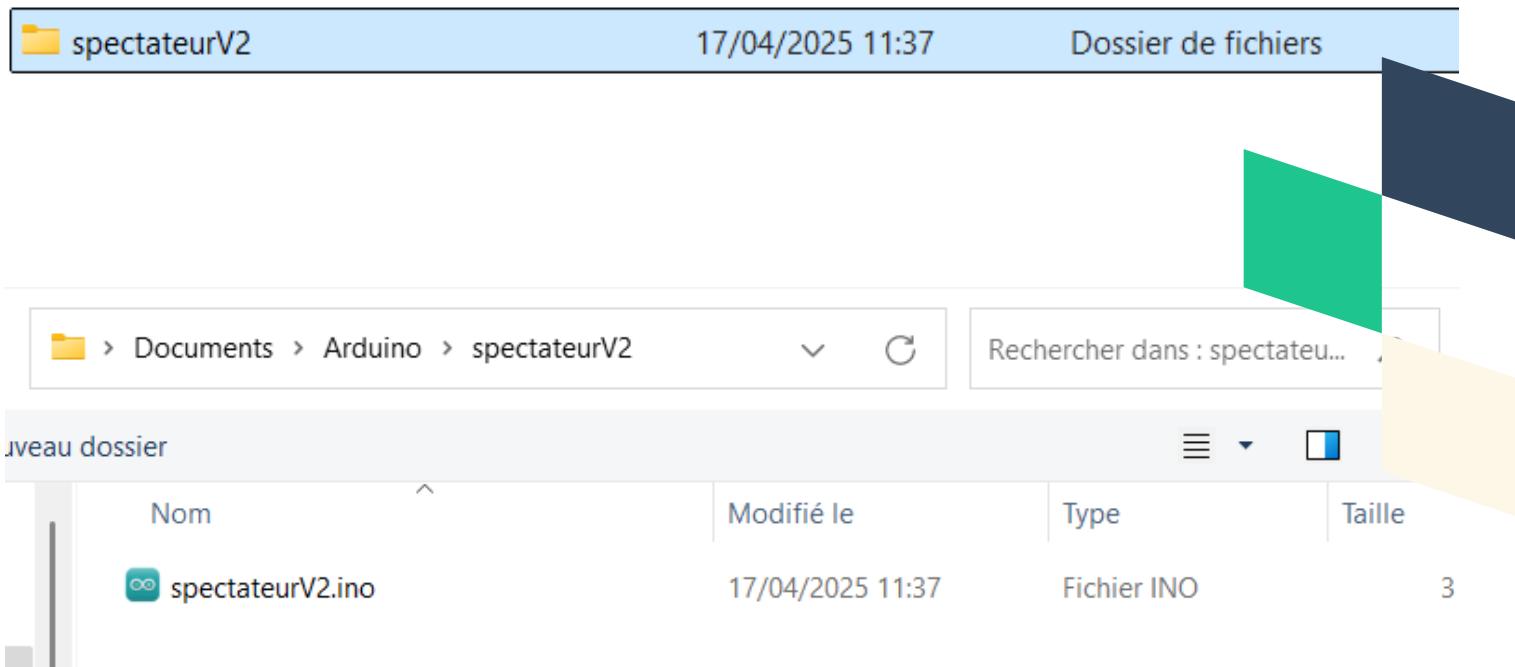


CONSEILLER  
NUMÉRIQUE

## UTILISATION ARDUINO IDE

UNE FOIS LE LOGICIEL LANCÉ, IL FAUDRA APPUYER SUR "FILE", PUIS "OPEN".

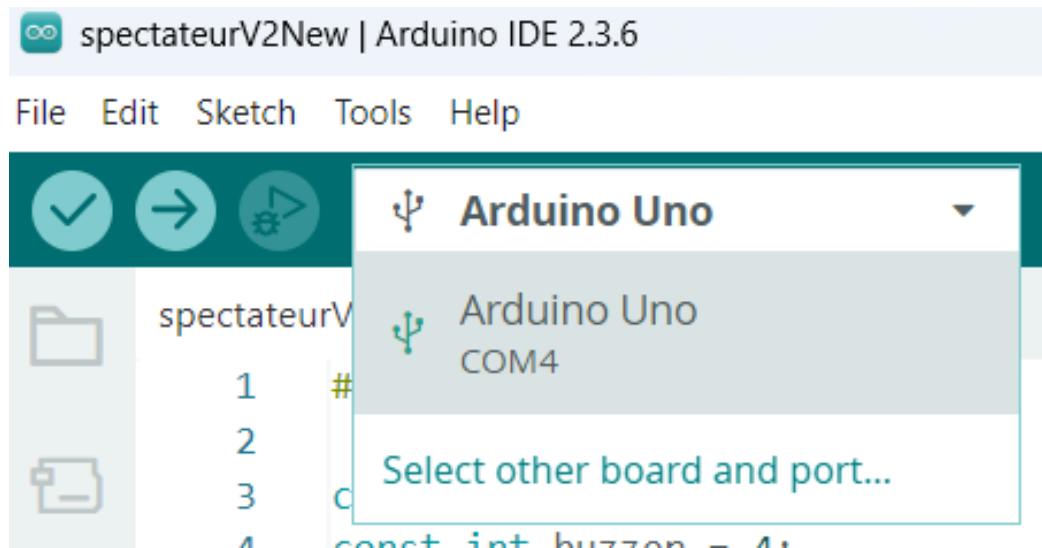
SI LES ÉTAPES PRÉCÉDENTES ONT ÉTÉ CORRECTEMENT EFFECTUÉ, VOUS ALLEZ POUVOIR CLIQUER SUR LE DOSSIER DANS LEQUEL SE TROUVE LE CODE, ET OUVRIR LE FICHIER "SPECTATEURV2NEW.INO".



FAIRE CECI OUVRIRA UNE NOUVELLE PAGE DE ARDUINO IDE.

FERMEZ LA PRÉCÉDENTE ET GARDEZ LA NOUVELLE. (CELLE-CI CONTIENDRA NORMALEMENT LES DEUX FICHI IMPORTANTS)

LORSQUE VOUS BRANCHEZ L'ARDUINO À L'ORDINATEUR, VOUS AUREZ JUSTE À CLIQUER À L'ENDROIT PRÉSENTÉ CI-DESSOUS ET CHOISIR "ARDUINO UNO".



UNE FOIS ÇA FAIT, VOUS ALLEZ POUVOIR VALIDER ET TRANSFÉRER VOTRE CODE JUSQU'À L'ARDUINO GRÂCE AU BOUTON "FLÈCHE" SE TROUVANT JUSTE À CÔTÉ.

## EXPLICATION DU CODE

SPECTATEURV2NEW.INO EST LUI-MÊME SÉPARÉ EN TROIS PARTIES :

- VOID SETUP
- VOID LOOP
- VOID PLAYMELODY

**"VOID SETUP" NE DOIT PAS ÊTRE MODIFIÉ.**

IL EST UTILISÉ POUR AFFECTER DES "TYPES DE FONCTIONNEMENT" AUX ÉQUIPEMENTS CONNECTÉS À L'ARDUINO.

"VOID LOOP" CONTIENT LE CODE MODIFIANT LES LUMIÈRES, **IL POURRA ÊTRE MODIFIÉ.**

**"VOID PLAYMELODY" NE DOIT PAS ÊTRE MODIFIÉ.** CETTE PARTIE DU CODE EST UTILISÉE POUR DÉCLARER LA FONCTION "PLAYMELODY", QUI EST UTILISÉE POUR JOUER UN SON.

```
void setup() { // |  
  pinMode(led_rouge1, OUTPUT); // |  
  pinMode(led_rouge2, OUTPUT); // |  
  pinMode(led_rouge3, OUTPUT); // |  
  pinMode(button, INPUT); // | NE PAS MODIFIER  
  pinMode(buzzer, OUTPUT); // |  
  digitalWrite(led_rouge1, LOW); // |  
  digitalWrite(led_rouge2, LOW); // |  
  digitalWrite(led_rouge3, LOW); // |  
}  
// |
```

```
void loop() {
  if (digitalRead(button) == HIGH) {
    // Insérer code ici
    playMelody(); //NE PAS TOUCHER ICI//
  } else {
    digitalWrite(led_rouge1, LOW);
    digitalWrite(led_rouge2, LOW);
    digitalWrite(led_rouge3, LOW);
  }
}
```

```
void playMelody() {
    int size = sizeof(melody) / sizeof(int);
    for (int note = 0; note < size; note++) {
        int duration = 1000 / durations[note];
        tone(buzzer, melody[note], duration); // Joue une note sur le buzzer
        int pauseBetweenNotes = duration * 1.30;
        delay(pauseBetweenNotes);
        noTone(buzzer); // Arrête la note avant de passer à la suivante
    }
}
```

## BRANCHEMENT

VOICI LES PREMIÈRES LIGNES DE CODE DU FICHIER SPECTATEURV2.

ELLES SERVENT À ASSIGNER LES LEDS, LE BUZZER ET LE BOUTON À DES "PINS", DES EMPLACEMENTS SUR L'ARDUINO.

IL N'Y AURA NORMALEMENT PAS BESOIN D'Y TOUCHER, LES ÉLÉMÉNTS SERONT DÉJÀ BRANCHÉS AU BON EMPLACEMENT.

SI CE N'EST PAS LE CAS, VOICI LE CODE AFIN DE LES REBRANCHER CORRECTEMENT.

SI LES MODULES NE SONT PAS BRANCHÉS AU MÊME ENDROIT, IL FAUT MODIFIER LES NUMÉROS SUR LE CODE.

```
#include "pitches.h"

Exemple :
const int button = 2; → le bouton est branché à l'emplacement 2
const int buzzer = 4;
const int led_rouge1 = 6;
const int led_rouge2 = 5;
const int led_rouge3 = 7;

int melody[] = {NOTE_C1, NOTE_C2}; //MODIFIABLE
int durations[] = {4, 8}; // Durées : 8 = croch
```

## MUSIQUE & SON

CI-DESSOUS EST REPRÉSENTÉE L'INTERFACE QUE POSSÈDERONT LES JEUNES POUR CRÉER LEUR MUSIQUE / SON.

 `int melody[] = {NOTE_C1, NOTE_C2, NOTE_C3, NOTE_C4};`  
 `int durations[] = {4, 8, 4, 8};`



LA PREMIÈRE LIGNE SERA CELLE RECEVANT LES "NOTES", COMME "NOTE\_C1" OU "NOTE\_B0".

**(À NOTER QU'UNE VIRGULE DOIT FORCÉMENT SUIVRE UNE NOTE, ET UN ESPACE DOIT FORCÉMENT SE TROUVER APRÈS LA VIRGULE.)**

`int melody[] = {NOTE_C1, NOTE_C2, NOTE_C3, NOTE_C4};` 



LA DEUXIÈME AFFECTE LE TEMPS QUE LA NOTE DURERA.

→ PLUS LE CHIFFRE CHOISI EST PETIT, PLUS LA NOTE DURERA LONGTEMPS)

**(UNE VIRGULE DOIT ÉGALEMENT SUIVRE LA VALEUR RENTRÉE, ET UN ESPACE DOIT SE TROUVER APRÈS LA VIRGULE)**

ATTENTION : NE PAS METTRE 0 (MÊME AVEC VIRGULE), CELA FAIT DURER LA NOTE INDÉFINIMENT.

`int durations[] = {4, 8, 4, 8};` 



CONSEILLER  
NUMÉRIQUE

## LES NOTES

LES NOTES QU'ILS PEUVENT AJOUTER SONT AFFICHÉ SI-  
DESSOUS ET EN COMMENTAIRE DANS LE CODE :

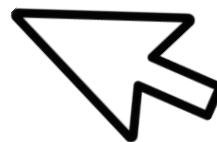
```
//NOTE_B0 NOTE_C1 NOTE_CS1 NOTE_D1 NOTE_DS1
//NOTE_B1 NOTE_C2 NOTE_CS2 NOTE_D2 NOTE_DS2
//NOTE_B2 NOTE_C3 NOTE_CS3 NOTE_D3 NOTE_DS3
//NOTE_B3 NOTE_C4 NOTE_CS4 NOTE_D4 NOTE_DS4
//NOTE_B4 NOTE_C5 NOTE_CS5 NOTE_D5 NOTE_DS5
//NOTE_B5 NOTE_C6 NOTE_CS6 NOTE_D6 NOTE_DS6
//NOTE_B6 NOTE_C7 NOTE_CS7 NOTE_D7 NOTE_DS7
//NOTE_B7 NOTE_C8 NOTE_CS8 NOTE_D8 NOTE_DS8
```

```
NOTE_E1 NOTE_F1 NOTE_FS1 NOTE_G1 NOTE_GS1 NOTE_A1 NOTE_AS1
NOTE_E2 NOTE_F2 NOTE_FS2 NOTE_G2 NOTE_GS2 NOTE_A2 NOTE_AS2
NOTE_E3 NOTE_F3 NOTE_FS3 NOTE_G3 NOTE_GS3 NOTE_A3 NOTE_AS3
NOTE_E4 NOTE_F4 NOTE_FS4 NOTE_G4 NOTE_GS4 NOTE_A4 NOTE_AS4
NOTE_E5 NOTE_F5 NOTE_FS5 NOTE_G5 NOTE_GS5 NOTE_A5 NOTE_AS5
NOTE_E6 NOTE_F6 NOTE_FS6 NOTE_G6 NOTE_GS6 NOTE_A6 NOTE_AS6
NOTE_E7 NOTE_F7 NOTE_FS7 NOTE_G7 NOTE_GS7 NOTE_A7 NOTE_AS7
```

## BIBLIOTHÈQUE SON

NOUS AVONS ÉGALEMENT TROUVÉ UNE BIBLIOTHÈQUE SUR GITHUB REGROUPANT DES MUSIQUES DE LA CULTURE POPULAIRE (MUSIQUE, FILMS, DESSINS ANIMÉS, SÉRIES) :

[HTTPS://GITHUB.COM/HIBIT-DEV/BUZZER/TREE/MASTER/SRC](https://github.com/hibit-dev/buzzer/tree/master/src)



POUR LES AJOUTER, IL SUFFIRA SIMPLEMENT DE REMPLACER LES VARIABLES MELODY ET DURATION PAR CELLES TROUVABLES DANS LE GITHUB CORRESPONDANT À LA MUSIQUE QUE L'ON VEUT METTRE.

ATTENTION : FINIR LE COPIER-COLLER JUSTE APRÈS LE "POINT VIRGULE"



RAPPEL DE CE QU'EST LA VARIABLE MELODY ET DURATION :

```
int melody[] = {NOTE_C1, NOTE_C2, NOTE_C3, NOTE_C4};  
int durations[] = {4, 8, 4, 8};
```

## EXEMPLE COURT

```
int melody[] = {NOTE_C1, NOTE_C2, NOTE_C3, NOTE_C4};  
int durations[] = {4, 8, 4, 8};
```



## LES LUMIÈRES

LES LUMIÈRES SONT CONTRÔLABLES AVEC DEUX FONCTIONS :

DIGITALWRITE(LED\_ROUGE`X`, HIGH)

CETTE FONCTION PERMET D'ALLUMER LA LED NUMÉRO `1`, `2` OU `3`, DÉPENDANT DU CHIFFRE QUE L'ON MET À LA PLACE DU `"X"`

DIGITALWRITE(LED\_ROUGE`X`, LOW)

CETTE FONCTION PERMET D'ÉTEINDRE LA LED NUMÉRO `1`, `2` OU `3`, DÉPENDANT DU CHIFFRE QUE L'ON MET À LA PLACE DU `X`

DELAY(`X`)

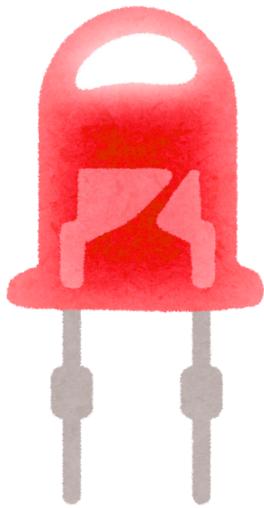
CETTE FONCTION PERMET D'EFFECTUER UNE PAUSE DANS LE CODE, LAISSANT `"X"` MILLISECONDES AVANT DE PASSER À LA SUITE.

```
void loop() {
    if (digitalRead(button) == HIGH) {
        // Insérer code ici
        playMelody(); //NE PAS TOUCHER ICI//
    } else {
        digitalWrite(led_rouge1, LOW);
        digitalWrite(led_rouge2, LOW);
        digitalWrite(led_rouge3, LOW);
    }
}
```



CONSEILLER  
NUMÉRIQUE

## EXEMPLE LUMIÈRES



```
void loop() {
    if (digitalRead(button) == HIGH) {
        digitalWrite(led_rouge1, HIGH);
        delay(1000);
        digitalWrite(led_rouge1, LOW);
        playMelody(); //NE PAS TOUCHER ICI//
    } else {
        digitalWrite(led_rouge1, LOW);
        digitalWrite(led_rouge2, LOW);
        digitalWrite(led_rouge3, LOW);
    }
}
```

SELON LE CODE ÉCRIT, LA LED ROUGE N°1 DEVRAIT S'ALLUMER PENDANT 1000 MILLISECONDES (1 SECONDE) AVANT DE S'ÉTEINDRE.

UNE FOIS ÉTEINTE, LA MUSIQUE PROGRAMMÉE AU-DESSUS DEVRAIT SE LANCER.