

*Nous aurons besoin de*



*L'Arduino Uno*



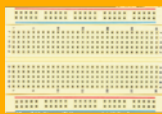
*Le programme  
EmetteurMorse*



*L'éditeur de  
programme*



*L'ordinateur*



*La planche  
d'essai*



*1 bouton  
poussoir*



*1 résistance  
220  $\Omega$*



*1 LED*



*Le cable  
USB*

Cette version améliorée allume une LED et émet un son avec le buzzer lorsqu'on appuie sur le bouton poussoir.

Il faut construire le circuit comme indiqué dans le schéma.

Cette fois-ci le programme est va devoir lire la valeur du bouton (LOW ou HIGH) et indiquer à la LED et au buzzer ce qu'ils doivent faire en utilisant les pins 13 et 8.

La ligne qui commence par `//` est un commentaire. Ce sont des informations pour le programmeur.

- `setup` qui dit ce qu'il faut faire 1 seule fois au démarrage
- `loop` qui dit ce qu'il faut faire de manière répétée à chaque cycle

Ce que fait la fonction est décrit entre les signes { }. Ces signes s'appellent des **accolades**.

`void loop() { ... }` est aussi une fonction.

Ces fonctions contiennent les instructions que l'on donne à l'Arduino. Ce sont aussi des fonctions mais celles-ci sont décrites dans du code qui sera envoyé avec notre programme. On n'a pas à les écrire. On appelle ce code fourni avec l'Arduino une **bibliothèque de fonctions**.

La LED s'éclaire quand il y a du courant (valeur HIGH) et s'éteint quand il n'y en a pas (valeur LOW).

- `pinMode(13, OUTPUT)` indique que l'alimentation est faite par le pin13
- `digitalRead(2)` lit la quantité de courant que le bouton laisse passer (soit LOW pour 0, soit HIGH pour 5 Volts)
- `digitalWrite(13, v)` envoie la quantité de courant v sur le pin 13. v est une variable et contient la valeur HIGH ou LOW selon ce qui a été lu sur le bouton.
- `delay(1000)` attend 1s (1000 ms)
- `digitalWrite(13, LOW)` envoie la quantité minimale de courant sur le pin 13. La lampe s'éteint.
- `tone(8, note, duree)` joue un son sur le buzzer

Les valeurs qui sont entre parenthèses après les fonctions s'appellent des **paramètres**. Par exemple, un paramètre permet d'utiliser `delay` pour attendre plus ou moins longtemps selon la valeur.

- `delay(1000)` attend 1s (1 000 ms)
- `delay(10000)` attend 10s (10 000 ms)