

TP 1 : Arduino avec Tinkercad

📌 Objectif : Progresser du niveau débutant à avancé en réalisant des exercices interactifs sur **Tinkercad** avec **Arduino**.

TP1.1. Premier Programme: Allumer et éteindre une LED (Blink).

Objectif : Apprendre à utiliser une **sortie numérique**.

Matériel : **Arduino Uno, 1 LED, 1 résistance 220Ω**.

Principe : Faire clignoter une LED avec un délai de 1 seconde.

Si "Hello world" est le premier programme que vous avez appris en programmation, "Blink" est votre premier programme en développement Arduino.

I. Préparation du TP sur Tinkercad

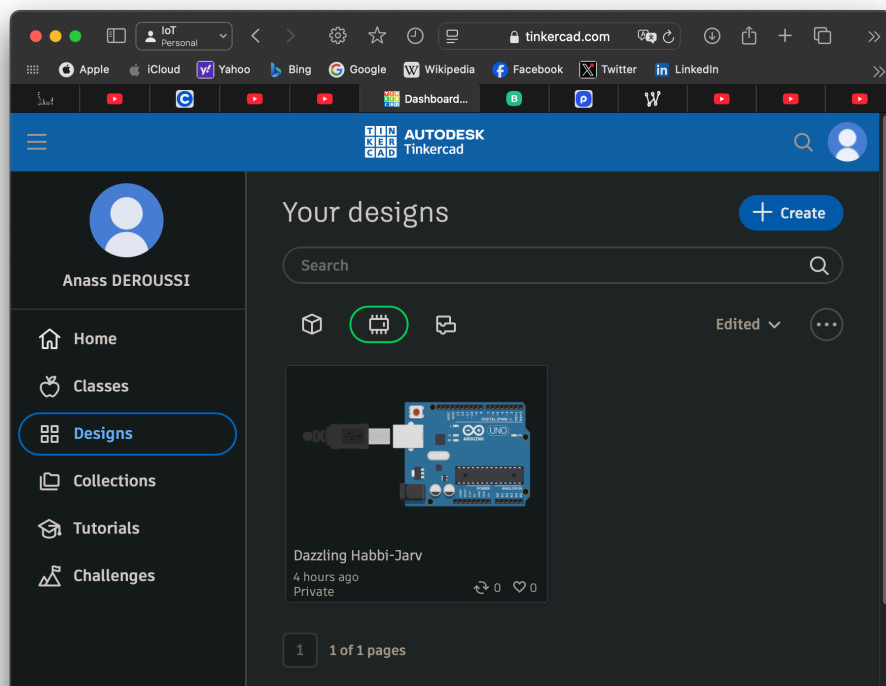
Étape 1 : Accéder à Tinkercad

1. Ouvrir Tinkercad

- o Accédez à [Tinkercad](https://tinkercad.com) et Connectez-vous ou créez un compte si nécessaire.

2. Créer un nouveau circuit

- o Cliquez sur "**Circuits**" dans le menu et Appuyez sur "**Créer un circuit**".



II. Montage du circuit.

Étape 2 : Ajouter les composants

Dans la bibliothèque de Tinkercad (section composants), ajoutez :

- ✓ **1 Arduino Uno**
- ✓ **1 LED (rouge, verte, etc.):** Pour que la LED s'allume, l'anode doit être connectée au + (exemple : 5V) et la cathode au - (GND).
- ✓ **1 Résistance de 220Ω** (pour protéger la LED)
- ✓ **1 Breadboard** (facultatif)
- ✓ **Fils de connexion**

Étape 3 : Connexion des composants

Suivez ces étapes pour assembler le circuit :

1. **Placer la LED sur la breadboard :**
 - o La **jambe longue (+) de la LED (anode)** sera connectée à la **broche 13** de l'Arduino.
 - o La **jambe courte (-) (cathode)** sera connectée à la résistance de **220Ω**.
2. **Ajouter la résistance :**
 - o Reliez **une extrémité de la résistance** à la **cathode de la LED**.
 - o Connectez **l'autre extrémité** au **GND (masse) de l'Arduino**.

Schéma des connexions :

Arduino Pin 13 → Anode de la LED (+)

Cathode de la LED (-) → Résistance 220Ω → GND de l'Arduino

III. Programmation dans Tinkercad

Étape 4 : Ajouter le code Arduino

1. **Ouvrir l'éditeur de code**
 - o Cliquez sur **"Code"**, puis sélectionnez **"Texte"** pour passer en mode programmation.

2. Entrer le programme suivant :

```
void setup() {  
  pinMode(13, OUTPUT); // Définir la broche 13 comme une sortie  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // Allumer la LED  
  delay(1000);           // Attendre 1 seconde  
  digitalWrite(13, LOW); // Éteindre la LED  
  delay(1000);           // Attendre 1 seconde  
}
```

IV. Exécution et Test

Étape 5 : Lancer la simulation

1. Vérifiez vos connexions.
2. Cliquez sur "Démarrer la simulation".
3. **Observez la LED** : elle doit s'allumer et s'éteindre toutes les secondes.

V. Expérimentations Supplémentaires

♦ Expérience 1 : Changer la vitesse de clignotement

- Modifiez la ligne `delay(1000);` en `delay(500);` pour que la LED clignote plus rapidement.

♦ Expérience 2 : Ajouter une deuxième LED

- Ajoutez une LED sur la broche 12 et programmez-la pour qu'elle s'allume quand la première s'éteint.

Conclusion

✓ Ce TP vous a permis d'apprendre :

- ✓ Comment utiliser une sortie numérique avec Arduino.
- ✓ Comment câbler une LED avec une résistance sur Tinkercad.
- ✓ Comment utiliser `pinMode()`, `digitalWrite()` et `delay()` en programmation.

🚀 Prochain défi : Contrôler l'allumage de la LED avec un bouton-poussoir !

TP1.2 : LED contrôlée par un bouton-poussoir

Objectif : Utiliser une **entrée numérique**.

Matériel : **Matériel requis dans Tinkercad**

1. **Arduino Uno**
2. **1 LED (rouge, verte, etc.)**
3. **1 bouton-poussoir**
4. **1 résistance de 220Ω (pour protéger la LED)**
5. **1 résistance de 10kΩ (pull-down pour le bouton-poussoir)**
6. **Fils de connexion**
7. **Breadboard (facultatif pour une meilleure organisation du circuit)**

Principe : Allumer la LED en appuyant sur le bouton.

Nous voulons réaliser un montage qui permet d'allumer une LED branchée sur le PIN 13 durant 5 secondes une fois un bouton poussoir actionné. Le bouton poussoir est lié au PIN A0. Réaliser le montage et écrivez le programme approprié.

Étape 1 : Créer un circuit sur Tinkercad

1. **Accédez à Tinkercad :**
 - Connectez-vous sur [Tinkercad](#) et créez un nouveau circuit.
2. **Ajoutez les composants nécessaires :**
 - Recherchez **Arduino Uno, LED, bouton-poussoir, résistances** et **breadboard** dans la bibliothèque de composants.

Étape 2 : Réaliser le câblage

1. **Brancher la LED :**
 - Reliez l'**anode (jambe longue)** de la LED à la **broche numérique 13** de l'Arduino.
 - Connectez la **cathode (jambe courte)** de la LED à une **résistance de 220Ω**.
 - Reliez l'autre extrémité de la résistance à la **masse (GND)** de l'Arduino.
2. **Brancher le bouton-poussoir :**

- Connectez **une borne** du bouton à la **broche A0** de l'Arduino.
- Connectez cette même borne à une **résistance de 10kΩ**, et reliez l'autre extrémité de la résistance à la **masse (GND)** (pull-down).
- Connectez **l'autre borne** du bouton au **5V** de l'Arduino.

Explications:

- **Pull-down avec résistance 10kΩ** : Assure que la broche A0 lit un signal LOW (0) lorsqu'aucun courant ne traverse le bouton.
- **Résistance 220Ω pour la LED** : Protège la LED en limitant le courant qui la traverse.

⚡ Résumé du câblage :

- **LED** :
 - Broche 13 → Résistance 220Ω → Anode (+) de la LED
 - Cathode (-) de la LED → GND.
- **Bouton-poussoir** :
 - Broche A0 → Bouton → Résistance 10kΩ → GND
 - L'autre borne du bouton → 5V.

Étape 3 : Programmer l'Arduino

1. **Ouvrir l'éditeur de code dans Tinkercad** :
 - Cliquez sur "**Code**", puis sélectionnez l'option "**Texte**".
2. **Écrire le programme suivant** :

```

int ledPin = 13;      // Broche de la LED
int buttonPin = A0;  // Broche du bouton-poussoir
int buttonState = 0; // Variable pour stocker l'état du bouton

void setup() {
  pinMode(ledPin, OUTPUT); // Configurer la LED comme sortie
  pinMode(buttonPin, INPUT); // Configurer le bouton comme entrée
}

void loop() {
  buttonState = digitalRead(buttonPin); // Lire l'état du bouton

  if (buttonState == HIGH) {           // Si le bouton est pressé
    digitalWrite(ledPin, HIGH);        // Allumer la LED
    delay(5000);                       // Attendre 5 secondes
    digitalWrite(ledPin, LOW);         // Éteindre la LED
  }
}

```

Explications :

- **pinMode()** : Configure les broches comme entrées (bouton) ou sorties (LED).
- **digitalRead()** : Lit l'état du bouton (HIGH = pressé, LOW = relâché).
- **digitalWrite()** : Allume ou éteint la LED.
- **delay(5000)** : Maintient la LED allumée pendant 5 secondes.

Étape 4 : Lancer la simulation

1. Cliquez sur "**Démarrer la simulation**" dans Tinkercad.
2. Appuyez sur le **bouton-poussoir** (en cliquant dessus dans la simulation).
 - **Attendu** : La LED s'allume pendant **5 secondes**, puis s'éteint.

Expérimentations supplémentaires

Utiliser millis() : Remplacez delay() par millis() pour rendre le programme non bloquant.