

Thème Numérique

Faire de la physique avec ARDUINO - 2

Nicolas HERVE & Gilles ESPINASSE (ENSFEA)



Nous allons faire une série d'articles sur l'utilisation de la carte Arduino.

Ces articles seront progressifs dans la maîtrise technique de la carte et des utilisations possibles.

Le premier article a permis de présenter la carte, les fonctions de base de la programmation et quelques manipulations pour se lancer.

Ce deuxième article va introduire de nouveaux éléments de programmation, toujours à propos des diodes.

Quelques nouveaux éléments de syntaxe pour approfondir

Paramétrer les entrées/sorties

Nous avons vu lors de « faire de la physique avec arduino-1 » la fonction :

digitalWrite(broche, valeur);

qui permet de mettre une broche numérique au niveau logique « bas » ou « haut ».

Cette fonction possède une fonction analogue pour les broches analogiques :

analogWrite(broche, valeur);

Elle permet de mettre une broche analogique. Les broches analogiques sont celles avec le symbole ~ : les sorties 3,5,6,9,10 et 11), qui peuvent prendre des niveaux compris entre 0 et 255 (ces broches codent en 8bits, c'est-à-dire en $2^8 = 256$ niveaux). Les broches A0 jusqu'à A5 sont également des broches analogiques qui peuvent prendre $2^{10} = 1024$ niveaux.

Les réciproques de ces fonctions sont :

digitalRead(broche); qui permet de lire l'état logique d'une broche.

analogRead(broche); qui permet de lire le niveau d'une broche analogique.

Les structures de contrôle

Les structures de contrôle sont des blocs d'instructions qui s'exécutent en fonction du respect d'un certain nombre de conditions.

Nous allons voir dans les exercices qui suivent 2 types de structure de contrôle :

- *for* :

Cette structure permet d'exécuter un code **pour** un certain nombre de fois.

La syntaxe est donnée dans l'exemple suivant.

Exemple : on cherche à faire varier continuellement la tension sur la sortie 6. Cette sortie admet 256 niveaux de valeur (de 0 à 255). On souhaite que la carte fournisse tout d'abord 0V (niveau 0) puis atteigne progressivement (en passant par tous les niveaux) 5V (niveau 255).

Le programme s'écrit ainsi :

//pour i de 0 à 255, augmente le niveau de la sortie 6 d'un pas de 1 toutes les 10 ms.

```
for (int i=0; i <= 255; i=i+1){  
    analogWrite(6, i);  
    delay(10);  
}
```

int i =0 signifie que l'on définit un entier i à l'état initial 0.

i <= 255 signifie qu'il va atteindre une valeur maximale de 255 (255 compris).

i=i+1 signifie que l'on fait augmenter i d'une unité à chaque incrémentation. (on peut aussi remplacer toute cette expression par : i++)

Pour diminuer d'une unité l'incrément, on utiliserait la commande i=i-1

analogWrite(6, i) signifie que l'on met la broche 6 au niveau i.

La boucle for permet d'exécuter le programme 256 fois, avec une seule commande !

- *if...else* :

exécute un code **si** certaines conditions sont remplies et éventuellement **sinon** exécutera un autre code.

exemple :

//si la valeur du capteur dépasse le seuil

```
if(valeurCapteur>seuil){  
    //allume une diode branchée sur la broche 13  
    digitalWrite(13, HIGH);  
}
```

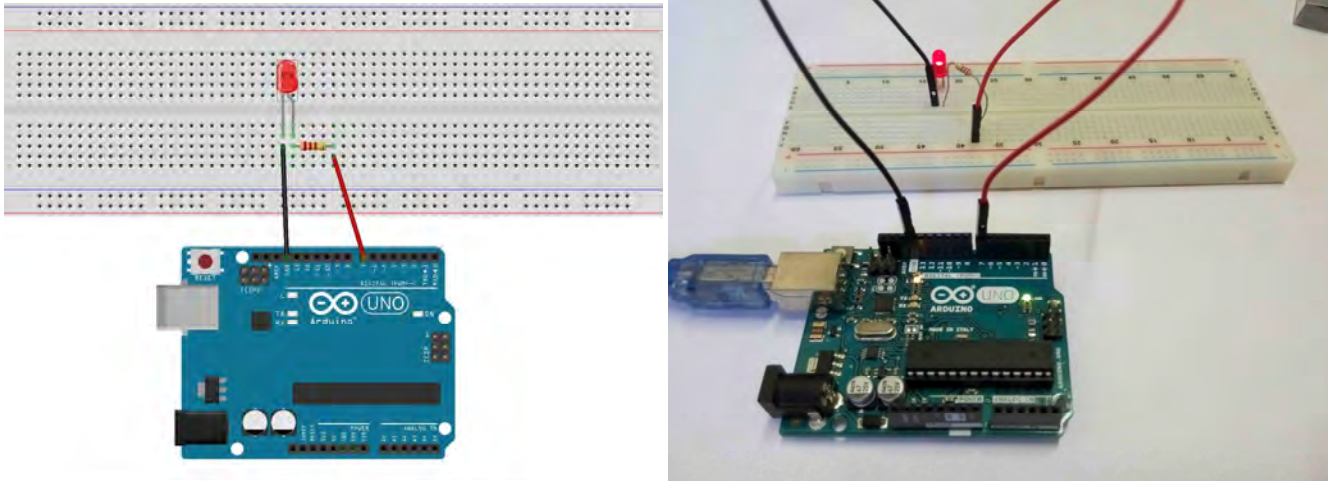
Montages associés à ces nouveaux éléments de syntaxe

Montage 1 :

Il s'agit d'allumer une diode de façon à ce qu'elle code SOS en Morse.

SOS se code en langage Morse par trois signaux courts suivis de trois signaux longs puis de trois signaux courts.

Le montage est le suivant :



Il y a plusieurs programmes possibles sur le logiciel ARDUINO :

- On peut soit programmer les 9 signaux successifs, comme dans l'exemple 3 de « faire de la physique avec arduino-1 ».
- On peut aussi optimiser le programme en utilisant une boucle for pour éviter les répétitions.

```
sketch_oct13a | Arduino 1.6.12
Fichier Édition Croquis Outils Aide

sketch_oct13a$

void setup() {
  pinMode(7, OUTPUT);
}
void loop() {
  for (int i =1;i<4;i++){
    digitalWrite(7,HIGH);
    delay(300);
    digitalWrite(7,LOW);
    delay(300);
  }
  for (int i =1;i<4;i++){
    digitalWrite(7,HIGH);
    delay(900);
    digitalWrite(7,LOW);
    delay(300);
  }
  for (int i =1;i<4;i++){
    digitalWrite(7,HIGH);
    delay(300);
    digitalWrite(7,LOW);
    delay(300);
  }
}
```

On peut encore optimiser ce programme en imbriquant 2 boucles for et en mettant une condition pour changer le temps d'allumage de la diode lors du « O » de SOS :



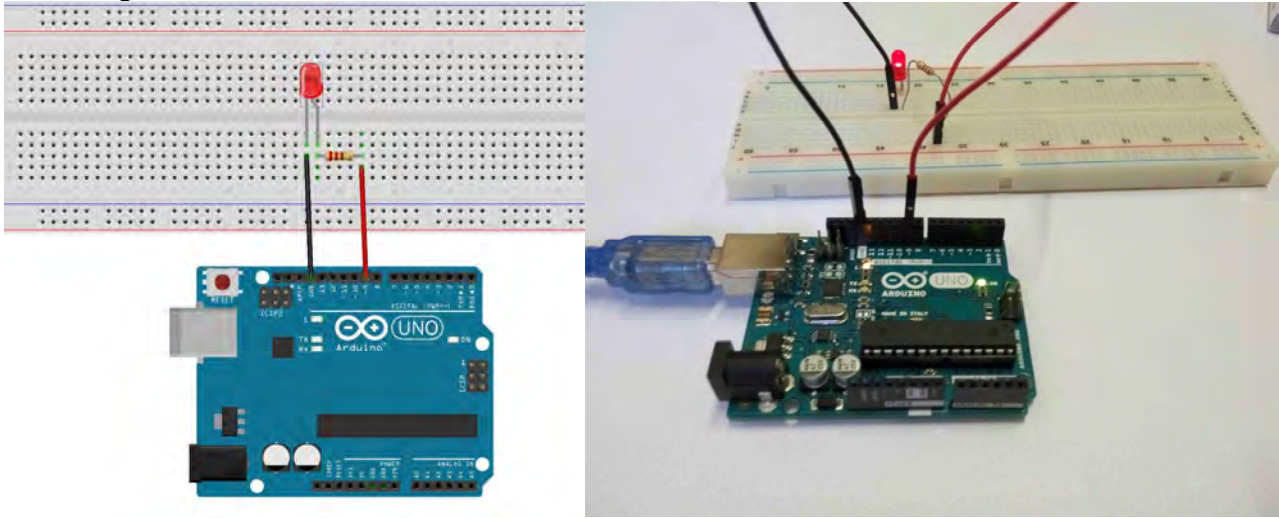
```
sketch_oct13a | Arduino 1.6.12
Fichier Édition Croquis Outils Aide

sketch_oct13a $
void setup() {
  pinMode(7, OUTPUT);
}
void loop() {
  for (int k =1;k<4;k++){
    for (int i =1;i<4;i++){
      digitalWrite(7,HIGH);
      if (k==2){
        delay(900);
      }
      else {
        delay(300);
      }
      digitalWrite(7,LOW);
      delay(300);
    }
  }
}
```

Montage 2 :

Il s'agit d'augmenter graduellement l'intensité lumineuse d'une diode, et de la faire diminuer.

Le montage est le suivant :



On choisit de faire varier l'intensité lumineuse toutes les 50 ms, avec un pas de 5. Il faut donc choisir une sortie analogique avec le symbole ~

Le programme ARDUINO est le suivant :

```
graduation | Arduino 1.6.9
Fichier Édition Croquis Outils Aide

graduation

void setup() {
  //on définit la broche analogique 9 comme étant une sortie
  pinMode(9,OUTPUT);
}

void loop() {
  // pour i de 0 à 255, on augmente le niveau de la sortie 9 d'un pas toute les 50ms
  for (int i=0;i<=255;i=i+5){
    analogWrite(9,i);
    delay(50);
  }

  // pour i de 255 à 0, on diminue le niveau de la sortie 9 d'un pas toute les 50ms
  for (int i=255;i>=0;i=i-5){
    analogWrite(9,i);
    delay(50);
  }
}

Enregistrement terminé.

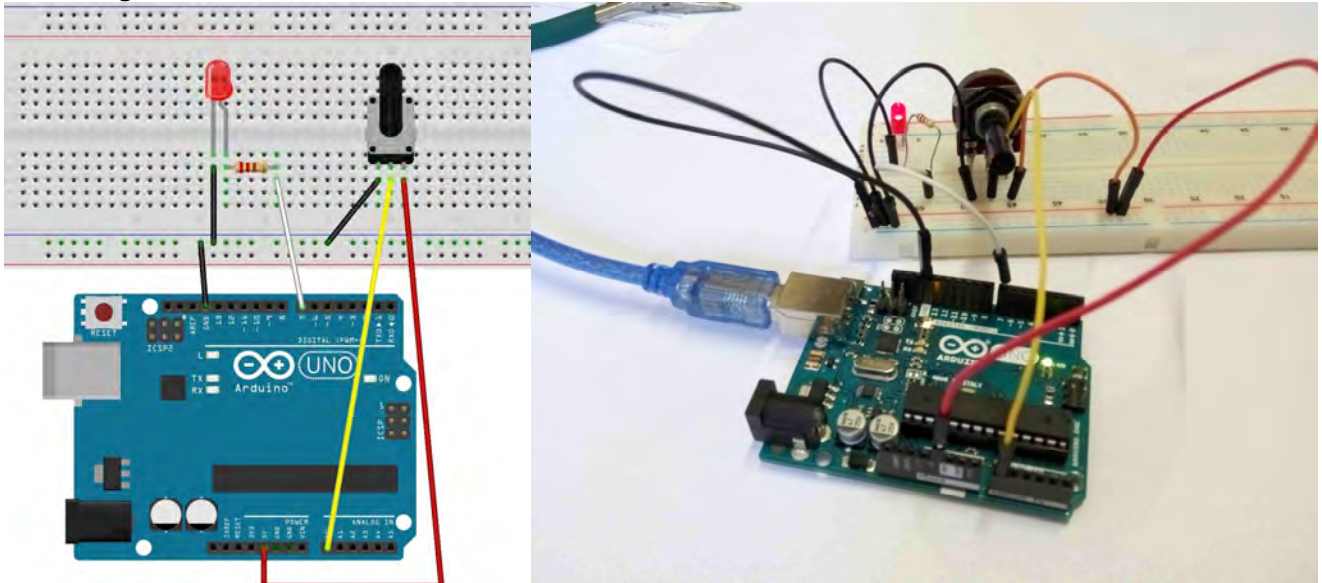
Le croquis utilise 1 314 octets (4%) de l'espace de stockage de programmes. Le maximum est de 32
```


Montage 3 :

Il s'agit d'allumer une diode en tournant un potentiomètre.

Un potentiomètre est une résistance variable. Il a trois pattes : une est à relier à la masse GND, une est à relier à 5 V et la dernière est à relier à une entrée analogique (exemple l'entrée A0).

Le montage est le suivant :



On choisit de faire allumer la diode si le potentiomètre est à mi-parcours (donc quand on dépasse le niveau $256/2 = 128$)

Le programme ARDUINO est le suivant :

```
potentiometre | Arduino 1.6.9
Fichier Édition Croquis Outils Aide

potentiometre

void setup() {
  //on définit la broche A0 comme étant une entrée (potentiomètre) et la broche 7 est une sortie (diode)
  pinMode(0,INPUT);
  pinMode(7,OUTPUT);
}

void loop() {
  // on définit la variable potar comme le niveau mesuré de la broche A0
  int potar = analogRead(0);

  // si la valeur sur la broche A0 (potentiomètre) dépasse le niveau 128, alors la diode s'allume
  if (potar>128){
    digitalWrite(7,HIGH);
  }
  else{
    digitalWrite(7,LOW);
  }
}
```

Bibliographie

DUMONT, A. & YERNAUX, B. (2017). *50 montages pédagogiques avec Arduino*. Dijon : Educagri édition.

VAN DREUMEL, W. (2016). *36 expériences de physique avec Arduino pour la maison et l'école: Newton a rendez-vous avec l'électronique*. Publitronic – Elektor.

