

## Séance ArduBlock et Arduino

### Partie 1 - Boite à sourire

Brancher la « boite à sourire » sur une prise USB avant d'allumer l'ordinateur.  
Allumer l'ordinateur et lancer le logiciel Arduino :



Vérifier le type de carte et le port de connexion dans le Menu Outils.  
Les cartes sont de 2 types Arduino Uno ou Arduino Nano/Atmega 328

#### **Exercice 1** : Faire clignoter 1 seconde diode

Télécharger le fichier d'exemple Blink :  
Fichier/exemple/Basic/Blink

Un programme arduino se décompose en 3 parties :

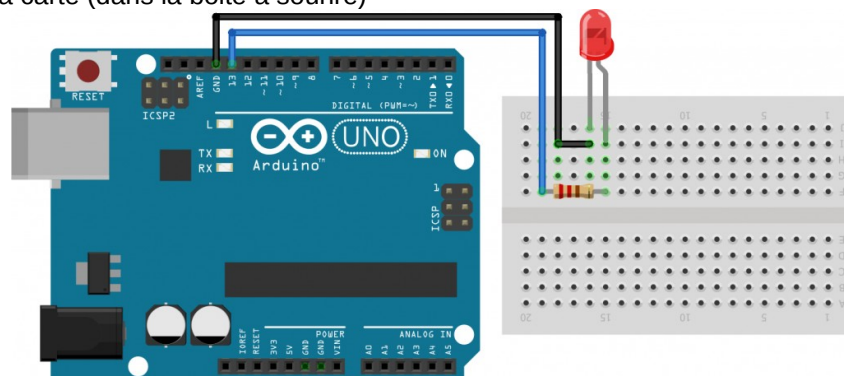
- Déclaration des variables
- Initialisation des variables (void setup)
- Programmation (void loop)

```
const int ledPin = 13; // déclaration de la variable du pin du bouton

void setup()
{
  // initialisation de la PIN de la LED
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite( ledPin , HIGH );
  delay( 1000 );
  digitalWrite( ledPin , LOW );
  delay( 1000 );
}
```

Branchement de la carte (dans la boite a sourire)



fritzing

Nota il faut protéger la carte et la LED avec une résistance de 220Ω

Cliquer sur Téléverser ➡

Le programme est transféré de l'ordinateur vers la carte arduino. Une led clignote.

#### **Exercice 2** : Faire clignoter 1 seconde led

Observation et appropriation du code

// les diodes sont sur les pins 7 à 13.

// Le programme s'exécutera dès que la carte sera alimenté en électricité.

### Exercice 3 : Chenillard

Repérer le numéro de toutes les diodes en les faisant s'allumer les unes après les autres dans l'ordre comme un chenillard.

## Partie 2 – Jeu du métronome

### Exercice 4 : Création d'un jeu Arduino : Métronome

#### Étape 1 : Commander l'allumage d'une diode

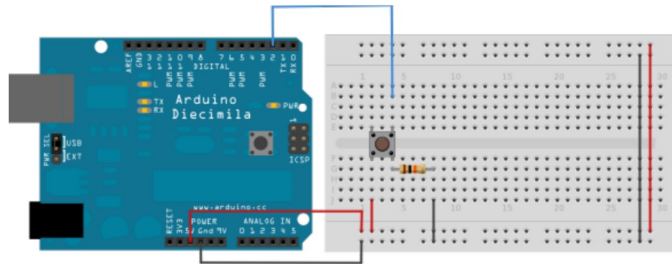
Nouveau sketch

```
const int ledPin = 13; // déclaration de la variable du pin du bouton
const int buttonPin = 2; // déclaration de la variable du pin du bouton
int buttonState = 0; // variable qui donne le statut du bouton
```

```
void setup() {
  // initialisation de la PIN de la LED
  pinMode(ledPin, OUTPUT);
  // initialisation de le PIN du bouton
  pinMode(buttonPin, INPUT);
}
```

```
void loop(){
  // lecture du statut du bouton
  buttonState = digitalRead(buttonPin);
```

```
  // si le bouton est appuyer
  if (buttonState == HIGH) {
    // allumer la LED:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```



#### Étape 2 : Création du jeu du métronome

```
const int buttonPin = 2; // déclaration de la variable du pin du bouton
const int ledPin = 13; // déclaration de la variable du pin du bouton
int buttonState = 0; // variable qui donne le statut du bouton
```

```
unsigned long dateDernierChangement = 0;
unsigned long intervalle = 0 ;
```

```
void setup() {
  // initialisation de la PIN de la LED
  pinMode(ledPin, OUTPUT);
  // initialisation de le PIN du bouton
  pinMode(buttonPin, INPUT);
  // initialisation du calcul de la durée de temps
  intervalle = millis() - dateDernierChangement;
}
```

```
void loop(){
  // lecture du statut du bouton
  buttonState = digitalRead(buttonPin);

  // si le bouton est appuyer
  if (buttonState == HIGH) {
    // initialisation du tempo de reference
    intervalle = millis() - dateDernierChangement;
    dateDernierChangement = millis();

    if (intervalle >1200 || intervalle < 800) {
      digitalWrite(ledPin, HIGH);
      delay(50) ;
      digitalWrite(ledPin, LOW);
      delay(50) ;
      digitalWrite(ledPin, HIGH);
    }
  }
}
```

```
    delay(50) ;
    digitalWrite(ledPin, LOW);
    delay(50) ;
    digitalWrite(ledPin, HIGH);
    delay(50) ;
    digitalWrite(ledPin, LOW);
    delay(50) ;
    digitalWrite(ledPin, HIGH);
    delay(50) ;
    digitalWrite(ledPin, LOW);
  }
  else {
    digitalWrite(ledPin, HIGH);
    delay(300) ;
    digitalWrite(ledPin, LOW);
  }
}
```

### Étape 3 : Visualiser l'écart de temps

Nous allons rajouter

dans la partie setup la ligne : `Serial.begin(9600);`

et juste en dessous de `dateDernierChangement = millis();`

`Serial.println(intervalle);`

ce code permet de transférer de l'arduino vers l'ordinateur la valeur de la variable « intervalle »

→ Ouvrir une fenêtre série configurée à 9600 bauds

### Étape 4 : Le jeu fonctionne et il est temps de mettre un score

Déclaration d'une nouvelle valeur `int` en début de programme

Initialisation de la variable à 0 dans le setup

implémentation de la variable dans la boucle `else` : `score++` ;

et écriture dans la console série de la variable `score`

`Serial.println("score"+score) ;`

```
const int buttonPin = 2; // déclaration de la variable du pin du bouton
const int ledPin = 13; // déclaration de la variable du pin du bouton
int buttonState = 0; // variable qui donne le statut du bouton
int score = 0 ; // score
```

```
unsigned long dateDernierChangement = 0;
unsigned long intervalle = 0 ;
```

```
void setup() {
// initialisation de la PIN de la LED
pinMode(ledPin, OUTPUT);
// initialisation de le PIN du bouton
pinMode(buttonPin, INPUT);
// initialisation du calcul de la durée de temps
intervalle = millis() - dateDernierChangement;
Serial.begin(9600);
}
```

```
void loop(){
// lecture du statut du bouton
buttonState = digitalRead(buttonPin);

// si le bouton est appuyer
if (buttonState == HIGH) {
// initialisation du tempo de reference
intervalle = millis() - dateDernierChangement;
dateDernierChangement = millis();
Serial.println(intervalle);
```

```
if (intervalle >1200 || intervalle < 800) {
digitalWrite(ledPin, HIGH);
delay(50) ;
digitalWrite(ledPin, LOW);
delay(50) ;
digitalWrite(ledPin, HIGH);
delay(50) ;
digitalWrite(ledPin, LOW);
delay(50) ;
digitalWrite(ledPin, HIGH);
delay(50) ;
digitalWrite(ledPin, LOW);
delay(50) ;
digitalWrite(ledPin, HIGH);
delay(50) ;
digitalWrite(ledPin, LOW);
Serial.println("score'+score) ;
}
else {
digitalWrite(ledPin, HIGH);
delay(300) ;
digitalWrite(ledPin, LOW);
score++ ;
}
}
}
```