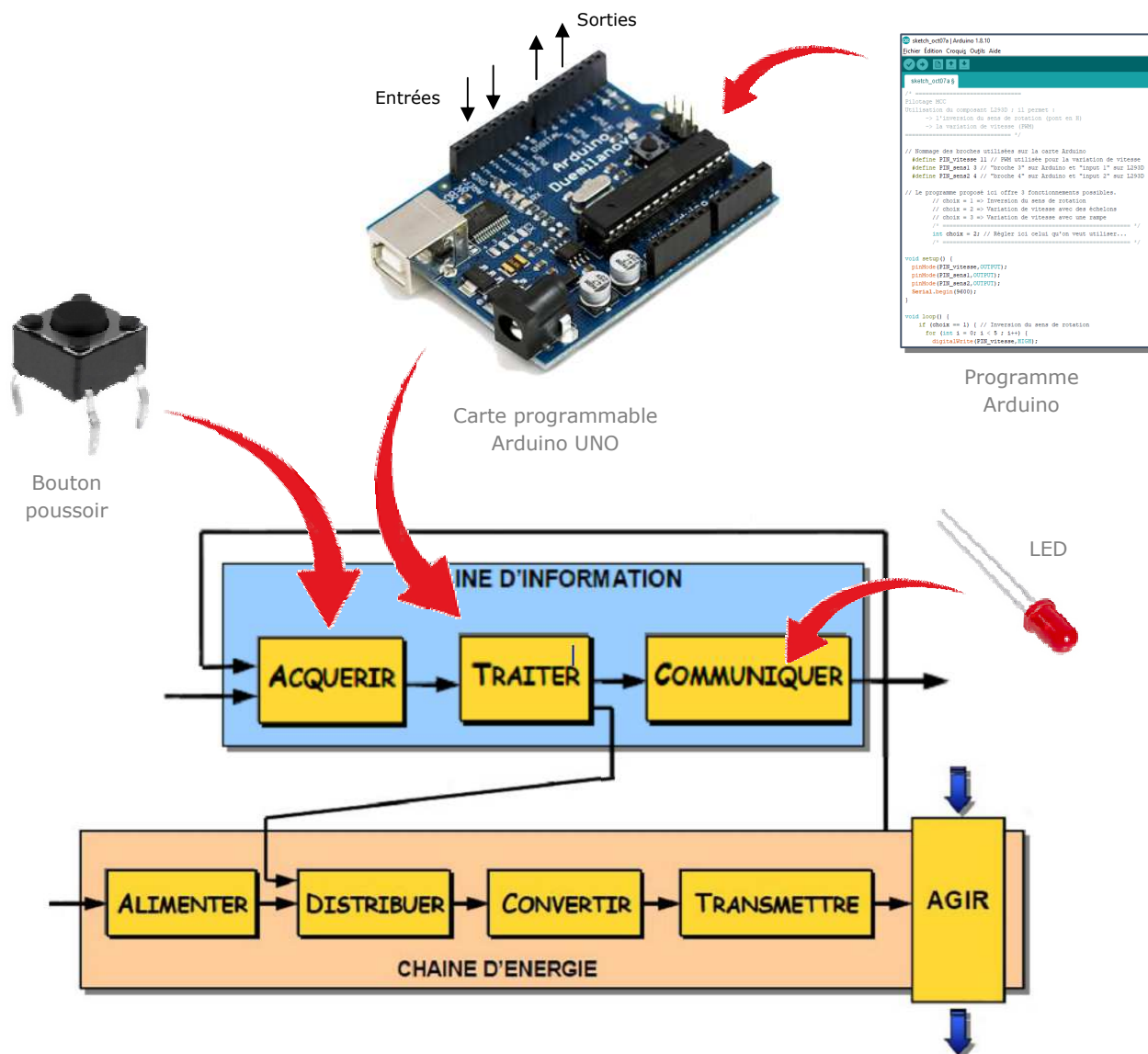




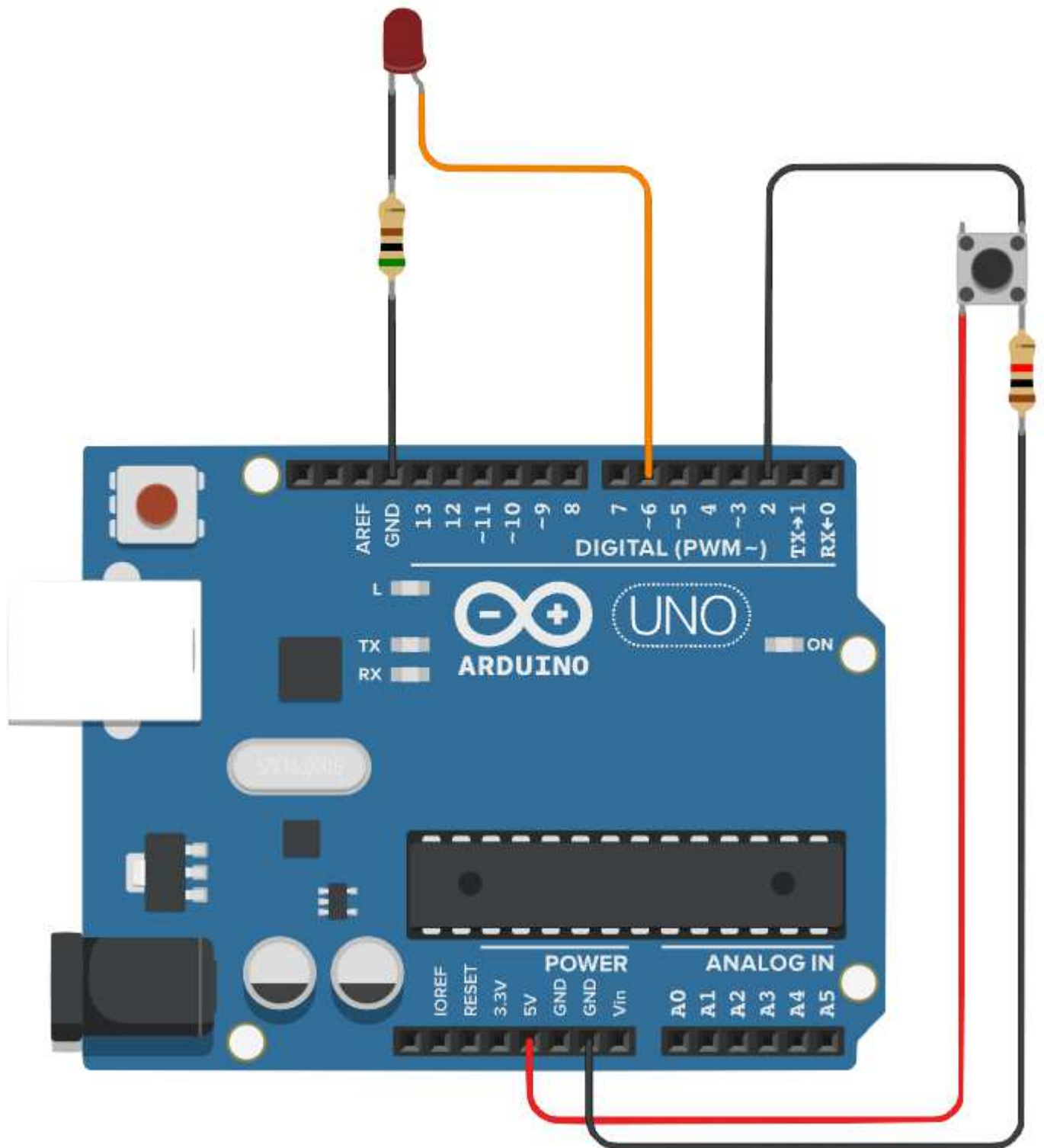
# MISE EN ŒUVRE

- **TRAITER** : ARDUINO UNO (EDI Arduino)
- **AQUERIR** : Bouton poussoir
- **COMMUNIQUER** : LED

## 1 – Mise en situation



## 2 – Plan de câblage / Montage



## 3 – Programmes

### 📌 PROGRAMME 1 : « LED allumée si appui.ino »

Ce programme permet d'allumer la LED quand on appui sur le bouton poussoir. La LED s'éteint quand cesse l'appui sur le bouton poussoir.

L'allumage et l'extinction sont « tout ou rien ».

📌 Bibliothèques requises : aucune

📌 Utilisation de la MLI : NON

```
1 // =====
2 // PILOTAGE D'UNE LED AVEC UN BOUTON POUSSOIR
3 // =====
4 /*
5 - Allumage et extinction "Tout Ou Rien" (TOR) de la LED
6 - Un appui sur le BP allume la LED
7 - Si on relâche le BP, la LED s'éteint
8 */
9
10 // Déclaration des variables "programme"
11 int LED = 6;
12 int BUTTON = 2;
13
14 // Déclaration des paramètres "utilisateur"
15 // Aucun
16
17 void setup() {
18     pinMode(BUTTON, INPUT);
19     pinMode(LED, OUTPUT);
20     digitalWrite(LED, LOW); // état initial de la LED : éteinte
21 }
22
23 void loop() {
24     if (digitalRead(BUTTON) == 1) {
25         digitalWrite(LED, HIGH);
26     } else {
27         digitalWrite(LED, LOW);
28     }
29     delay(1); // pour la stabilité
30 }
31
```

## 📌 PROGRAMME 2 : « LED – BP – Variation.ino »

Ce programme permet d'allumer la LED quand on appui sur le bouton poussoir. La LED s'éteint quand cesse l'appui sur le bouton poussoir.

L'allumage et l'extinction sont progressifs.

📌 Bibliothèques requises : aucune

📌 Utilisation de la MLI : OUI

```
1 // =====
2 // PILOTAGE D'UNE LED AVEC UN BOUTON POUSSOIR
3 // =====
4 /*
5 - Allumage et extinction progressif de la LED
6 - La LED doit être branchée sur une sortie MLI
7 - A noter : une simple impulsion sur le BP amènera la LED
8   à s'allumer complètement
9 - Un autre fonctionnement est possible avec les interruptions...
10 */
11
12 // Déclaration des variables "programme"
13 int LED = 6;
14 int BUTTON = 2;
15 int cpt1 = 0; // Variable pour incrémentation / décrémentation
16 int cpt2 = 0; // Variable pour incrémentation / décrémentation
17
18 // Déclaration des paramètres "utilisateur"
19 // Aucun
20
21 void setup() {
22   pinMode(BUTTON, INPUT);
23   pinMode(LED, OUTPUT);
24   digitalWrite(LED, LOW); // état initial de la LED : éteinte
25 }
26
27 void loop() {
28   if(digitalRead(BUTTON) == 1 && cpt1 == 0) {
29     cpt1++;
30     for (cpt2 = 0 ; cpt2 < 150 ; cpt2++) {
31       analogWrite(LED, cpt2);
32       delay(10);
33     }
34   }
35   if (digitalRead(BUTTON) == 0 && cpt1 == 1) {
36     cpt1--;
37     for (cpt2 = 150 ; cpt2 > 0 ; cpt2--) {
38       analogWrite(LED, cpt2);
39       delay(10);
40     }
41   }
42 }
```

### 📌 PROGRAMME 3 : « LED – BP – Variation parametres.ino »

Ce programme est identique au précédent : il permet d'allumer la LED quand on appui sur le bouton poussoir et La LED s'éteint quand cesse l'appui sur le bouton poussoir

L'allumage et l'extinction sont progressifs.

Sont paramétrés :

- Les seuils minimum et maximum de luminosité, (de 0 à 255)
- les durées d'allumage et d'extinction. (en s)

📌 Bibliothèques requises : aucune

📌 Utilisation de la MLI : OUI

```
1 // =====
2 // PILOTAGE D'UNE LED AVEC UN BOUTON POUSSOIR
3 // =====
4 /*
5 - Allumage et extinction progressif de la LED
6 - La LED doit être branchée sur une sortie MLI
7 - A noter : une simple impulsion sur le BP amènera la LED
8   à s'allumer complètement
9 - Un autre fonctionnement est possible avec les interruptions...
10 */
11
12 // Déclaration des variables "programme"
13   int LED = 6;
14   int BUTTON = 2;
15   bool Etat = 0; // état logique de la LED (0 = éteinte ; 1 = allumée)
16   int cpt = 0; // Variable pour incrémentation / décrémentation
17   int pause1 = 0; // intensité minimale (0 à 255)
18   int pause2 = 150; // intensité maximale (0 à 255)
19
20 // Déclaration des paramètres "utilisateur"
21   int Imin = 0; // intensité minimale (0 à 255)
22   int Imax = 200; // intensité maximale (0 à 255)
23   // Veiller à avoir Imin < Imax
24   int duree1 = 2; // durée en s de l'allumage (passage de Imin à Imax)
25   int duree2 = 3; // durée en s de l'extinction (passage de Imax à Imin)
26
27 void setup() {
28   pinMode(BUTTON, INPUT);
29   pinMode(LED, OUTPUT);
30   digitalWrite(LED, LOW); // Etat initial de la LED : éteinte
31   // Calcul de la pause en ms entre chaque itération pour l'allumage
32   pause1 = 1000 * duree1 / (Imax - Imin);
33   // Calcul de la pause en ms entre chaque itération pour l'extinction
34   pause2 = 1000 * duree2 / (Imax - Imin);
35 }
```

```
36
37 void loop() {
38     if(digitalRead(BUTTON) == 1 && Etat == 0) {
39         Etat = 1;
40         for (cpt = Imin ; cpt < Imax ; cpt++) {
41             analogWrite(LED, cpt);
42             delay(pause1);
43         }
44     }
45     if (digitalRead(BUTTON) == 0 && Etat == 1) {
46         Etat = 0;
47         for (cpt = Imax ; cpt > Imin ; cpt--) {
48             analogWrite(LED, cpt);
49             delay(pause2);
50         }
51     }
52 }
53
```