


Activité de rentrée 2023

– Partie programmation –



Le mot du président



Au sommaire

Programmation Arduino – Prérequis & Conseils

4

5

Niveau 0 – Aide-mémoire

Niveau 1 – Clignotant

6

8

Niveau 2 – Tricolore

Niveau 3 – Jour, Nuit, Jour...

9

10

Niveau 4 – Jour, Nuit, version soft

Niveau 5 – 🎵 Joyeux anniversaire 🎵

11

13

Niveau 6 – A vous de jouer !

Programmation Arduino – Prérequis & Conseils

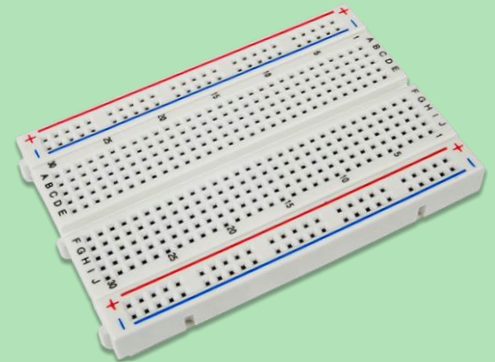


Tout d'abord, téléchargez Arduino sur votre machine si ce n'est pas déjà fait.

> Cliquez ici pour télécharger Arduino <

Déroulé de la séance de programmation

Chaque clan disposera d'une zone avec des composants et deux breadboards [plaques de prototypage]. Vérifiez que vous êtes bien avec des membres de votre clan. On avisera si les groupes sont déséquilibrés en nombre. Vous pourrez faire la programmation sur votre ordinateur personnel ou sur ceux qu'on propose dans le local.



Conseils [qui vont servir même plus tard !]

- Ne pas oublier les « ; » à la fin de chaque instruction
 - Faire attention aux minuscules/MAJUSCULES dans les variables : rouge ≠ Rouge
- Vérifiez si le programme est bon avec la coloration syntaxique et le bouton de compilation [✓]
- Regarder si la carte est bien détectée par le programme
 - ~~— Ne pas frapper sa machine [ou son voisin] si ça ne fonctionne pas !~~

Et surtout, n'hésitez pas à demander de l'aide à un membre d'Ares ou votre voisin si vous êtes coincés ! Bonne séance !

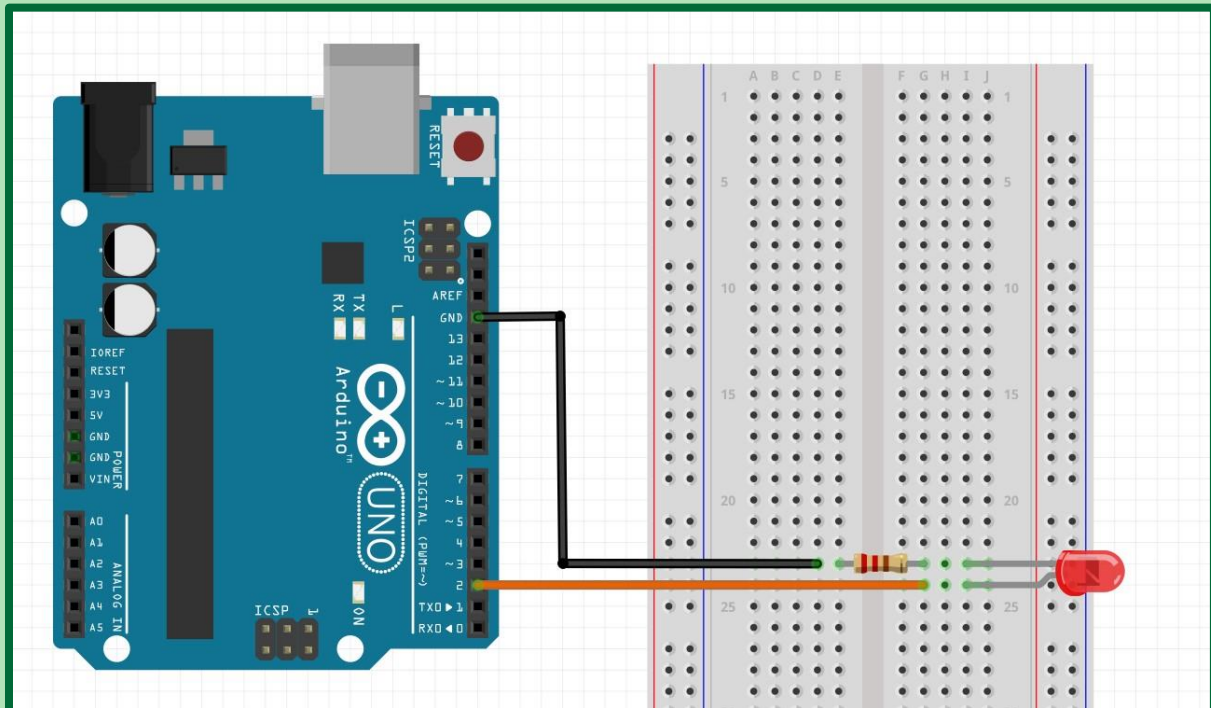
Niveau 0 – Aide-mémoire

Les variables		Les opérateurs	
int : entier naturel codage sur 4 octets [-2^{31} ; $2^{31} - 1$]	float : nombre à virgule codage sur 4 octets	Arithmétiques : +, -, *, /, % Affectation : =, +=, -=, %=, =, ^=, <=<, >=> Logique : &&, , <, >, <=, >=, ==, !=	
char : caractère codage sur 1 octet [-128; 127]	double : nombre à virgule codage sur 8 octets		
Condition		Boucles	
<pre>if (x<3){ printf(« x est inférieur à 3 ») } else{ printf(« x est supérieur ou égal à 3 ») }</pre>		<pre>int x = 0; while(x < 5){ printf(« coucou\n ») x++; }</pre>	<pre>int somme = 0; for(int i = 0; i < 100; i++){ somme+=i; }</pre>
Directives de préprocesseur		Les tableaux	
<pre>//Inclusion d'une bibliothèque, ici servo #include <servo.h> //Lors de la compilation, tous les N du code seront remplacés par des 7 #define N 7</pre>		<pre>#define SIZE 5 #define SIZE2 3 //Exemples de tableaux à 1 dimension int tab[SIZE] = {1, 9, 0, 1}; char tabchar[SIZE2] = {"a", "r", "e", "s"}; //Initialisation du premier tableau for(int i = 0; i<SIZE; i++){ tab[i] = 0; }</pre>	

Fonctions Arduino	
pinMode (int pin, INPUT)	Définir le type d'une sortie d'une broche (INPUT = entrée, OUTPUT = sortie)
digitalWrite (int pin, HIGH)	Définir l'état d'une broche numérique (HIGH = 1, LOW = 0)
digitalRead (int pin)	Lire l'état d'une broche numérique
delay (1000)	Création d'une temporisation de 1000 ms = 1 seconde
analogWrite (int pin, int dutyCycle)	Crée un signal analogique (PWM) de rapport cyclique dutyCycle ∈ [0; 255]
analogRead (int pin)	Lire la valeur d'une broche analogique
Serial.begin (int baud)	Initialisation de la liaison série. En général, baud = 9600
Serial.println (« Ares »)	Affiche « Ares » sur le port série
int = map (int val, int lmin, int lmax, int Omin, int Omax)	Changement d'échelle. Réajuste les valeurs de val ∈ [lmin; lmax] pour [Omin; Omax]
tone (int pin, int frequency)	Joue une fréquence « frequency » sur un buzzer.
noTone (int pin)	Stoppe la génération d'une onde sonore sur un buzzer.
Servo monServo	Initialisation d'un objet servo. Permet de contrôler un servomoteur
monServo. attach (int pin)	Attache la broche pin à l'objet monServo. Permet de commander le servomoteur
monServo. write (90)	Définit la position du servomoteur sur 90 degrés

Niveau 1 – Clignotant

Les montages seront donnés pour tous les niveaux suivants sous cette forme :



Et comme à Ares nous sommes généreux, voilà en page suivante un code qui vous permettra d'allumer la LED et son explication :

Après téléversement du code sur votre carte, tout le code compris dans le « setup » ne se lancera qu'une seule fois

Après téléversement du code sur votre carte, tout le code compris dans le « loop » s'exécutera à l'infini !

```
1  const int ledPin = 2;
2
3  void setup() {
4      pinMode(ledPin, OUTPUT);
5  }
6
7  void loop() {
8      digitalWrite(ledPin, HIGH);
9  }
10
11
```

On définit un entier qui correspond à la broche sur laquelle on a connecté la LED, ici 2

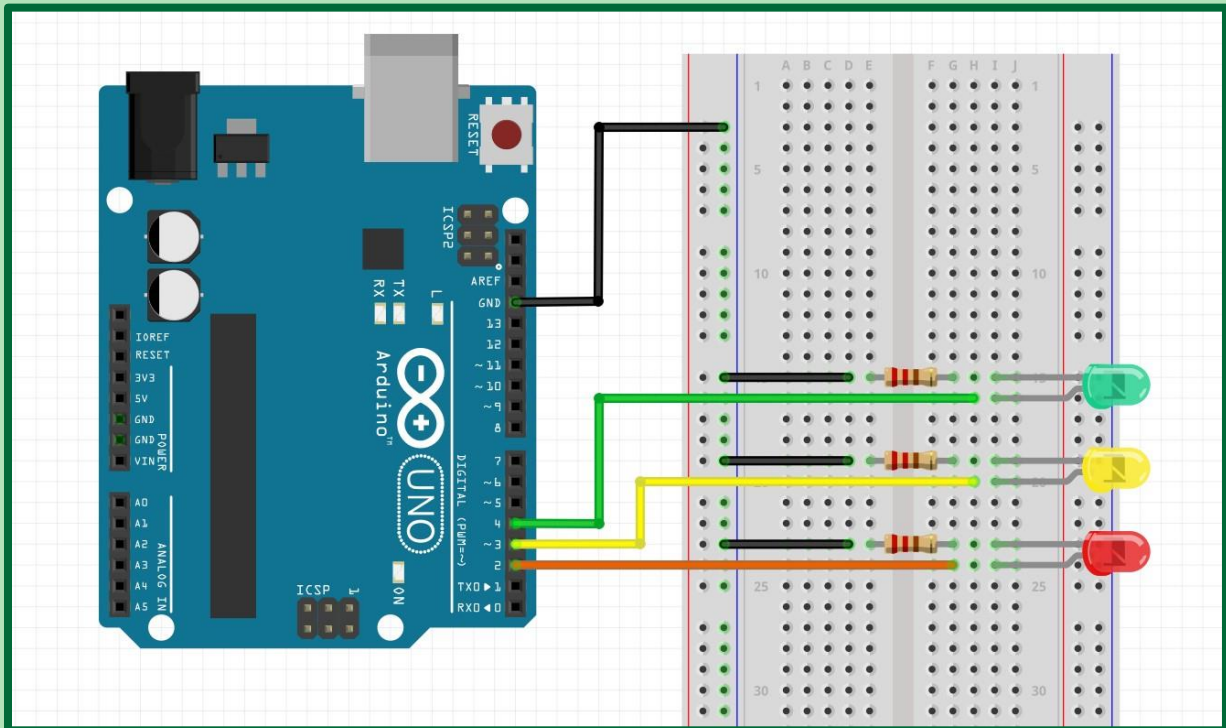
La LED étant une sortie, on définit la broche ledPin en tant que tel avec la fonction « pinMode[] »

Dans la boucle infinie, on utilise digitalWrite pour mettre la LED à l'état haut (HIGH) et ainsi

Le saviez-vous ? : Le langage de programmation utilisé pour coder sur Arduino est le C++. Il est souvent utilisé pour coder des applications mobiles, des jeux vidéo, des logiciels de bureautique, en robotique...

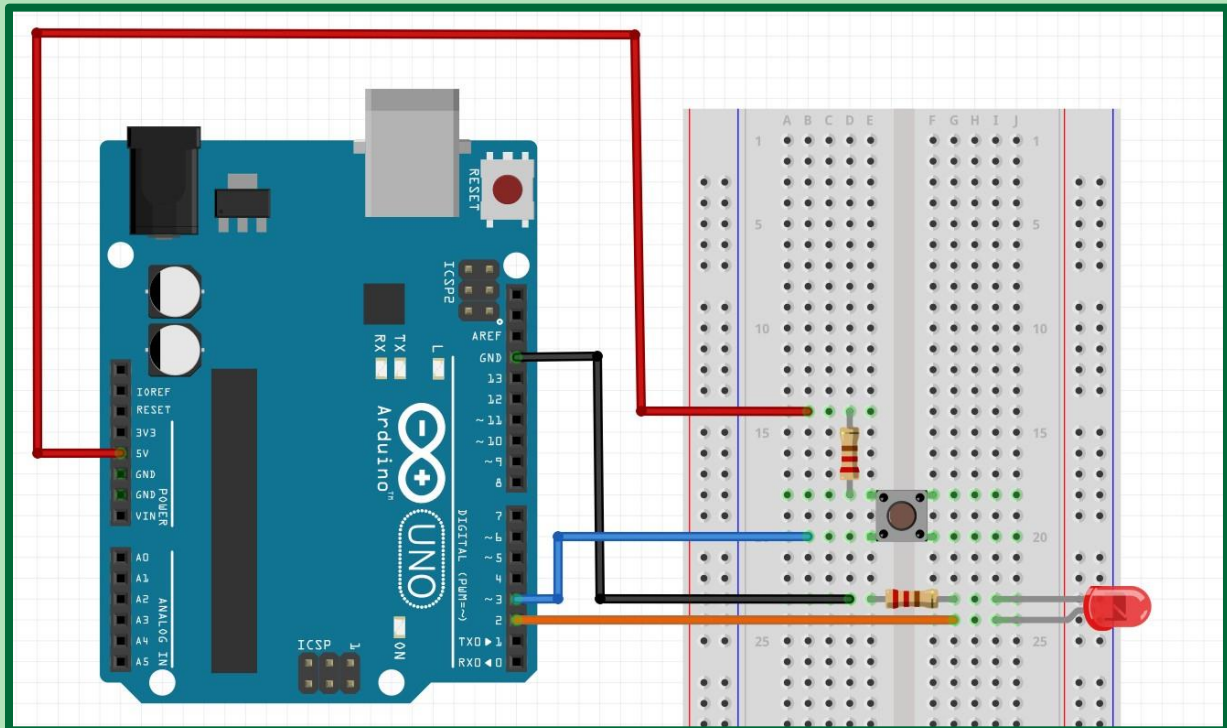
Mission : Faire clignoter la LED en changeant son état toutes les secondes

Niveau 2 – Tricolore



Mission : Changer de manière successive l'état des 3 LEDs

Niveau 3 – Jour, Nuit, Jour...

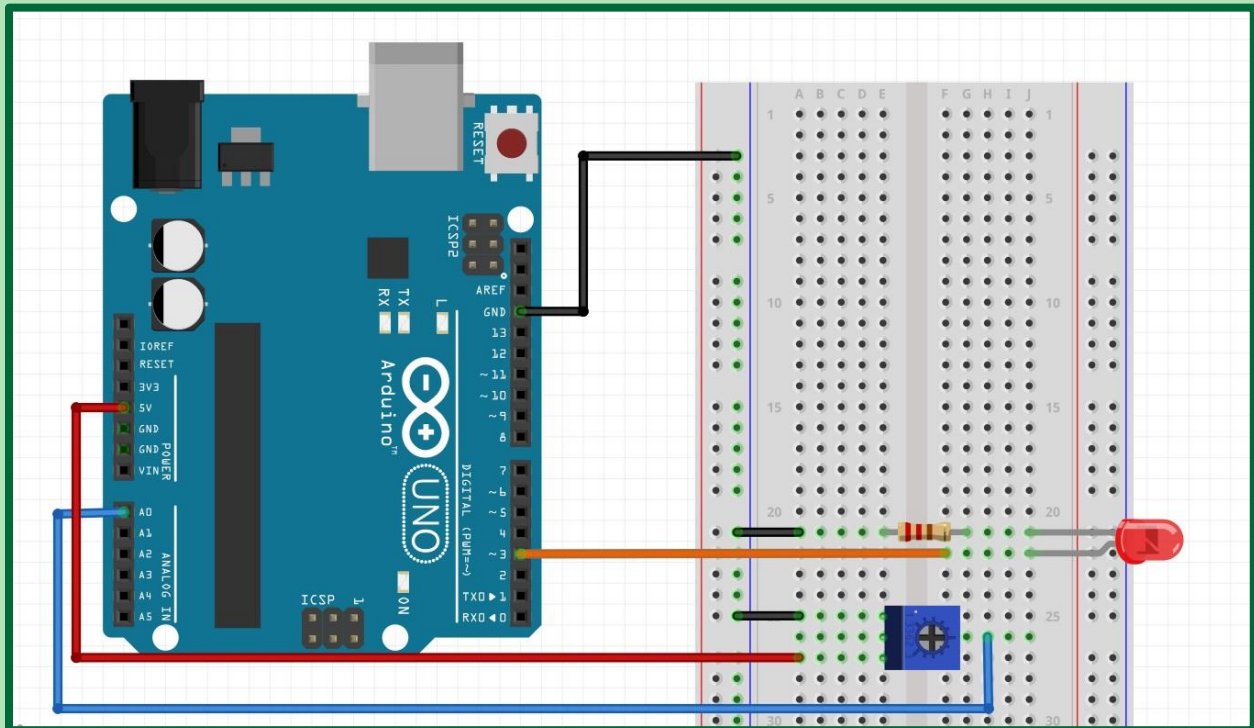


Mission : Commander l'allumage de la LED grâce à un bouton

Peut-être que pour connaître l'état du bouton, une fonction existe pour lire l'état d'une broche numérique... Peut-être même qu'elle est dans l'aide-mémoire. La vie est bien faite non ?



Niveau 4 – Jour, Nuit, version soft

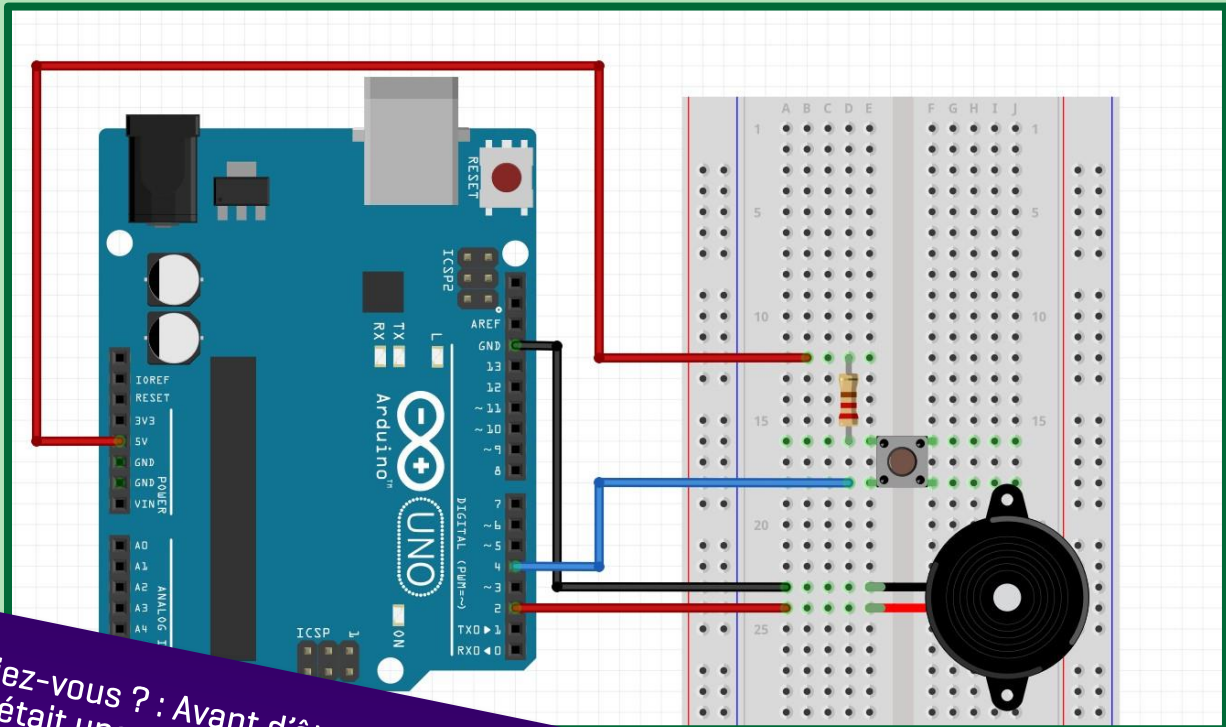


Mission : Faire varier l'intensité de la LED émise à l'aide d'un potentiomètre

La broche du potentiomètre est connectée à un port analogique. Hmm... Analogique, analogique, analog...



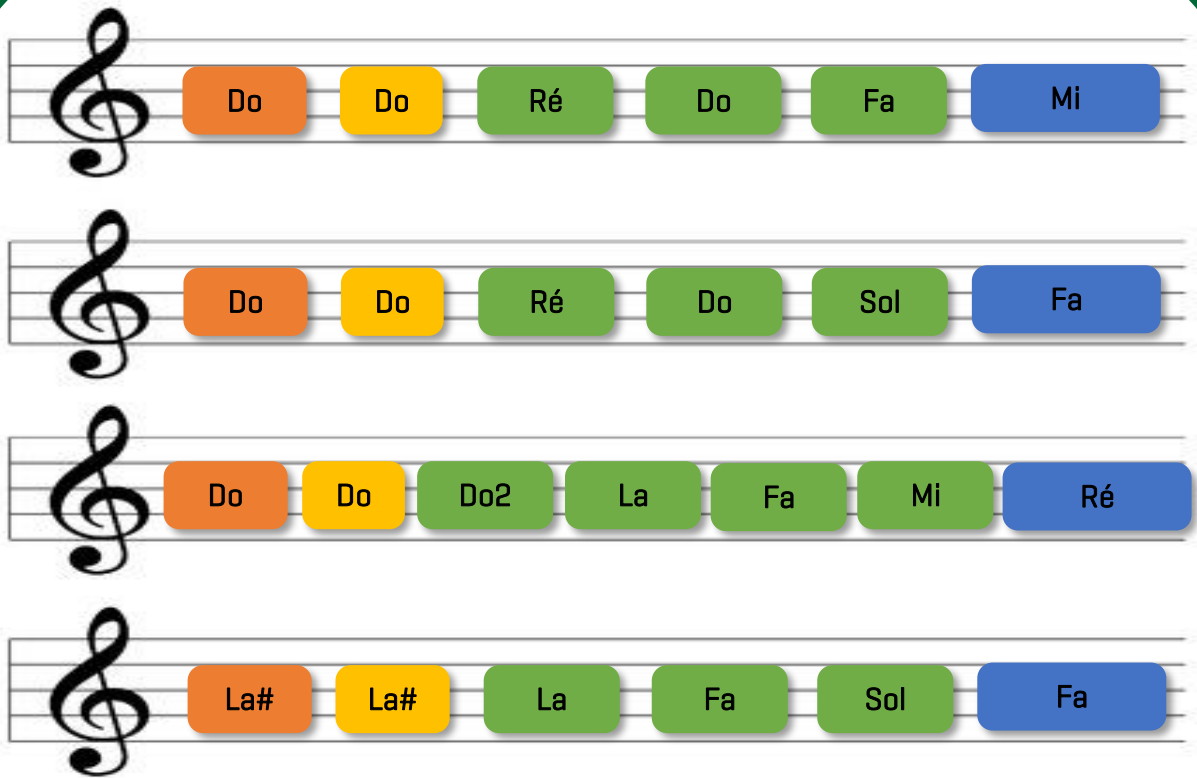
Niveau 5 – Joyeux anniversaire 🎵



Le saviez-vous ? : Avant d'être un pôle du BDTech, Ares était une association de robotique à part fondée le 27 septembre 1996. C'est bientôt son anniversaire !

Fréquences des notes et partition simplifiée de « Joyeux anniversaire » :

Notes	Do	Do#	Ré	Ré#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si	Do2
f (Hz)	262	277	294	311	330	349	370	391	415	440	466	493	523



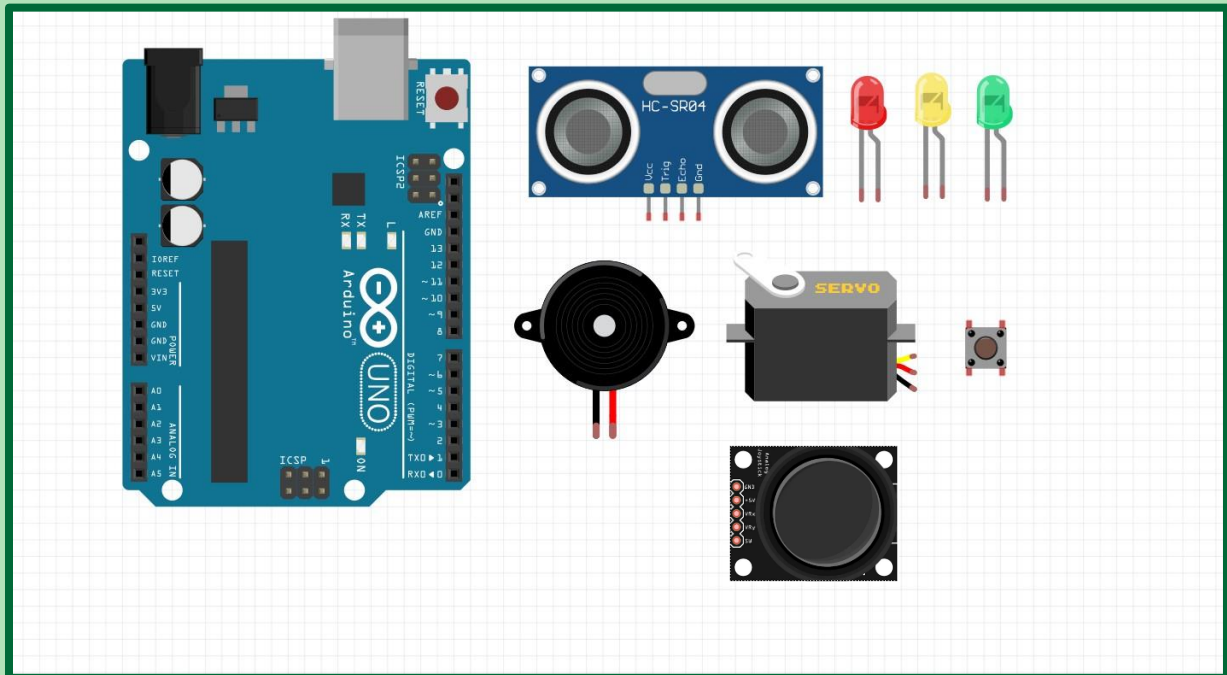
Il y'a 25 notes dans jouées « Joyeux anniversaire ». On peut utiliser la méthode bourrin qui serait d'écrire 25 fois les fonction `tone()` & `noTone()` pour chaque note, mais je suis sûr qu'il y'a une solution plus rapide ! Une couleur de note correspond à une durée différente, pensez à varier les délais entre celles-ci.



Mission : Jouer « Joyeux anniversaire » lorsque le bouton est appuyé

Bonus : Points supplémentaires pour votre clan si vous arrivez à jouer une autre mélodie !

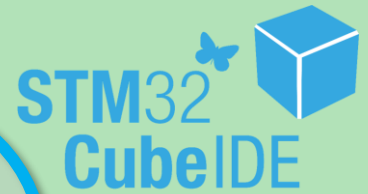
Niveau 6 – À vous de jouer !



Mission : Avec le matériel proposé et les autres membres de votre clan, réalisez quelque chose d'original !

FIN.

Micro-introduction à STM32



Ici à l'ENSEA, vous n'allez plus revoir Arduino de sitôt !
L'école nous apprend à utiliser STM32 Cube IDE un logiciel un peu plus complexe, mais très important car il est utilisé au sein de beaucoup d'entreprises.

Deux différences majeures

On peut programmer en langage C++, mais aussi en langage C. C'est ce dernier qui est enseigné à l'école.

Il faut définir les entrées & sorties du microcontrôleur,

Niveau 1 EX – Clignotant version STM32

