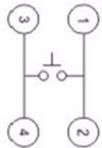
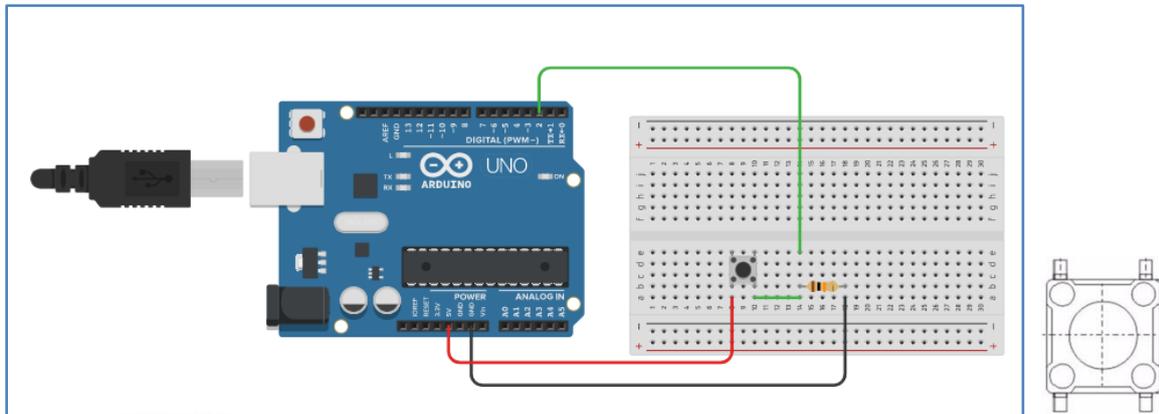
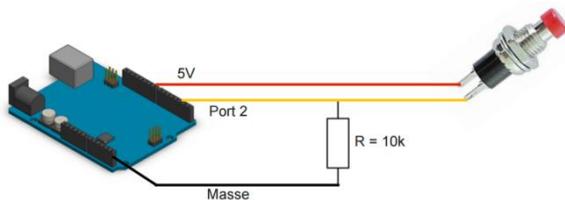


### Affichage de l'état d'un bouton poussoir, sur la console



Résistance 10kΩ

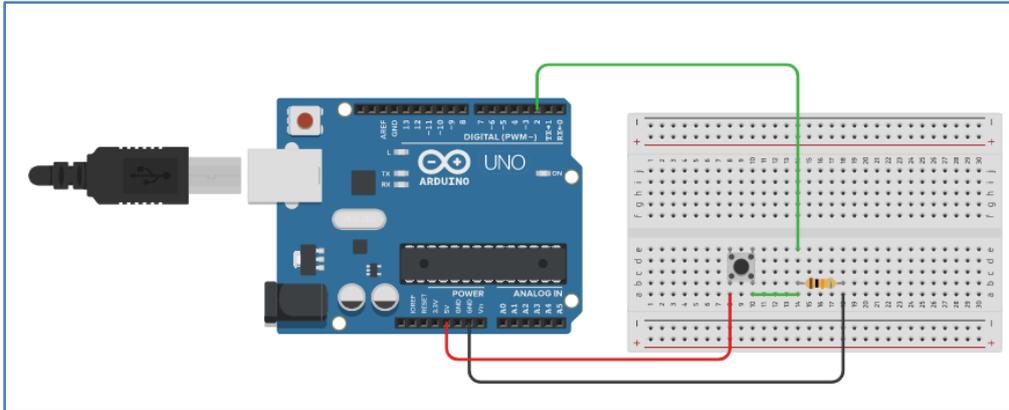


<pre>int etat_bouton;  void setup() {   Serial.begin(9600);   pinMode(2, INPUT);   etat_bouton=0; }  void loop() {   etat_bouton = digitalRead(2);   Serial.println(etat_bouton, DEC); }</pre>		Définit la variable "etat_bouton" comme un entier
		Affiche l'état de la variable dans le terminal série
		Initialise 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

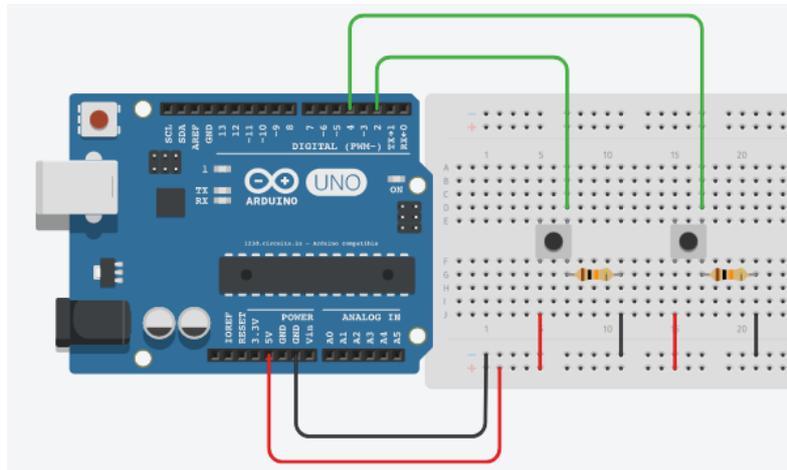
<pre> boolean etat_bouton; int bp_counter;  void setup() {   Serial.begin(9600);   pinMode(2, INPUT);   etat_bouton=0;   bp_counter=0; }  void loop() {   etat_bouton = digitalRead(2);    if(etat_bouton&gt;&gt;0)   {bp_counter++;   }   Serial.println("nombre d'impulsions:");   Serial.println(bp_counter);   delay(500); } </pre>		Déclaration des variables
		Initialisation de la communication
		Lecture de l'état de la broche 2
		Initialisation de la valeur des variables
		Test de la variation de l'état de la variable « etat_bouton »
		Incrémentation de la variable « bp_counter »
		Affichage à l'écran du nombre d'impulsions
		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;
}

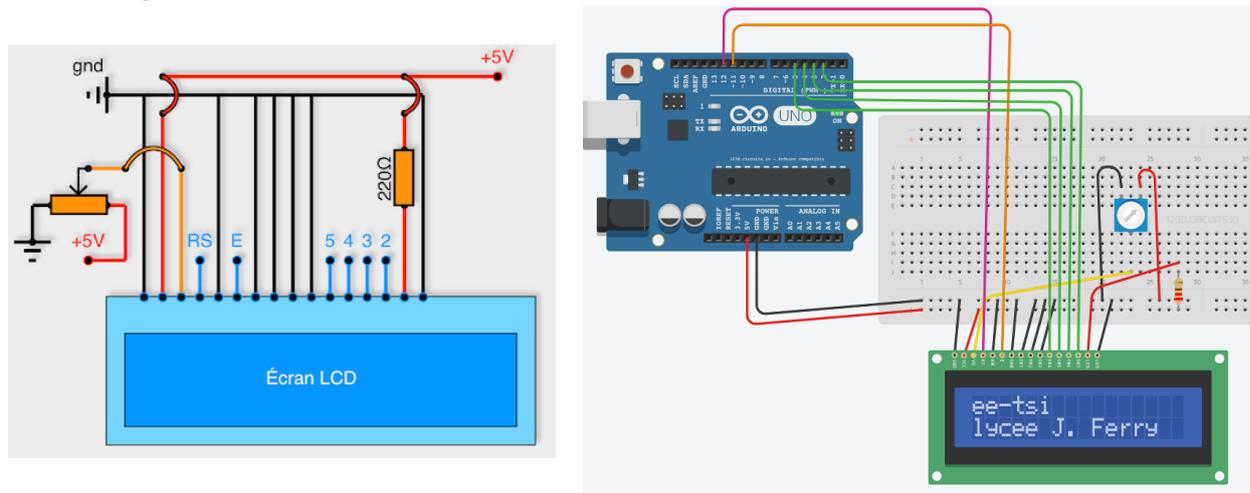
void loop() {
  bp1_etat_actuel = digitalRead(bp1);
  bp2_etat_actuel = digitalRead(bp2);

  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

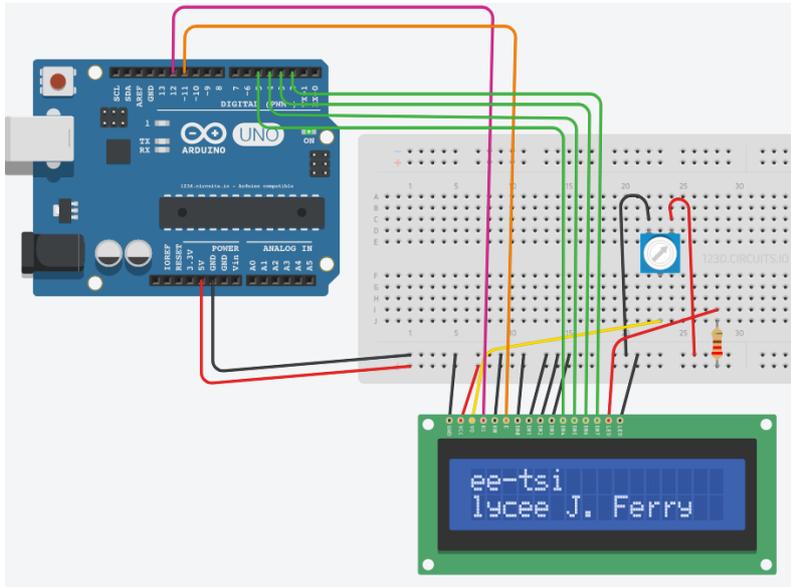
LCD est l'abréviation anglaise de "liquid crystal display" qui veut dire : afficheur à cristaux liquides. Cette technologie permet de créer des écrans plats qui consomment peu d'énergie.



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

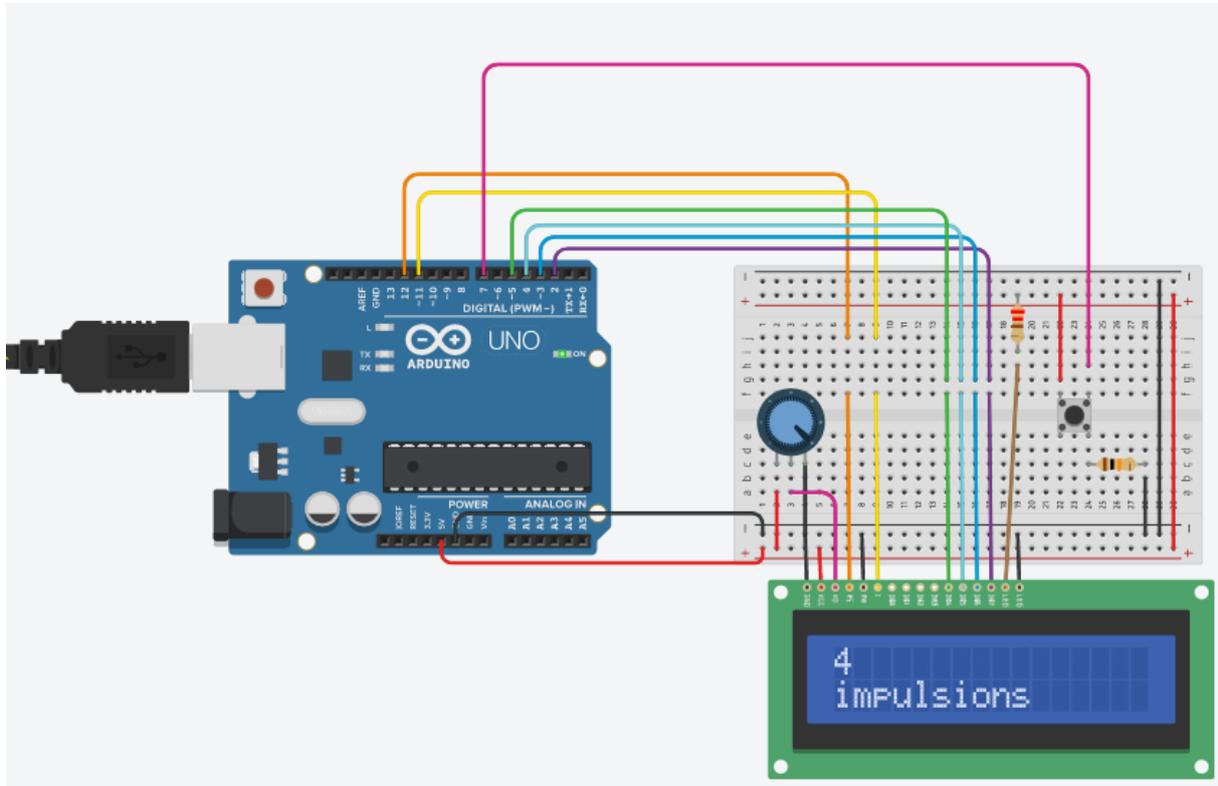
- Les deux premiers pins tout à gauche servent à l'alimentation de l'écran.
- Le troisième pin est connecté à un potentiomètre et sert pour régler l'affichage (le contraste de l'écran).
- Le quatrième, noté RS, est connecté au pin 12 de l'Arduino dans notre exemple. Il sert à sélectionner la zone mémoire 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



<pre>#include &lt;LiquidCrystal.h&gt; LiquidCrystal monEcran(12,11,5,4,3,2);</pre>	<p>1</p> <p>2</p>	<p>Attendre 1 seconde</p>
<pre>void setup() {   monEcran.begin(16,2);</pre>	<p>3</p>	<p>Déplacer le curseur</p>
<pre>  monEcran.clear(); }</pre>	<p>4</p>	<p>Ecrire sur l'écran</p>
<pre>void loop() {   monEcran.print("ee-tsi");</pre>	<p>5</p>	<p>Utilisation de la bibliothèque "LiquidCrystal"</p>
<pre>  delay(1000);   monEcran.setCursor(0,1);</pre>	<p>6</p>	<p>Effacer l'écran</p>
<pre>  monEcran.print("lycee J. Ferry");   delay(1000);</pre>	<p>7</p>	<p>Indiquer le nombre de lignes et de colonnes de l'écran</p>
<pre>  monEcran.clear(); }</pre>		<p>Indiquer les broches pour le branchement de l'écran</p>

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)



S si

# Programmation

## Carte ARDUINO



TP

```
#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;

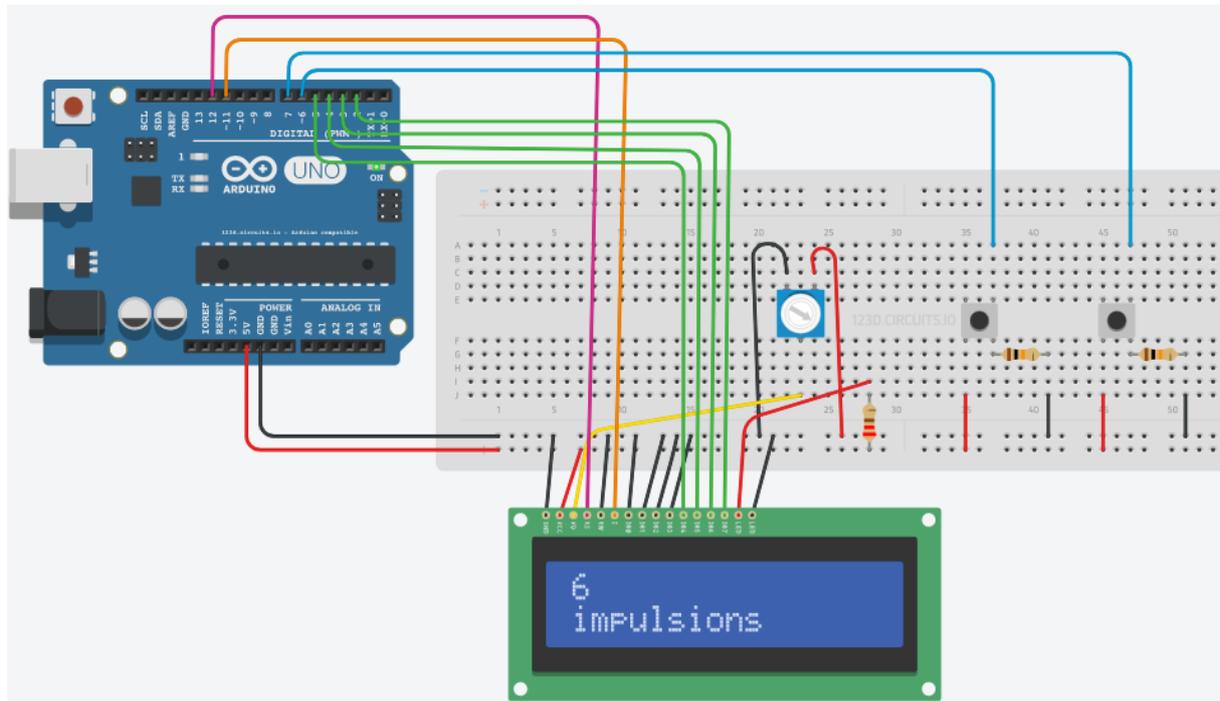
void setup() {
  monEcran.begin(16,2);
  monEcran.clear();
  pinMode(7, INPUT);
}

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





S si

# Programmation

## Carte ARDUINO



TP

```
#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);
  }
}
```