



Fiche : Manipulation Arduino

Table des matières

1	Introduction	1
2	Matériel nécessaire	1
3	Utilisation d'Arduino pour l'allumage d'une DEL RGB	2
3.1	Montage expérimental	2
3.2	Le code	3
3.3	Utilisation en LP	4
4	Application à la mesure de la vitesse du son	4
4.1	Montage expérimental	4
4.2	Le code	4
4.3	Mesure et résultats	6

1 Introduction

Nous proposons de réaliser deux manipulations utilisant le boîtier Arduino. Elles peuvent être réalisées dans le cadre de leçons de physique. La première manipulation permettant de montrer la synthèse additive rentrerait dans la LP9 images et couleurs. L'autre permettant de remonter à la célérité du son pourrait rentrer dans les leçons traitant des ondes, sons ou de capteurs.

Ces petits boîtiers sont en train de devenir des outils important de pédagogie. En effet, ils permettent une approche peu couteuse et appliquée d'activités de codage et permettent d'illustrer les capacités d'électrocinétique en proposant un pannel varié d'activités. Lors de ces manipulations, les élèves devraient alors réaliser les montages électriques, comprendre les codes proposés et faire les applications. Il ne leur sera pas demandé de savoir coder à ce niveau.

Sources : <http://edu.mrpigg.ca/son.html>, https://dane.ac-nancy-metz.fr/programmation-au-lycee/?fbclid=IwAR370UjFJalAyE1lyTBjxaLf6RilCQk9DPX43srVyuLTxF3An-LL_FYYWHU, TP de M1 de Emma, Académie de Nantes.

2 Matériel nécessaire

Pour réaliser cette manipulation, il faut :

- Carte Arduino
- fils connecteurs
- fil de connexion USB pour liaison ordinateur - Arduino
- DEL
- Diode
- résistance 220 Ω

- Capteurs/émetteurs (HC-SRO4)
- Ordinateur avec le logiciel Arduino

3 Utilisation d'Arduino pour l'allumage d'une DEL RGB

Cette manipulation pourrait être utilisée dans le cadre de la synthèse additive dans la LP9.

3.1 Montage expérimental

Le montage expérimental est celui présenté dans la Fig. 1.

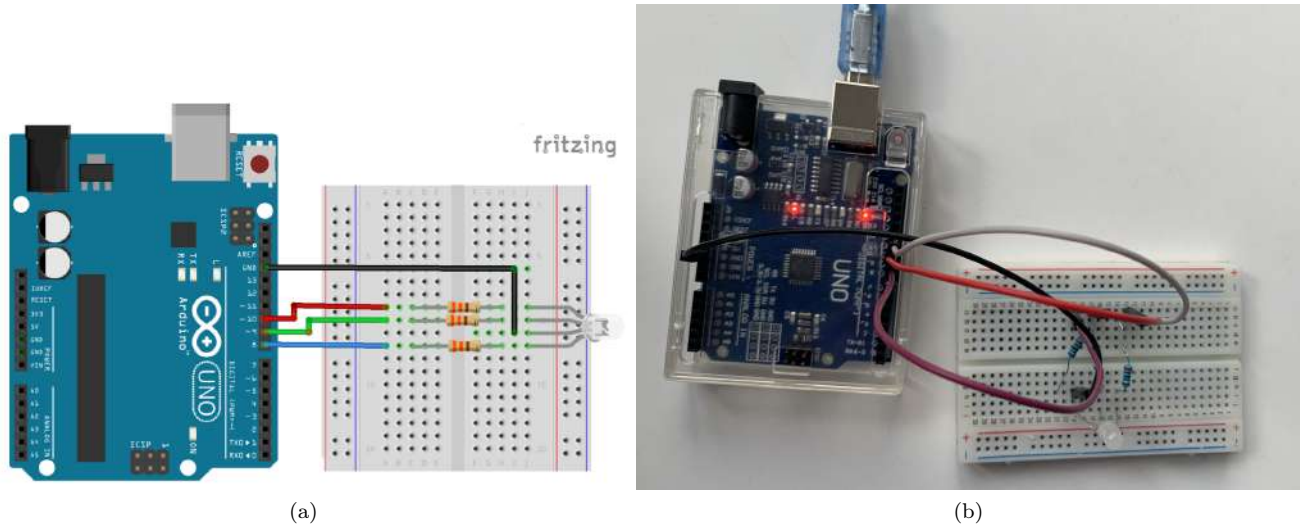


FIGURE 1 – (gauche) schéma du montage expérimental (droite) photographie du montage d'une diode à cathode commune

Sur la partie gauche on a la carte arduino. Pour la faire fonctionner, il faut brancher à l'ordinateur (il y a une alimentation mais on n'en a pas besoin ici). Il faut ouvrir l'application Arduino, taper ou mettre le code et finalement envoyer le code vers la carte. **Ne pas oublier de l'envoyer entre chaque manipulation**

Attention, le branchement de la DEL RGB est important. Il ne faut pas se tromper parce qu'il y a ici 3 diodes comme on peut le voir dans la Fig. 2. Les résistances permettent de ne pas griller les DEL.

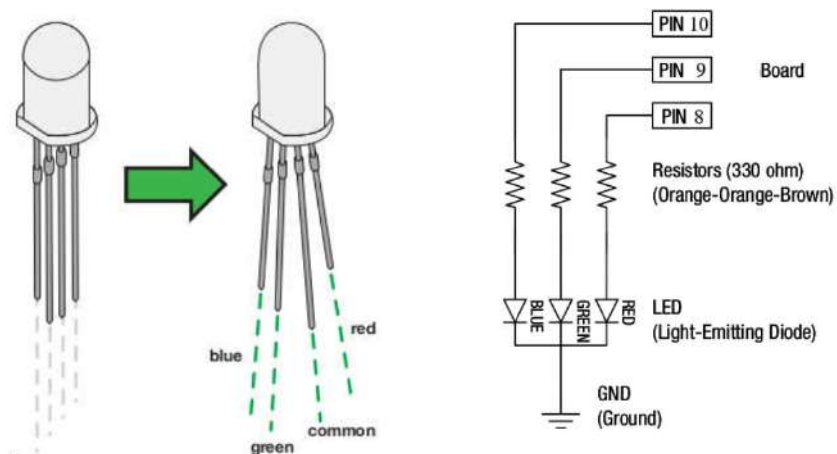


FIGURE 2 – (gauche) Schéma d'une diode (droite) schéma électrique d'une diode à cathode commune.

On voit bien sur le schéma, que la "pâte" la plus grande correspond au ground et que donc il faudra bien faire attention à comment on fait le circuit et le prendre en compte.

Pour une diode à **anode commune** le montage sera inversé. Le schéma de la diode à anode commune est présenté dans la Fig. 3.

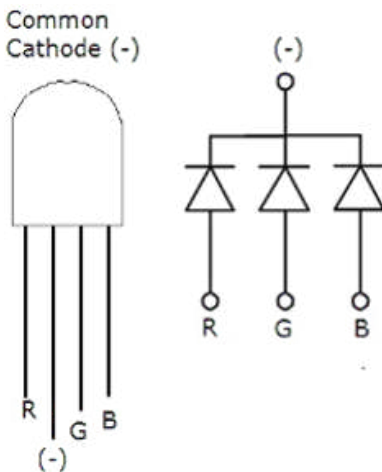


FIGURE 3 – schéma d'une diode RGB à anode commune

Le montage à réaliser dans ce cas est présenté dans la Fig. 4 :

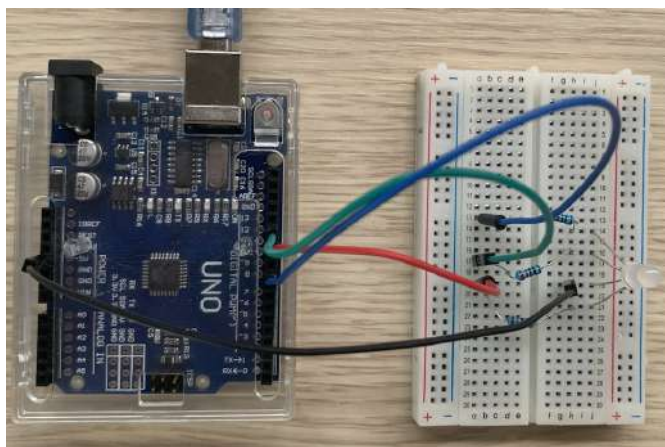


FIGURE 4 – Montage pour une diode RGB à anode commune

3.2 Le code

Arduino utilise un langage de programmation, le C++. Le fichier du code utilisé pour cette manipulation peut être téléchargé avec le lien suivant : [celui-ci](#). Celui-ci est également présenté dans la Fig. 5.

Le seul point un peu particulier de ce code repose sur les lignes 4 et 5. En effet, cette partie là repose sur le fait qu'on est avec une anode commune : par défaut on envoie du courant avec la borne 5V en continue. Pour éteindre il faut envoyer un signal d'où le 1. Dans la partie *Void setup()* on initialise les broches correspondant aux couleurs. Dans la partie *Void loop()* on exécute le code. On allume ou on éteint les diodes et on met un delay pour que ça clignote. Si on veut juste montrer la synthèse on peut couper après le premier **digitalWrite(ROUGE, OFF);** (en gardant l'accolade de fin).

```
int BLEU=8;
int VERT=12;
int ROUGE=11;
int ON = 0;
int OFF = 1;

void setup() {

  pinMode(ROUGE, OUTPUT);
  pinMode(BLEU, OUTPUT);
  pinMode(VERT, OUTPUT);
}

void loop() {
  digitalWrite(BLEU, ON);
  digitalWrite(VERT, OFF);
  digitalWrite(ROUGE, OFF);
  delay(2000);
  digitalWrite(BLEU, OFF);
  digitalWrite(VERT, ON);
  digitalWrite(ROUGE, ON);
  delay(2000);
}
```

FIGURE 5 – Code utilisé pour la DEL pour une **anode commune**

3.3 Utilisation en LP

On peut en jouant sur l'allumage des diodes RGB illustrer la synthèse additive et ainsi le montrer bien aux élèves.

4 Application à la mesure de la vitesse du son

Cette manipulation peut être faite dans toutes les leçons discutant d'ondes ou du son. On peut également l'utiliser pour capteurs.

4.1 Montage expérimental

Le montage expérimental est celui présenté dans la Fig. 6

On branche :

- GND du capteur sur le GND de l'arduino.
- VCC du capteur sur le 5 V de l'arduino.
- Trigg du capteur sur une sortie analogique de l'arduino.
- Echo du capteur sur une autre sortie analogique de l'arduino

Le capteur effectue la différence de temps entre le trigg et l'écho pour avoir le temps mis par l'onde pour faire l'aller-retour jusqu'à l'écran. Le temps est en microseconde.

4.2 Le code

Le code utilisé pour faire la mesure est celui présenté dans la Fig. 7. Commentons-le un peu :

- Les trois premières lignes non commentées correspondent à l'initialisation des entrées et de la variable délai.
- La partie Void setup correspond à l'initialisation des modes du capteur/émetteur et à l'ouverture d'un "terminal" pour afficher les mesures.

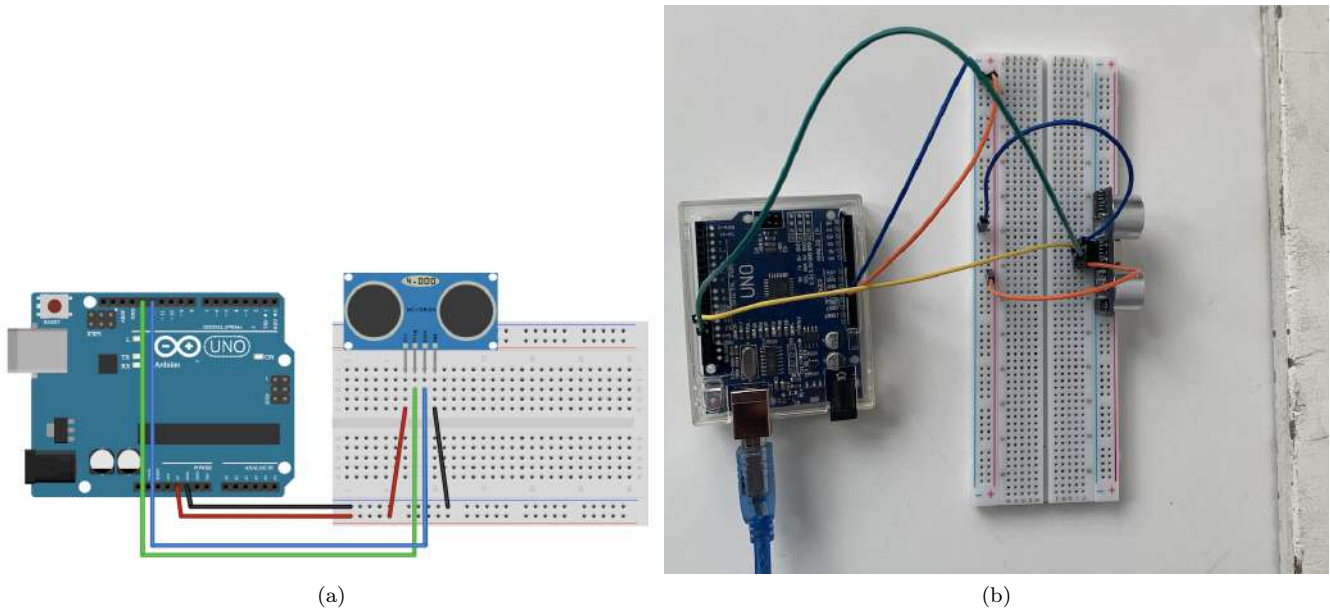


FIGURE 6 – (gauche) Schéma du montage expérimental <http://edu.mrpigg.ca/son.html> (droite) photographie du montage

```
//Calcul de la vitesse du son avec Arduino
//Affichage du temps de reception d'un écho

int emetteur = 13; //branché sur trig du HC-SR04
int recepteur = 12; //branché sur echo du HC-SR04
long dureeEcho;    //variable type long

void setup() {
  Serial.begin (9600);
  pinMode(emetteur, OUTPUT);
  pinMode(recepteur, INPUT);
}

void loop() {
  //Émission d'un pulse ultrason sur la pin 13: emetteur allumé pour 10 microsecondes
  digitalWrite(emetteur, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(emetteur, LOW);

  //Mesure du temps de reception de l'écho sur la pin 12
  dureeEcho = pulseIn(recepteur, HIGH);

  //Affichage du temps, rafraichi toutes les 200ms
  Serial.println(dureeEcho);
  delay (1000);
}
```

FIGURE 7 – Code capteur, source : <http://edu.mrpigg.ca/son.html>

- La partie Void loop correspond à la réalisation des mesures. C'est une boucle qui va se réaliser jusqu'à ce qu'on stoppe. Cette partie est déjà bien commentée.

4.3 Mesure et résultats

Le montage de mesure est celui présenté dans la Fig. 8

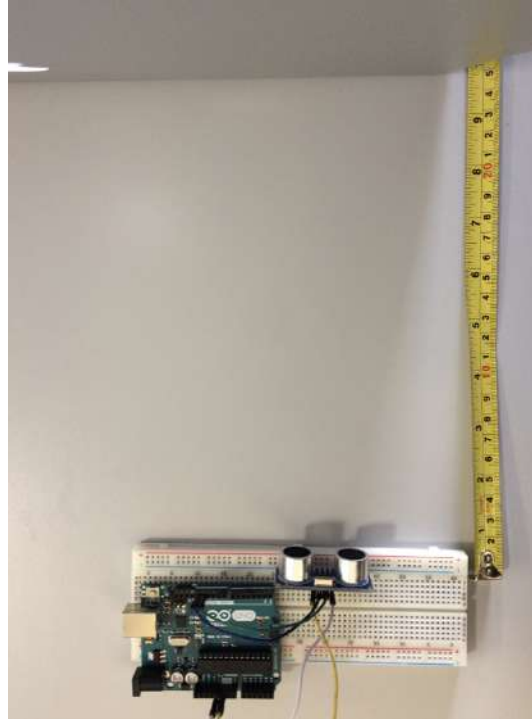


FIGURE 8 – Montage de mesure, source : <http://edu.mrpigg.ca/son.html>

Le protocole est le suivant :

- A une distance donnée on prend plusieurs mesures. *Plus on va s'éloigner moins ce sera stable à cause de la dispersion.* On fait la moyenne sur le temps.
- On déplace le mobile et on refait la mesure.
- On trace $t = f(d)$ et on retrouve la vitesse (on peut tracer l'inverse aussi).