

1. Présentation

	<p>CARTE</p> <p>Arduino est une plateforme matérielle et logicielle de développement d'applications embarquées.</p> <p>Elle se compose d'une carte électronique basée autour d'un <u>microcontrôleur</u> (ATMEL AVR) comportant un certain nombre d'entrées et de sorties (les broches ou pins <u>analogiques et numériques</u>) permettant la connexion de <u>capteurs, ou d'actionneurs</u>.</p>
	<p>Logiciel ou IDE :</p> <p>Le logiciel de programmation des modules Arduino est une application <u>Java</u>, libre servant d'éditeur de code et de compilateur, et qui peut transférer le <u>firmware</u> et le programme au travers de la liaison USB.</p> <p>Le langage de programmation utilisé est un mélange de C et de C++ , restreint et adapté aux possibilités de la carte.</p>
	<p>Modules « shield »</p> <p>Cartes supplémentaires se connectant sur le module Arduino pour augmenter les possibilités et les capacités de la carte.</p> <p>Exp : afficheur graphique couleur, interface Ethernet, GPS, WIFI, GSM, <u>Carte de puissance moteur</u>.....</p>

Q1 : Utiliser internet et donner l'explication des mots soulignés en rouge.

Q2 : Donner le numéro des broches analogiques et les broches numériques sur la carte.

Documentation : <http://lense.institutoptique.fr/mine/arduino-entrees-analogiques/>

2. Utilisation d'un simulateur

Cette partie à pour objectif la mise en œuvre d'un simulateur d'Arduino.

Ouvrir le navigateur **Chrome**, et taper l'adresse "**https://www.tinkercad.com/circuits**" dans la barre des adresses puis cliquer sur "**S'INSCRIRE**" et suivre les étapes de 1 à 15.

The image consists of nine numbered screenshots illustrating the process of creating a Tinkercad account and navigating to the Circuits page:

- Step 1:** The browser address bar shows `tinkercad.com/circuits`.
- Step 2:** The 'Connexion' menu is open, and the 'S'INSCRIRE' button is highlighted.
- Step 3:** The 'Personnel' section is visible, with the 'General account' button highlighted.
- Step 4:** The 'Créer un compte' form shows the 'Pays, France' dropdown selected.
- Step 5:** The 'Créer un compte' form shows the 'SUIVANT' button highlighted.
- Step 6:** The 'Créer un compte' form shows the 'Mot de passe' field with instructions: 'Créer un mot de passe, un mail de confirmation vous sera envoyer. Ensuite connectez-vous.'
- Step 7:** The 'Créer un compte' form shows the 'CRÉER UN COMPTE' button highlighted.
- Step 8:** The 'Introducing Tinkercad Classroom' page shows the 'Circuits' button highlighted.
- Step 9:** The 'Circuits' page shows the 'Créer un circuit' button highlighted.

The image shows a sequence of steps in the Tinkercad software:

- Step 10:** The component library is open, and the 'Clignotant' (Blink) component is highlighted with a red box.
- Step 11:** The 'Code' tab is selected in the top right, also highlighted with a red box.
- Step 12:** The 'Texte' (Text) block is selected in the block palette, highlighted with a red box.
- Step 13:** A dialog box asks 'Etes-vous sûr?' (Are you sure?). The 'Continuer' (Continue) button is highlighted with a red box.
- Step 14:** The 'Démarrer la simulation' (Start simulation) button is highlighted with a red box.
- Step 15:** The code editor shows the following code:


```

1 /*
2  This program blinks pin 13 of the Arduino (the
3  built-in LED)
4  */
5
6 void setup()
7 {
8   pinMode(13, OUTPUT);
9 }
10
11 void loop()
12 {
13  // turn the LED on (HIGH
14  digitalWrite(13, HIGH);
15  delay(1000); // Wait for
16  // turn the LED off by ma
17  digitalWrite(13, LOW);
18  delay(1000); // Wait for 1000 millisecond(s)
19 }

```

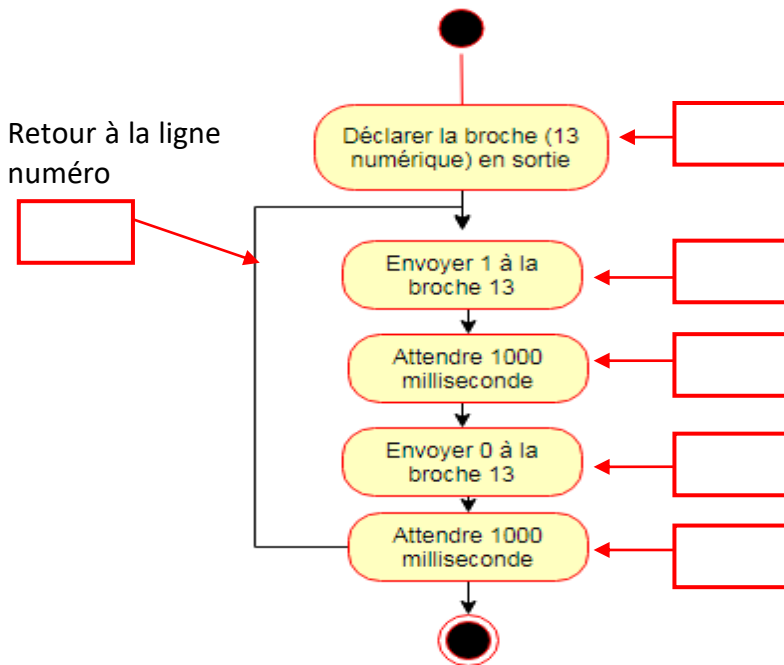
 A red box highlights the code, and a red arrow points to the '1000' in line 15. A red box next to the arrow contains the text: 'Changer la valeur 1000 par 2000 et démarrer la simulation une deuxième fois' (Change the value 1000 to 2000 and start the simulation a second time).

Q3 : Chercher sur le site arduino la signification des lignes de code suivantes.

pinMode :
INPUT, OUTPUT :
digitalWrite :
HIGH:
LOW:
delay:
void setup() :
void loop() :

Q4 : Que signifie le numéro **13** sur le code ?

Q5 : Soit l'organigramme suivant, mettre les lignes **8, 14,15,17,18** du programme à l'endroit correspondant.



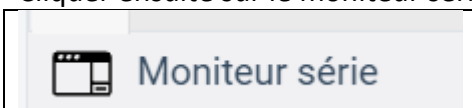
Q6. Pour comprendre la fonction **loop**, utiliser le code ci-dessous

```

void setup(){
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Bonjour");
}
  
```

Cliquer ensuite sur le moniteur série pour observer l'affichage.



Q7. Que fait **Serial.println** et **Serial.begin** ? .

Documentation : <https://www.arduino.cc/reference/fr/language/functions/communication/serial/begin/>

Q8 : Commenter la première ligne du programme ci-dessous avec l'écriture suivante
« **//-----Votre commentaire** »

Documentation : <https://culture-informatique.net/japprends-a-programmer-variables/>

```
Programme
int maled=13; //-----Votre commentaire ici

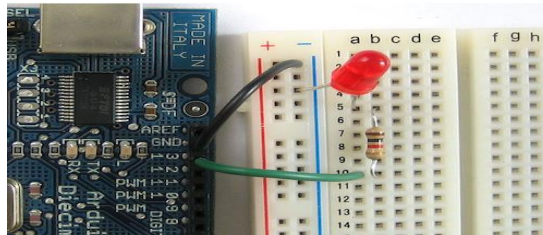
void setup(){
pinMode(maled, OUTPUT);
}

void loop(){
  digitalWrite(maled, HIGH);
  delay(1000);

digitalWrite(maled, LOW);
  delay(1000);
}
```

3. Câblage d'un circuit "virtuel"

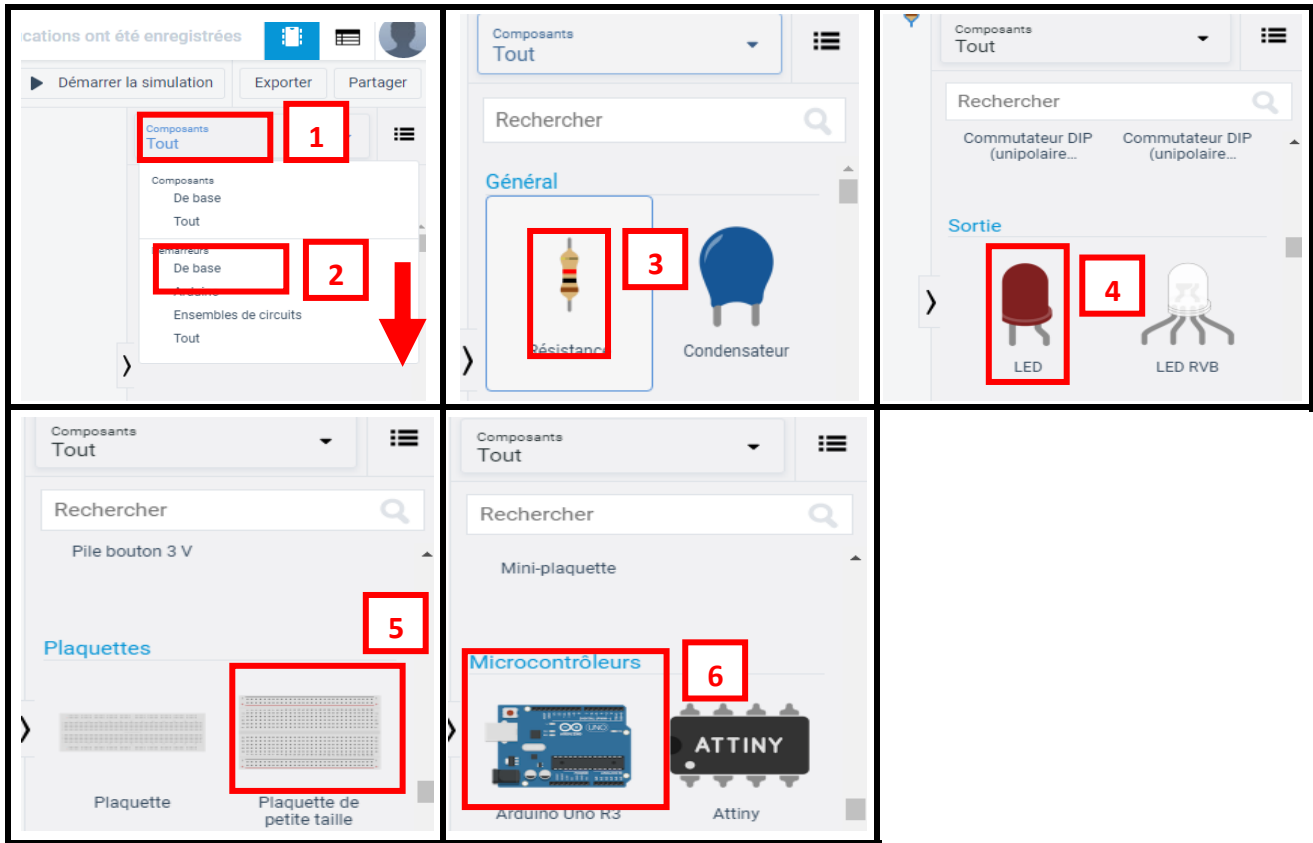
On désire que la carte Arduino fasse clignoter deux **leds** branchées sur deux sorties digitales avec une plaque d'essai.



Il faut tout d'abord **ajouter les composants**. Deux **leds**, deux **résistances**, Une **breadboard(plaque d'essai)**, un **arduino**.

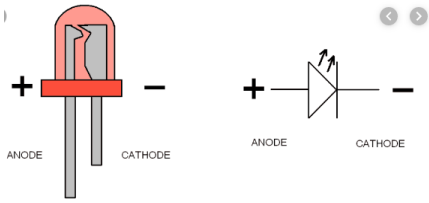


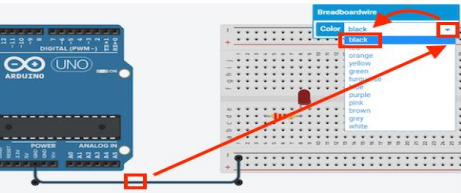
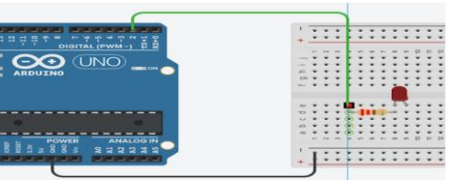
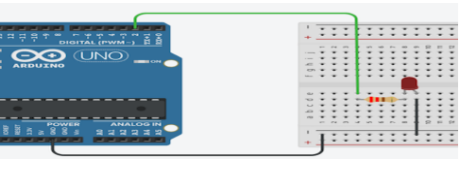
Cliquer sur composants et naviguer dans la liste de composants grâce à l'**ascenseur sur la droite**.

Effectuer un "**Clic and Drop**" pour déposer les composants dans la zone de conception :

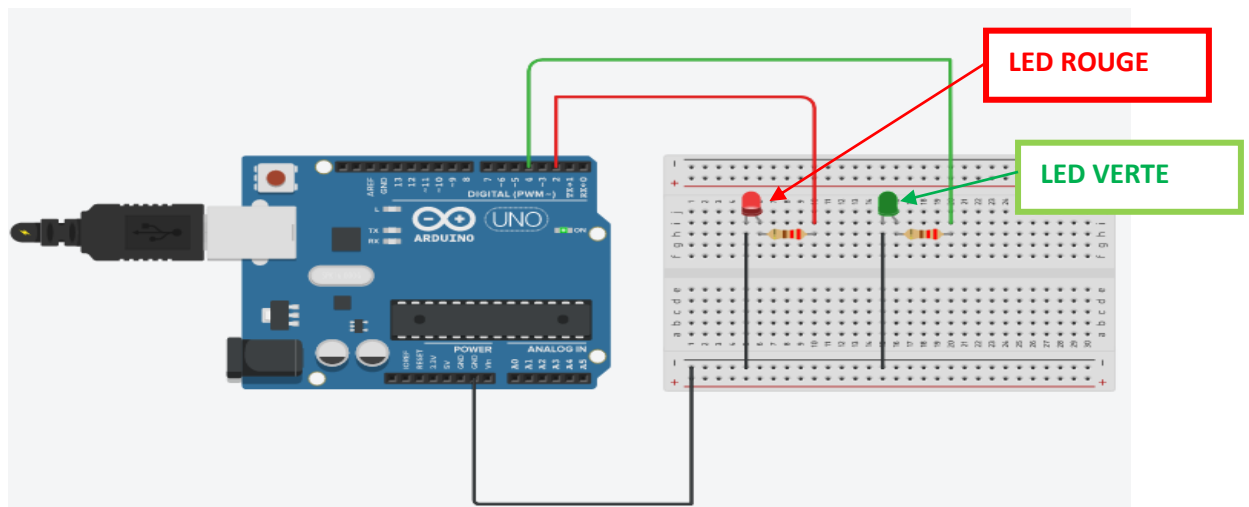


Réflexion et câblage !!!!!.

<p>Un mot sur la Breadboard :</p> <p>On peut traduire par "Planche à pain". C'est une platine de prototypage qui permet de câbler facilement des composants.</p> <p>Les lignes de connexions sont représentées ci-contre :</p>	
<p>Les sorties digitales de la carte Arduino délivrent une tension trop élevée pour les LED. Il faut donc les protéger avec une résistance de 220 ohm.</p> <p>Afin de changer la valeur de la résistance, cliquer sur la résistance, puis entrer la valeur "220" et enfin sélectionner l'unité "ohm" :</p>	
<p>La résistance est mal orientée pour notre montage.</p> <p>Cliquer sur la résistance et cliquer (<i>plusieurs fois</i>) sur le bouton orienter afin que la LED soit verticale.</p>	

<p>Rien ne se passe si la LED est câblée à l'envers !</p> <p>En effet la patte "Anode" doit être reliée à la sortie numérique ensuite la résistance, alors que la cathode doit être reliée à la masse.</p> <p>On peut repérer l'anode car sa patte est la plus (+)</p>	
<p>Disposer les composants tel qu'indiqué sur l'illustration ci-contre (<i>attention, il faut que l'une des pattes de la résistance soit connectée avec une patte de la LED</i>) :</p>	
<p>Il faut maintenant réaliser les fils de connexion. On va relier la pin de masse de la carte Arduino (GND) sur la ligne (-) de la breadboard.</p> <p>Cliquer sur la pin GND et rejoindre la ligne (-) en cliquant pour couder le fil :</p>	
<p>Ce premier fil étant le fil de masse (Ground ou GND en anglais), il faut qu'il soit de couleur de noire.</p> <p>Cliquer sur le fil et changer sa couleur pour du noir</p>	
<p>On désire utiliser la pin digitale n°2 de la carte Arduino pour allumer la LED.</p> <p>Réaliser la connexion entre la pin 2 et la ligne de breadboard reliée à la patte de la résistance :</p>	
<p>Enfin, relier la 2ème patte de la LED à la masse avec un fil noir :</p>	

Objectif : Réaliser le schéma ci-dessous :



4. Réalisation d'un algorithme

Faire clignoter les deux leds, rouge et verte du montage précédent une après l'autre

Pour cela, il faut mettre le programme ci-dessous dans la zone de programmation texte et simuler.

```
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(2, LOW);
  digitalWrite(4, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(4, LOW);
}
```

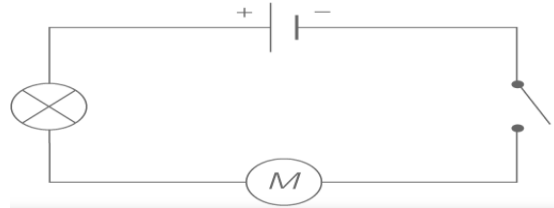
Q9. Réaliser l'algorithme du programme en utilisant le lien suivant : <https://app.diagrams.net/>

Q10. Rajouter une troisième led de couleur bleu et faire clignoter les trois leds une après l'autre.

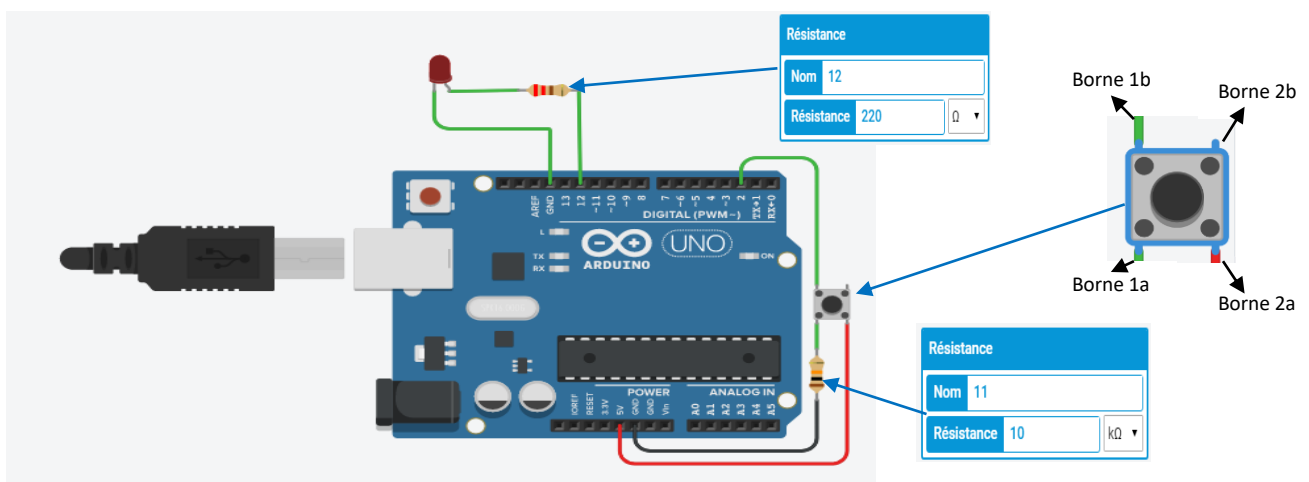
5. Programmation itérative avec un interrupteur.

5.1) L'interrupteur

Un **interrupteur** est un organe, physique ou virtuel, permettant d'interrompre ou d'autoriser le passage d'un **flux électrique**.



Dans la bibliothèque des composants chercher un **bouton poussoir**, une **led**, un **Arduino Uno**, deux **résistances** et réaliser le câblage ci-dessous.



Avant de commencer il faut lire la valeur du bouton, pour cela il faut utiliser le programme ci-dessous ensuite **ouvrir le moniteur série** et **appuyer sur le bouton** pour voir le changement.

```
int bouton; //---Déclaration de la variable bouton

void setup(){

pinMode(2, INPUT); //----Initialisation de la broche 2 en entrée
Serial.begin(9600); // ----Initialisation de la transmission port série

}

void loop(){

bouton=digitalRead(2); //-----lire la valeur du bouton et la mettre dans la variable bouton
Serial.println(bouton); // -----Afficher la valeur de la variable

}
```

Q11. Utiliser la fonction de comparaison **if** avec les fonctions **Serial.begin** et **Serial.println** pour afficher « **Bonjour** » si le bouton est à **1** sinon afficher « **Bonsoir** »,

Documentation : <https://plaisirarduino.fr/if-if-else/>

Ensuite, mettre le programme de l'interrupteur dans la zone programmation texte et appuyer sur le bouton.

```
Programme interrupteur
1 int button=2;
2 int led=12;
3 int boutonvalue;

4 void setup(){
5   pinMode(button, INPUT);
6   pinMode(led, OUTPUT);
7 }

8 void loop(){
   boutonvalue = digitalRead(button);
   if(boutonvalue == HIGH) {digitalWrite(led, HIGH); }

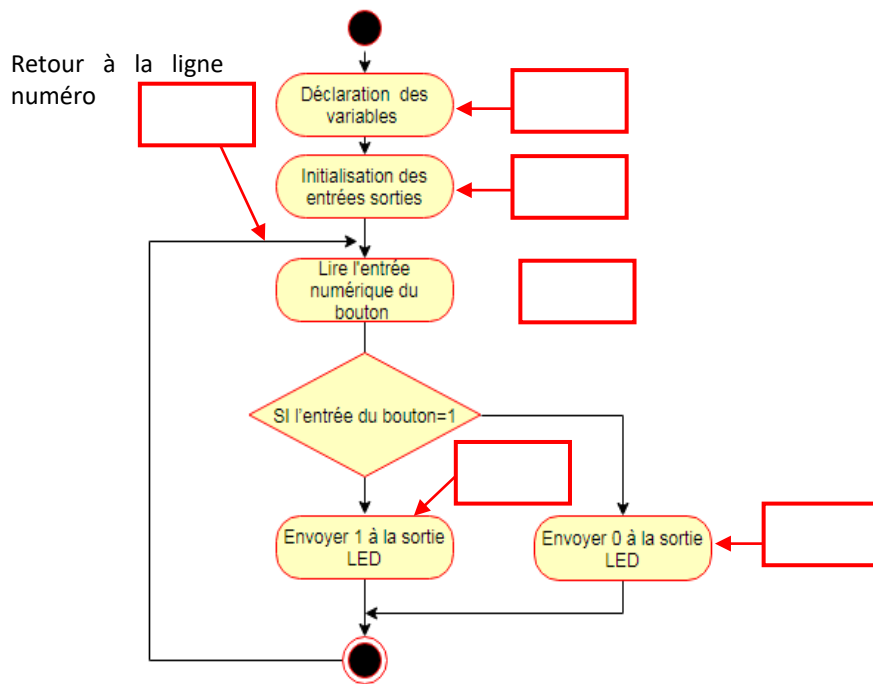
   else {digitalWrite(led, LOW); }
```

Q12 : Cherche sur le site Arduino la définition de la ligne de code **digitalRead**.

Q13 : Dans une condition **if** il est possible d'utiliser de nombreux opérateurs de comparaison, citer ces opérateurs.

Documentation : <https://plaisirarduino.fr/if-if-else/>

Q14 : Mettre les numéros des lignes du code interrupteur à l'endroit correspondant sur l'organigramme ci-dessous. **Une case peut prendre plusieurs numéros !**



Q15 : Changer le `bouttonvalue == HIGH` de la ligne 9 par `bouttonvalue = HIGH`. Commenter ce changement.

Q16. Il faut maintenir le bouton pour que la led reste allumer, il est possible d'utiliser **deux boutons**, un **pour allumer** et l'autre **pour éteindre**.

Réaliser ce schéma et le programme correspondant.

Réflexion sur le code (tester ce code sur un seul bouton) !!

```

int bouton=2;
int led=12;
int boutonvalue;

void setup(){
  pinMode(bouton, INPUT);
  pinMode(led, OUTPUT);
}

void loop(){
  boutonvalue = digitalRead(bouton);

  if(boutonvalue == HIGH)
  {
    digitalWrite(led, HIGH);
  }
}

```

