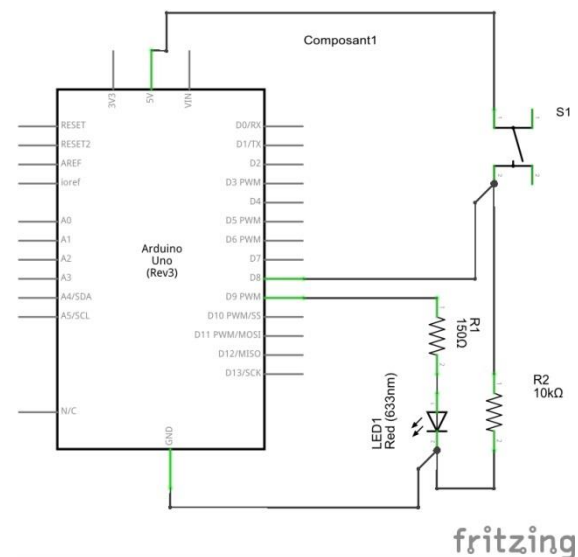
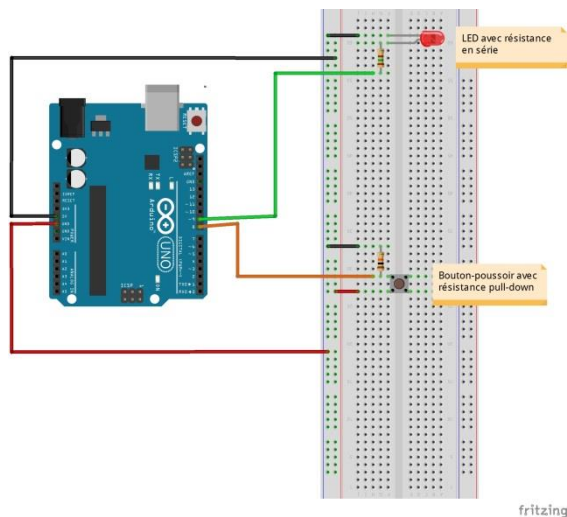


INTENTION PEDAGOGIQUE

PUBLIC	<input checked="" type="checkbox"/> Tous <input type="checkbox"/> MLDS <input type="checkbox"/> HPI	Avoir déjà réalisé un circuit avec l'arduino.	NIVEAU NECESSAIRE AVOIR REALISER LA FICHE 1
POUR QUOI ?			
Durée	2 heures		
Domaine scientifique abordée	Electronique, Electricité		
Objectifs de la séance	Commander une DEL avec un interrupteur, découvrir les résistances de fonction pull-Down et pull-up		
COMMENT ?			
COMPETENCES VISEES	<ul style="list-style-type: none">- Découverte la commande pour acquérir une entrée digital.- Piloter la DEL en fonction de l'état de l'interrupteur.		
PRODUCTION FINALE	La réalisation individuelle d'un circuit avec un interrupteur et une DEL.		
MATERIEL A METTRE A DISPOSITION	<ul style="list-style-type: none">- UN ORDINATEUR POUR CHAQUE APPRENANT AVEC LE LOGICIEL ARDUINO ⁽¹⁾ INSTALLEE <p>(1) Pour télécharger le logiciel arduino se rendre sur le site www.arduino.cc</p>		
MATERIEL APORTE PAR L'ANIMATEUR	<ul style="list-style-type: none">- UNE CARTE ARDUINO POUR CHAQUE APPRENANT- UN CABLE DE LIAISON ENTRE L'ORDINATEUR ET LA CARTE ARDUINO- UNE DEL (STARTER KIT)- UN INTERRUPTEUR		
CONNAISSANCES ACQUISES LORS DE L'EXPERIENCE	EN ARDUINO : <ul style="list-style-type: none">- LA PROGRAMMATION D'UNE BROCHE NUMERIQUE EN ENTREE (INPUT) ;- STRUCTURE DE CONTROLE (IF, ELSE)- ENREGISTRER L'ETAT D'UN BOUTON DANS UNE VARIABLE- FONCTION DIGITALREAD() ;		
	EN ELECTRONIQUE : <ul style="list-style-type: none">- RESISTANCE PULL-DOWN ET PULL-UP		

SCHEMA ELECTRIQUE



La résistance pour la DEL est de 150 Ω et celle du bouton-poussoir est une résistance de 10K Ω .

[Lien vers la vidéo du montage](#)

CODE DE L'EXPERIENCE

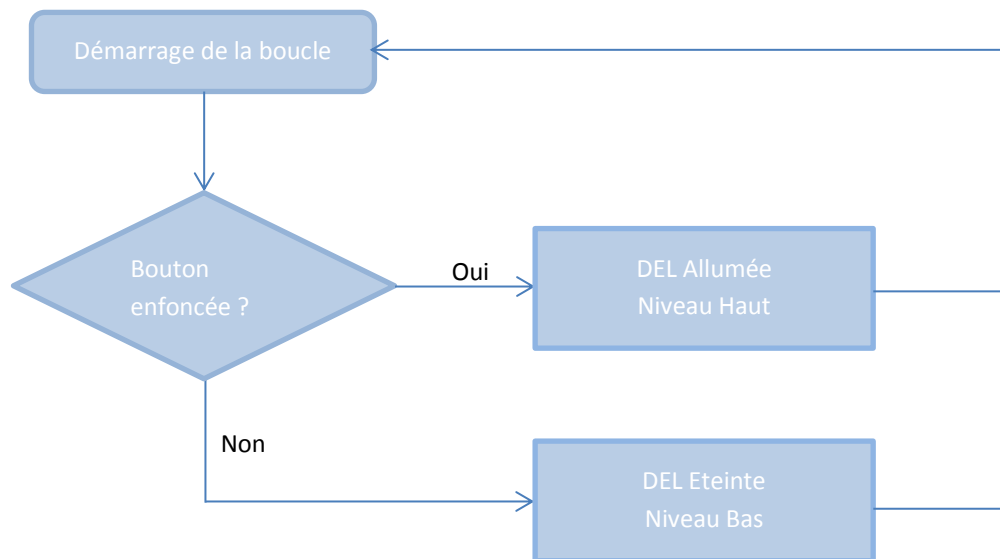
```
const int Led = 8; // LED en broche 8
const int Bouton = 9; // le bouton en broche 9
int BoutonEtat; // variable pour enregistrer l'état du bouton

void setup() {
  // initialize the digital pin as an output.
  pinMode(Led, OUTPUT); // Broche de la led en tant que sortie
  pinMode(Bouton, INPUT); // Broche du bouton poussoir en tant d'entrée
}

void loop() {
  BoutonEtat = digitalRead(Bouton);
  if (BoutonEtat == HIGH) {
    digitalWrite(Led, HIGH); // Allume la LED
  }
  else {
    digitalWrite(Led, LOW); // éteint la LED
  }
}
```

REVUE DE CODE

Fonctionnement du programme :



L'organigramme se lit de la manière suivante, lorsque l'exécution du programme arrive dans la partie de la boucle sans fin, nous regardons toujours l'état de l'entrée du bouton-poussoir et enregistrer dans une variable (ici BoutonEtat), La ligne qui permet de faire ce traitement est la suivant :

FONCTION DIGITALREAD() :

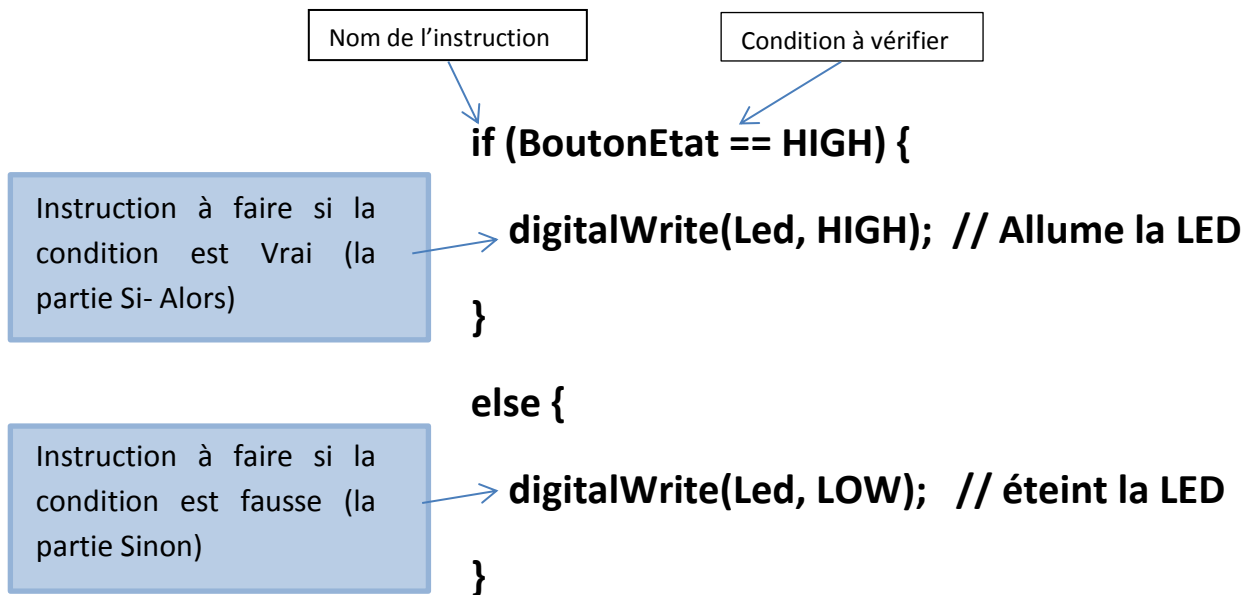
Non de la fonction permettant la lecture de l'entrée. Cette fonction prend en paramètre le numéro de l'entrée à lire

BoutonEtat = digitalRead(Bouton);

Variable d'enregistrement des valeurs de retour de la fonction digitalRead();

Paramètre : Numéro du pin à définir celui-ci peut aussi être inclut dans une variable

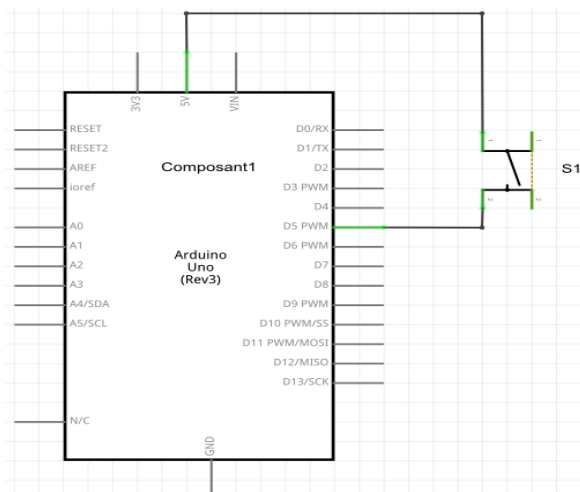
Comme vous pouvez le voir la fonction digitalRead() à un élément qui diffère de la fonction vue précédemment digitalWrite() et cet élément est le fait qu'elle revoie une information une fois le traitement terminé. Cette information une information de type booléen (soit HIGH ou LOW). Le traitement de cette information est effectué à la suite par une structure de contrôle de if – else (SI – Alors – Sinon).

LA STRUCTURE DE CONTROLE (IF – ELSE):**Forme de la structure de contrôle :**

⚠ Une erreur très fréquente, est de confondre l'opérateur d'égalité et l'opérateur d'affectation. L'opérateur d'égalité est « == » et l'opérateur d'affectation est = ont des missions complètement différentes, mais sont souvent utilisés l'un à la place de l'autre.

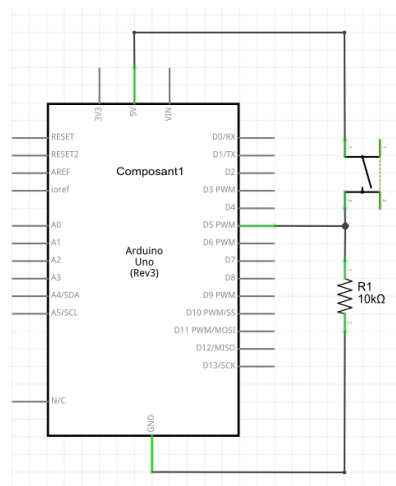
EXPLICATION SCHEMA**RESISTANCE DE TIRAGE :**

Un des problèmes est la gestion du bouton-poussoir qui est relié d'un côté au +5V et de l'autre côté à l'arduino. Le circuit tel que vous le voyez ici, ne fonctionne pas vous l'auriez peut être imaginé. Si une entrée n'est pas alimentée par aucun niveau défini sous une forme HIGH ou LOW, le comportement dépend de facteurs divers venant des parasites. Ce qui donne un circuit non stable pour remédier à ce problème, il existe une des solutions qui est la résistance pull-down. Cette dernière tire littéralement le niveau ou plutôt le potentiel vers le bas. Comme un courant passe également par cette résistance, celle-ci doit être relativement forte.



Le schéma ci-contre montre cette résistance, qui tire la broche 5 de l'arduino à travers ses $10k\Omega$ vers la masse quand le bouton poussoir n'est pas enfoncé.

Quand le bouton-poussoir est relâché, l'entrée numérique a donc un niveau LOW défini et reconnu par le microcontrôleur. Si par contre, le bouton-poussoir est enfoncé, la résistance fait chuter les +5V de la tension de service. Celle-ci est appliquée directement à la broche 5, qui est alors au niveau HIGH défini.



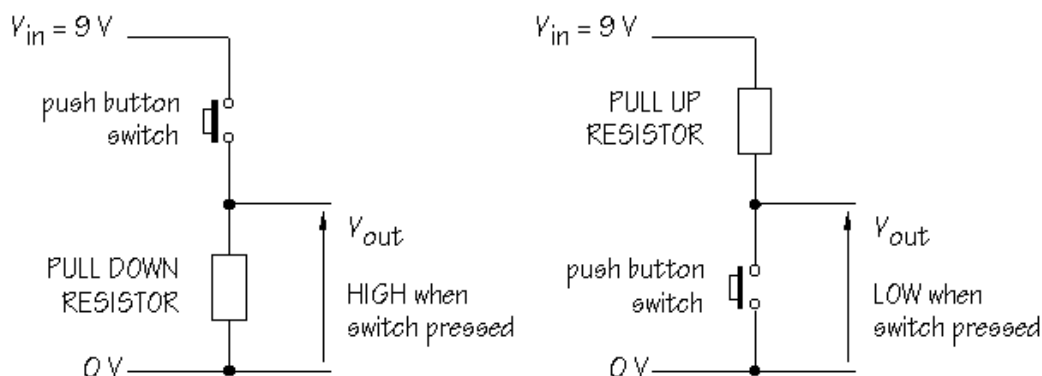
CHRONOLOGIE DE LA SEANCE

Toutes les étapes énoncées par la suite doivent être réalisées par les apprenants.

1. Présentation par l'intervenant sur un arduino, le résultat attendu pour la fin de la séance.
2. Définir le fonctionnement du programme que l'on souhaite faire dans la boucle « loop »,
 - Allumer la DEL lorsque l'on appui sur le bouton-poussoir
3. Laisser les apprenants réaliser leur programme en les assistant lorsqu'ils sont bloqués.
4. Vérifier le programme avant de le transférer sur la carte.
5. Pour les plus rapide demander de faire clignoter la DEL lorsque l'on appui sur le bouton-poussoir.

ASTUCE

Voici d'autres possibilités pour des niveaux d'entrée définis :



Etat du Bouton-poussoir	Potentiel de la broche	
	0 V (masse, niveau LOW)	+ 5 V (tension d'alimentation, niveau HIGH)
Ouvert		
Fermé	+ 5 V (tension d'alimentation, niveau HIGH)	0 V (masse, niveau LOW)

Il est possible d'utiliser la résistance pull-up du microcontrôleur, Une résistance pull-down ou pull-up séparée est en fait superflue car votre microcontrôleur dispose déjà de résistances pull-up internes incorporées aux broches numériques, qui peuvent être également mises en service par le logiciel le cas échéant. On peut mis en place grâce à ces instructions :

```
pinMode(pin, INPUT);           // Programmer la broche comme entrée
digitalWrite(pin, HIGH);        // Activer la résistance pull-up interne
```