

Partie 1 - Faire clignoter des leds

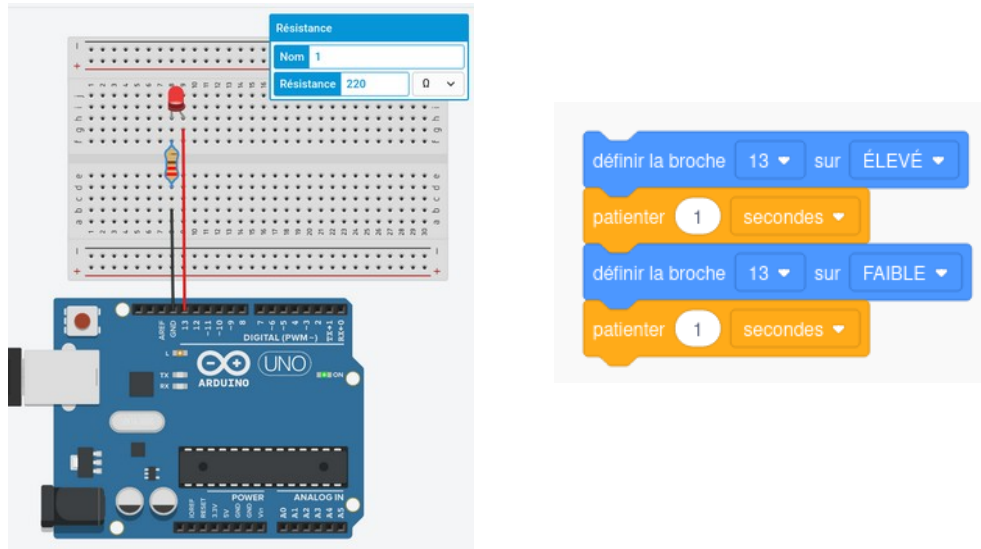
L'objectif est de simuler le montage ci-dessous à l'aide de la plateforme Tinkercad.

La LED s'allumera et s'éteindra toutes les secondes.

Allumer l'ordinateur et lancer le logiciel Tinkercad pour simuler le montage:

Exercice 1 : Faire clignoter 1 seconde la led

Construire le montage sous tinkercad



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

Appuyer sur CODE + TEXT pour visualiser le code.

Un programme arduino se décompose en 3 parties comme pour Processing :

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

```
void setup()
{
  // initialisation de la PIN de la LED
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite( 13 , HIGH );
  delay( 1000 );
  digitalWrite( 13 , LOW );
  delay( 1000 );
}
```

Branchement de la carte (arduino)

Lancer le logiciel arduino

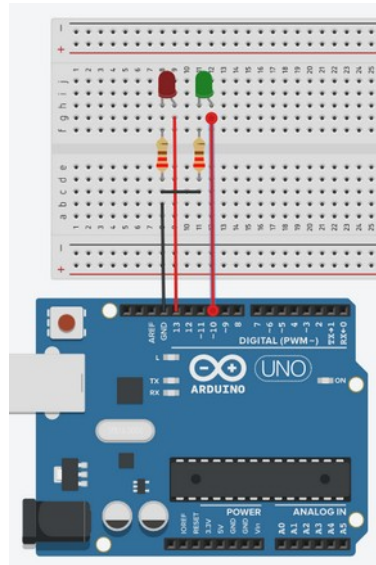
Réaliser le branchement

Copier coller le code dans le logiciel arduino

Cliquer sur Téléverser ➡

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

Exercice 2 : Faire clignoter une seconde led en alternance

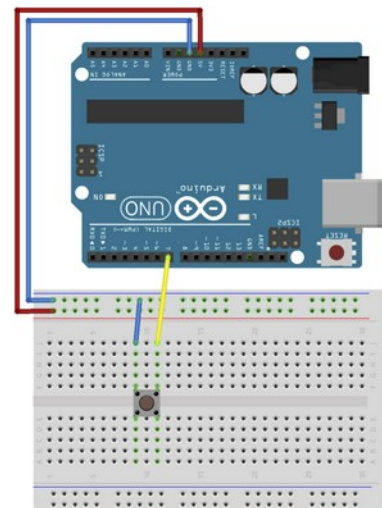
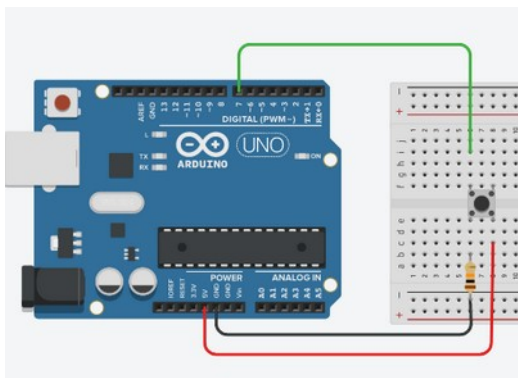


Observation et appropriation du code
// les diodes sont sur les pins 10 et 13.

- Tester sous tinkercad
- Copier coller sur le logiciel Arduino
- Effectuer le montage sur la breadboard
- Téléverser le programme

Exercice 3 : le bouton

Il y a deux façon de connecter un bouton



Essayer de trouver le programme pour faire allumer une diode (diode intégrée pin 13) si l'on appuis sur le bouton.

- Tester le programme sur Tinkercad
- Connecter le bouton et Téléverser.

```

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);

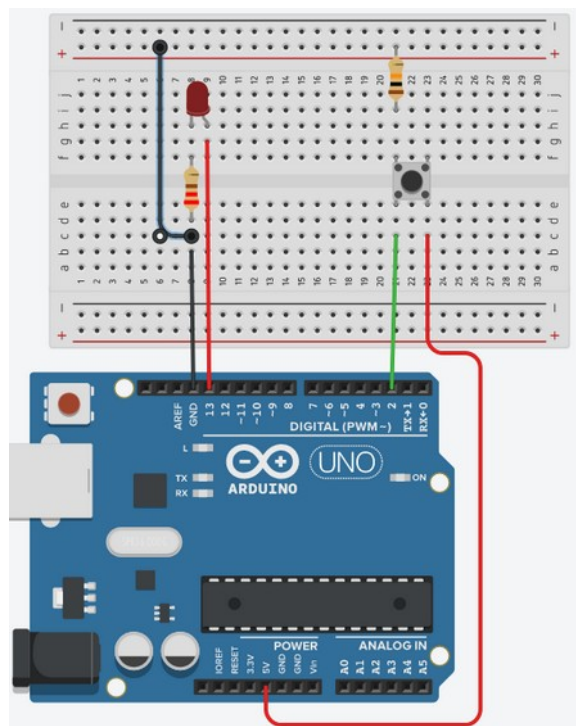
  // si le bouton est appuyer
  if (buttonState == HIGH) {
    // allumer la LED:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}

```

Partie 2 – Jeu du métronome

Exercice 4 : Création d'un jeu Arduino : Métronome

Le jeu du métronome consiste a appuyer de façon très régulière sur le bouton poussoir pour garder la cadence de 1 seconde. Si la cadence n'est pas conservé, on perd:)



```

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 boutonState = 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
  boutonState = digitalRead(buttonPin);

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

    if (intervalle >1200 || intervalle < 800) {
// signal de défaite
      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);
      delay(50) ;
      digitalWrite(ledPin, HIGH);
      delay(50) ;
      digitalWrite(ledPin, LOW);
    }
    else {
// signal de réussite le jeu continu
      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 serie de la variable score

Serial.println("score"+score) ;

```

const int boutonPin = 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++ ;
```

```
  }
```

```
  }
```

```
}
```