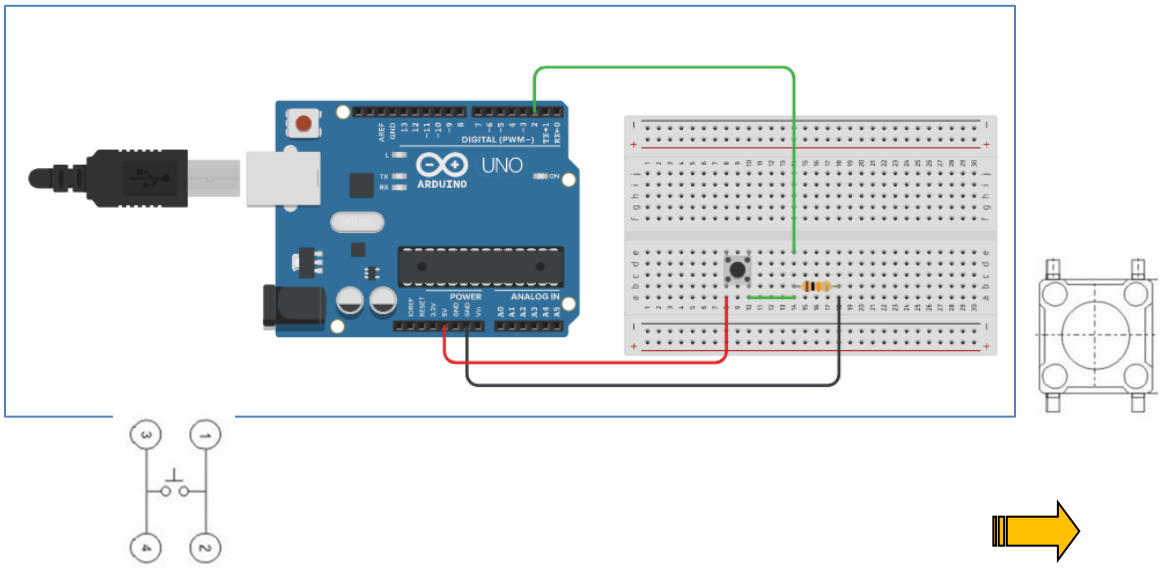
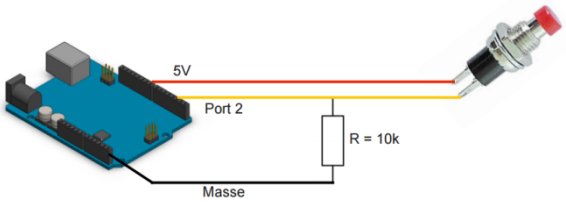
Affichage de l'état d'un bouton poussoir, sur la console



Résistance 10kΩ



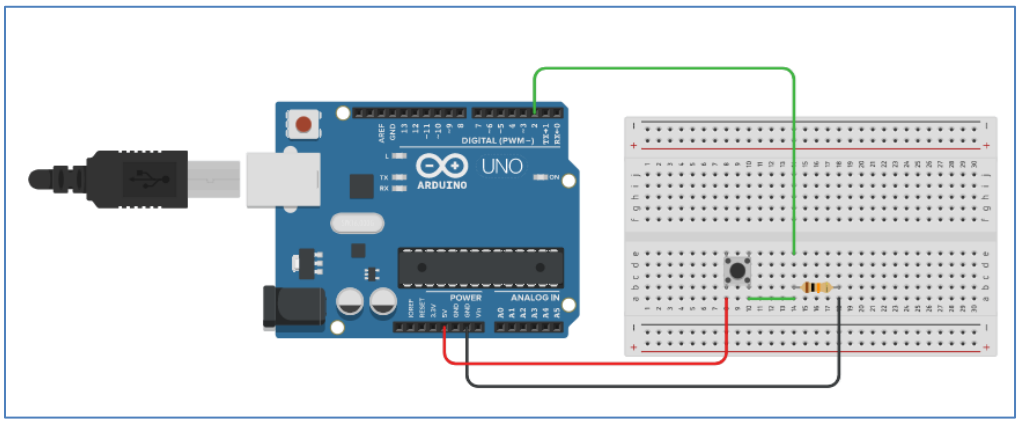
`int etat_bouton;`
 	
`void setup() {`
`  Serial.begin(9600);`
`  pinMode(2, INPUT);`
`  etat_bouton=0;`
`}`
 	
`void loop() {`
`  etat_bouton = digitalRead(2);`
`  Serial.println(etat_bouton, DEC);`
`}`
		Définit la variable "etat_bouton" comme un entier

		Affiche l'état de la variable dans le terminal série
		Initialise la communication série
		Lit l'état de la broche 2 et met le résultat dans la variable
		Initialise la broche 2 comme une entrée

Pour tester le programme virtuellement : https://www.tinkercad.com/

https://www.tinkercad.com/

Affichage du nombre d'impulsions sur un bouton poussoir, sur la console



R=10kΩ

`boolean etat_bouton;` `int bp_counter;`		Déclaration des variables

`void setup() {`
`  Serial.begin(9600);`
`  pinMode(2, INPUT);`
`  etat_bouton=0;`
`  bp_counter=0;`
`}`
		Initialisation de la communication
		Lecture de l'état de la broche 2
		Initialisation de la valeur des variables

`void loop() {` `  etat_bouton = digitalRead(2);`		Test de la variation de l'état de la variable « etat_bouton »

`  if(etat_bouton>>0)`
`  {bp_counter++;`
`  }`
		Incrémentation de la variable « bp_counter »

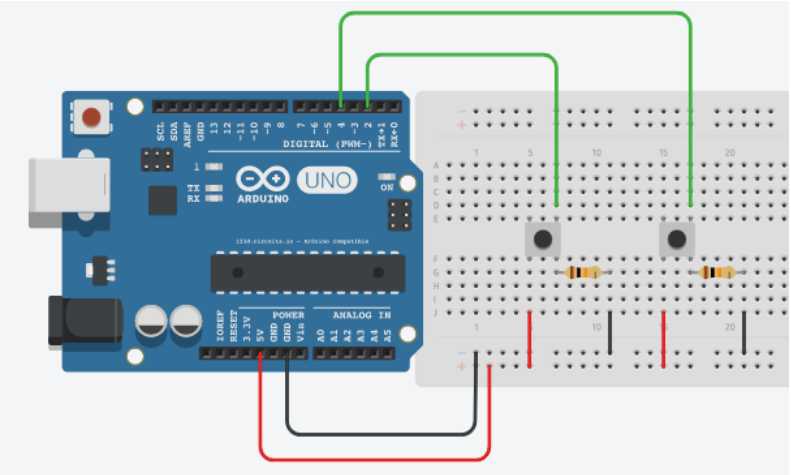
`  Serial.println("nombre d'impulsions:");` `  Serial.println(bp_counter);`		Affichage à l'écran du nombre d'impulsions
`  delay(500);` `}`		Affichage à l'écran des mots « nombre d'impulsions : »

++ : Signifie incrémenter (ajouter 1 point)

>> : Signifie très supérieur à

Problème : si on reste appuyé sur le bouton, le compteur incrémente le score toutes les 500ms

Affichage d'un nombre d'impulsions sur l'un, l'autre ou les deux boutons poussoir, sur la console



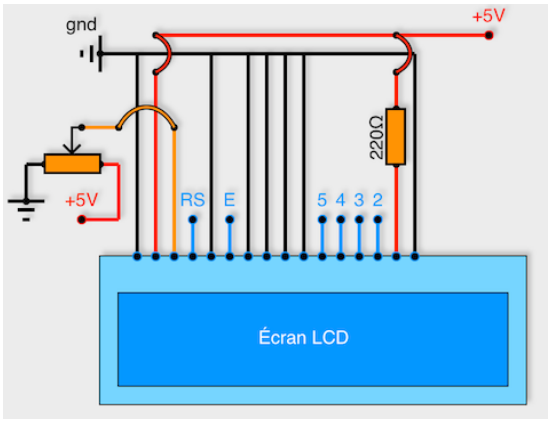
Résistances 10kΩ

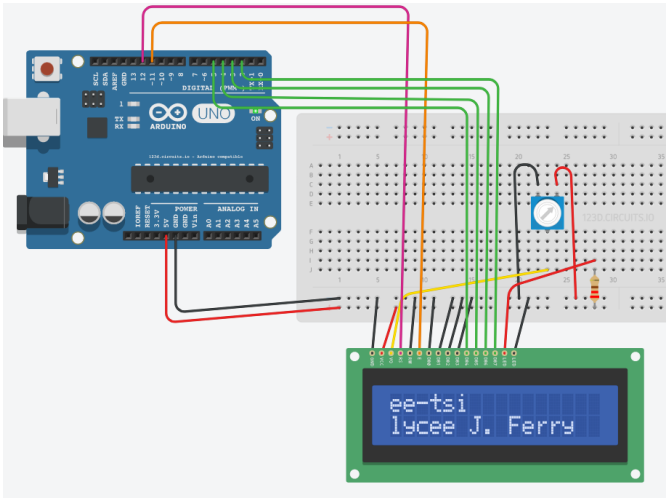
```
int bp1=2;
int bp2=4;
int bp_counter;
boolean bp1_etat_actuel;
boolean bp1_etat_precedent;
boolean bp2_etat_actuel;
boolean bp2_etat_precedent;

void setup() {
  Serial.begin(9600);
  pinMode(bp1, INPUT);
  pinMode(bp2, INPUT);
  bp1_etat_actuel=0;
  bp1_etat_precedent=0;
  bp2_etat_actuel=0;
  bp2_etat_precedent=0;
}

void loop() {
  bp1_etat_actuel = digitalRead(bp1);
  bp2_etat_actuel = digitalRead(bp2);

  if((bp1_etat_actuel!=bp1_etat_precedent)||(bp2_etat_actuel!=bp2_etat_precedent))

  {bp_counter++;
  }
  Serial.println("Nombre d'impulsions:");
  Serial.println(bp_counter);
  delay(500);
}
```

Problème : ça incrémente lorsque l'on passe de 0 à 1 et lorsque l'on passe de 1 à 0.

Câblage d'un écran LCD

LCD est l'abbréviation anglaise de "liquid crystal display" qui veut dire : afficheur à cristaux liquides. Cette technologie permet de créer des écrans plats qui consomment peu d'énergie.

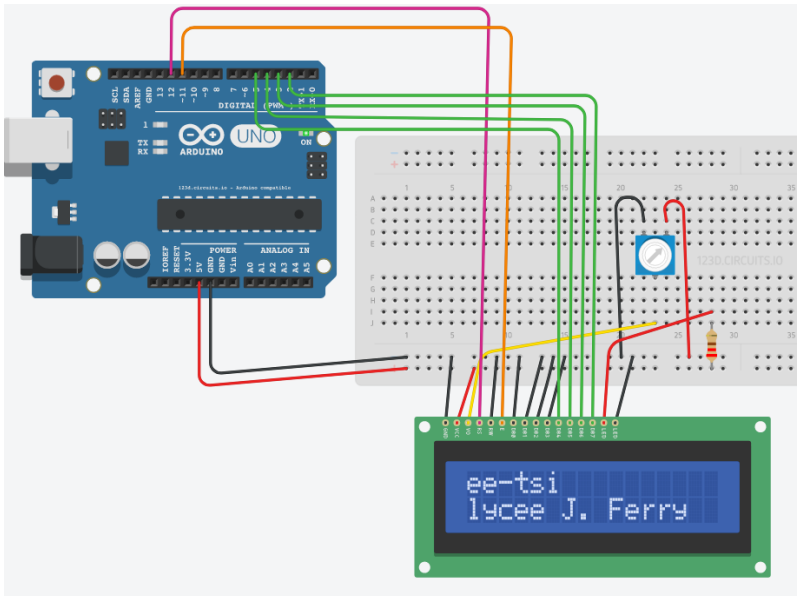




En partant de la gauche, voici à quoi servent les broches, aussi appelées "pins":

- Les deux premiers pins tout à gauche servent à l'alimentation de l'écran.
- Le troisième pin est connecté à un potentiomètre et sert pour régler l'affichage (le contraste de l'écran).
- Le quatrième, noté RS, est connecté au pin 12 de l'Arduino dans notre exemple. Il sert à sélectionner la zone mémoire de l'écran LCD dans laquelle nous allons écrire (Register Select).
- Le cinquième doit toujours être connecté au ground. C'est un sélecteur de mode lecture ou écriture. On peut le connecter à un pin, mais dans notre cas c'est inutile. Comme il doit recevoir un signal à 0V, on le connecte au ground (état R/W).
- Le sixième, noté E, est connecté au pin 11 de l'Arduino dans notre exemple. Il permet de lancer ou non l'écriture dans les zones mémoires (Enable).
- Les quatre suivants (reliés au ground) servent pour la communication 8 bits. Pour la communication 4 bits, il est conseillé de les relier au ground. Ils représentent les bits de poids fort.
- Les quatre qui suivent, notés 2, 3, 4, 5, se connectent dans notre exemple sur les pins 2, 3, 4, 5 de l'Arduino. Ils servent pour la communication (8 bits ou 4 bits) et doivent toujours être connectés. Ils représentent les bits de poids faible (ou servent pour envoyer d'abord les bits de poids faible, puis les bits de poids fort)
- Les deux pins tout à droite servent pour alimenter la LED du rétro-éclairage.

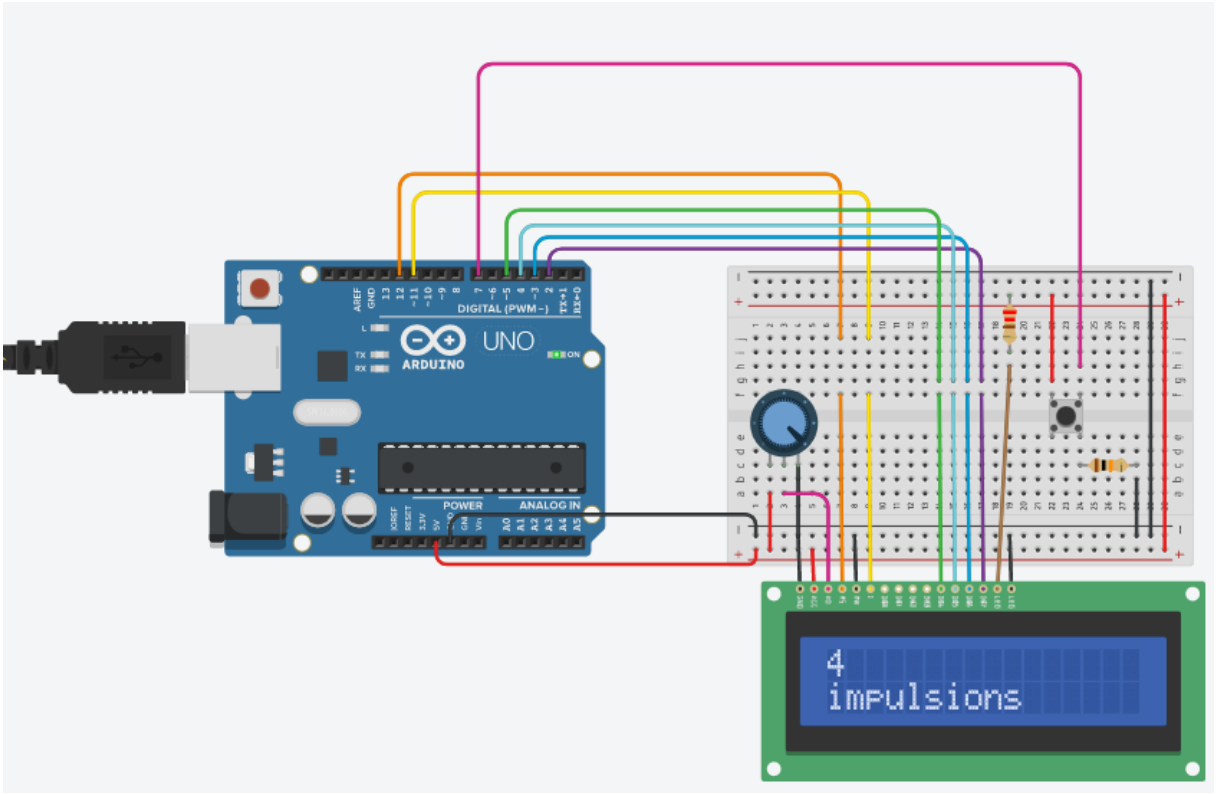
Affichage d'un message




```
#include <LiquidCrystal.h>                    1
LiquidCrystal monEcran(12,11,5,4,3,2);        2


void setup() {
  monEcran.begin(16,2);                       3
  monEcran.clear();                           4
}


void loop() {
  monEcran.print("ee-tsi");                   5
  delay(1000);
  monEcran.setCursor(0,1);                    6
  monEcran.print("lycee J. Ferry");
  delay(1000);                                7
  monEcran.clear();
}
```

	Attendre 1 seconde
	Déplacer le curseur
	Ecrire sur l'écran
	Utilisation de la bibliothèque "LiquidCrystal"
	Effacer l'écran
	Indiquer le nombre de lignes et de colonnes de l'écran
	Indiquer les broches pour le branchement de l'écran

Affichage du nombre d'impulsions sur l'écran

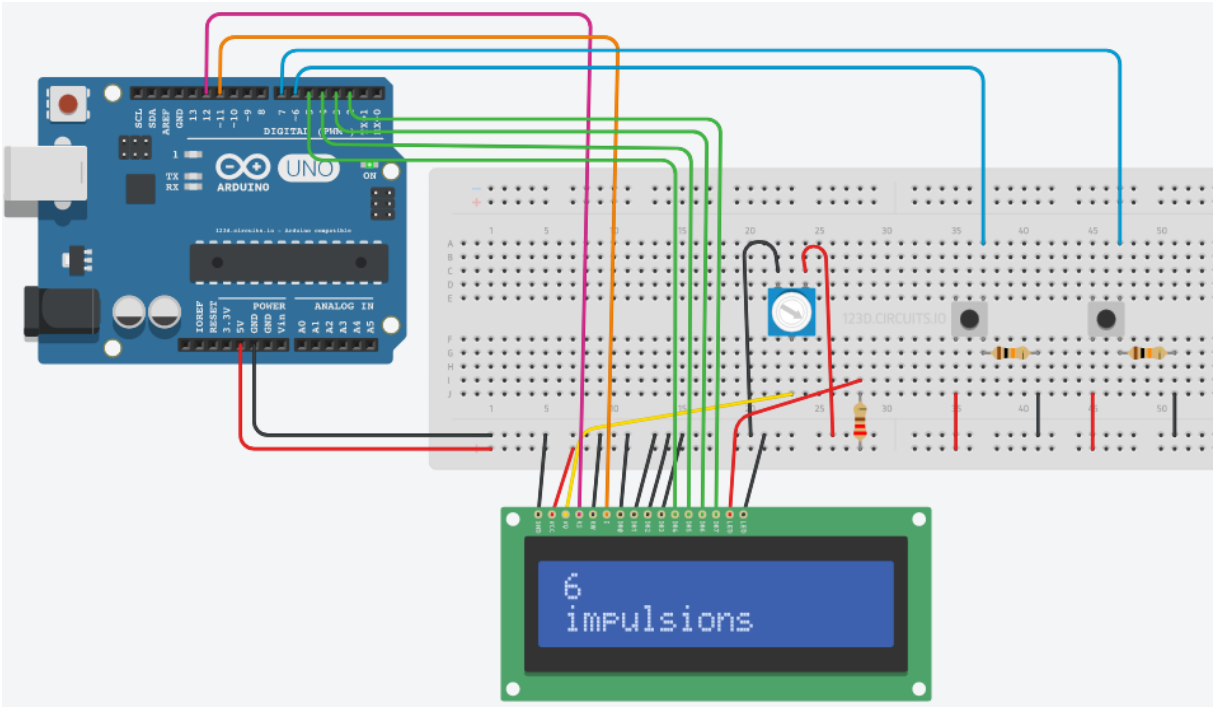


Les broches 15 et 16 n'existant pas sur mon écran (pas de rétroéclairage

Inutile de relier les 4 bits de poids fort à la masse (cf image ci-dessus)

```
#include <LiquidCrystal.h>
LiquidCrystal monEcran(12,11,5,4,3,2);
int bpl_etat_actuel = 0;
int bpl_etat_precedent = 0;
int bp_counter = 0;

void setup() {
  monEcran.begin(16,2);
  monEcran.clear();
  pinMode(7, INPUT);
}

void loop() {
  bpl_etat_actuel = digitalRead(7);

  if (bpl_etat_actuel != bpl_etat_precedent) {
     if (bpl_etat_actuel == HIGH) {
      bp_counter++;
      monEcran.print(bp_counter);
      monEcran.setCursor(0,1);
      monEcran.print("impulsions");
      monEcran.setCursor(0,0);
    }

    delay(50); // Wait for 5 millisecond(s)
  }
  bpl_etat_precedent = bpl_etat_actuel;
}
```

Affichage du nombre d'impulsions sur l'écran



```
#include <LiquidCrystal.h>
LiquidCrystal monEcran(12,11,5,4,3,2);
int bp1 = 6;
int bp2 = 7;
int bp_counter;
boolean bp1_etat_actuel;
boolean bp1_etat_precedent;
boolean bp2_etat_actuel;
boolean bp2_etat_precedent;

void setup() {
  monEcran.begin(16,2);
  monEcran.clear();
  pinMode(bp1, OUTPUT);
  pinMode(bp2, OUTPUT);
  Serial.begin(9600);
  bp_counter=0;
  bp1_etat_actuel=0;
  bp1_etat_precedent=0;
  bp2_etat_actuel=0;
  bp2_etat_precedent=0;
}

void loop() {
  bp1_etat_actuel=digitalRead(bp1);
  bp2_etat_actuel=digitalRead(bp2);

  if ((bp1_etat_actuel!=bp1_etat_precedent)||(bp2_etat_actuel!=bp2_etat_precedent))
  {
    bp_counter++;
    monEcran.setCursor(0,0);
    monEcran.print(bp_counter);
    monEcran.setCursor(0,1);
    monEcran.print("impulsions");
    Serial.println(bp_counter);
    delay(500);
  }
}
```