

# Thème numérique

## Faire de la physique avec ARDUINO - 4



Ce quatrième chapitre va introduire de nouveaux éléments de programmation, notamment la notion de bibliothèques, très utiles quand on veut utiliser des composants plus complexes.

Le projet :

- Actionner un servo-moteur
- Afficher la température sur un écran LCD

### Quelques nouveaux éléments de syntaxe

#### Appeler une bibliothèque externe :

Une bibliothèque est un ensemble de fonctions utilitaires, regroupées et mises à disposition des utilisateurs de l'environnement Arduino afin de ne pas avoir à réécrire des programmes parfois complexes.

Une bibliothèque est souvent nécessaire pour faire fonctionner des capteurs ou des actionneurs évolués.

Pour les importer dans votre programme, vous devez les sélectionner dans Sketch > Import Library.

L'instruction suivante devra donc être ajoutée au début de votre programme.

```
#include <la_bibliothèque.h>
```

Exemple 1 : on souhaite se servir d'un servo-moteur, c'est-à-dire d'un moteur dont on peut contrôler la position de l'arbre.

Pour cela, on fait appel à une bibliothèque liée à ce composant, qui est appelée servo

La ligne de code est alors :

```
#include <servo.h>
```

On a alors accès à de nouvelles fonctions qui sont spécifiques du composant :

- Servo NOM ; permet de nommer ou d'identifier le servomoteur en lui donnant un NOM
- NOM.attach (broche) ; permet de déclarer le servomoteur nommé NOM comme étant sur la broche (généralement la broche 9)
- NOM.write (angle) ; permet d'actionner le servomoteur nommé NOM en lui donnant un angle exprimé en degré

Exemple 2 : on souhaite afficher un texte sur un écran LCD

Pour cela, on fait appel à la bibliothèque liée à ce composant, qui est appelée liquidCrystal

La ligne de code est alors :

```
#include <liquidCrystal.h>
```

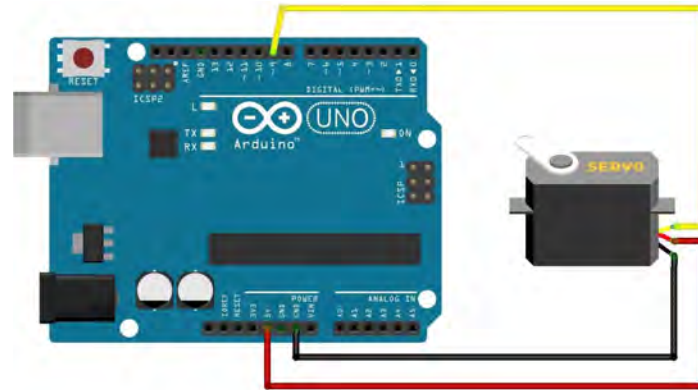
On a alors accès à de nouvelles fonctions :

- LiquidCrystal NOM(broche, broche, broche, broche, broche, broche); permet de nommer l'écran LCD et de paramétrer son branchement (voir montage associé au chapitre)
- NOM.begin(nombre de caractères que l'on peut afficher sur une ligne, nombre de lignes) ; permet d'initialiser les dimensions de l'affichage. Pour un afficheur 16x2, on écrit donc NOM.begin(16,2) ;
- NOM.clear() ; permet d'effacer l'écran
- NOM.print("texte") ; permet d'afficher un texte
- NOM.setCursor(colonne, ligne) ; permet d'initialiser la position où va se faire l'affichage. Par exemple, le saut de ligne se code NOM.setCursor(0,1) ; car la ligne 1 est la deuxième ligne de l'écran

## Montages associés aux éléments de syntaxe du chapitre 4

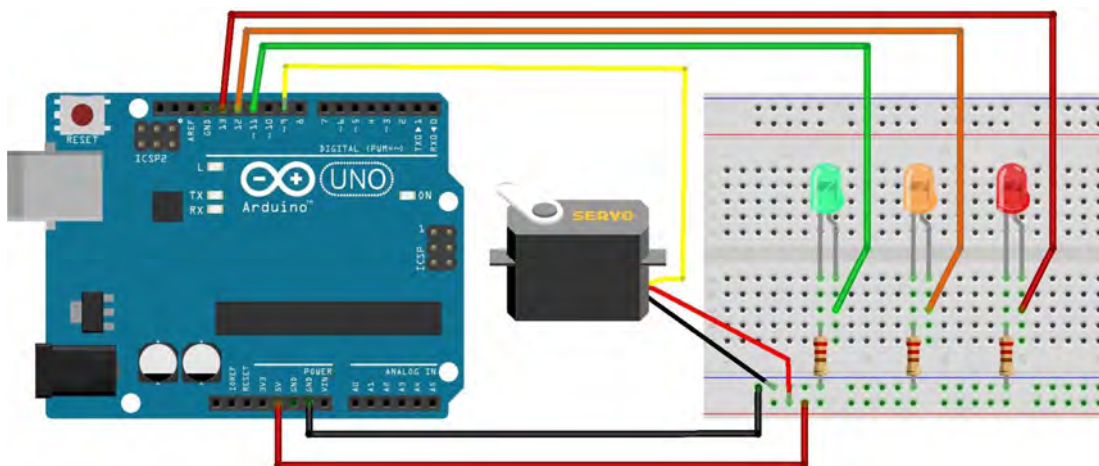
### Câblage d'un servomoteur

Un servomoteur possède 3 fils de connections : le rouge pour l'alimentation (3,3V ou 5V), le noir pour la masse, et le jaune pour recevoir les instructions (généralement branché sur la broche n°9).



### Montage 1 :

Faire lever une barrière quand le feu tricolore passe au vert.



### Programme :

```
#include <Servo.h>
```

```
Servo servomoteur; // crée l'objet pour contrôler le servomoteur
```

```
int Rouge = 13; // Numéro de la broche où est connectée la LED rouge du feu
```

```
int Orange = 12; // Numéro de la broche où est connectée la LED orange du feu
```

```
int Vert = 11; // Numéro de la broche où est connectée la LED verte du feu
```

```
    // le code dans cette fonction est exécuté une fois au début  
void setup() {
```

```
    // on accroche notre servomoteur branché sur la broche 9
```

```
servomoteur.attach(9); // utilise la broche 9 pour le contrôle du servomoteur
servomoteur.write(0); // positionne le servomoteur à 0°

// indique que les broches des LED sont des sorties :
pinMode(Rouge, OUTPUT);
pinMode(Orange, OUTPUT);
pinMode(Vert, OUTPUT);

// le feu est au rouge
digitalWrite(Vert, LOW);
digitalWrite(Orange, LOW);
digitalWrite(Rouge, HIGH);
}

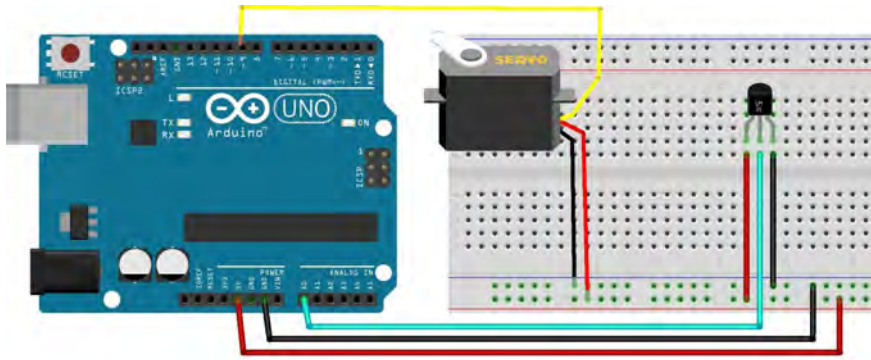
// le code dans cette fonction est exécuté en boucle
void loop() {

delay(4000);
digitalWrite(Rouge, LOW); // on éteint le feu rouge
digitalWrite(Vert, HIGH); // on fait passer le feu tricolore au vert
servomoteur.write(90); //on lève la barriere
delay(4000);
digitalWrite(Orange, HIGH); // on allume le feu orange
digitalWrite(Vert, LOW); // on éteint le feu vert
delay(1000);
digitalWrite(Rouge, HIGH); // on allume le feu rouge
digitalWrite(Orange, LOW); // on éteint le feu orange
servomoteur.write(0); //on baisse la barriere

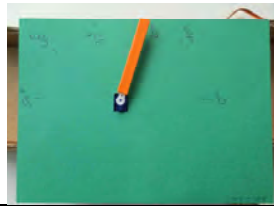
}
```

## Montage 2 :

Mesurer la température et la visualiser par l'évolution d'un servomoteur (en vérifiant sur le port série).



Pour donner quelque chose comme :



## Programme :

```
// Couplage Thermomètre LM35 avec un servomoteur pour l'indication de la température
```

```
#include <Servo.h>
```

```
Servo servomoteur; // crée l'objet pour contrôler le servomoteur
```

```
int valeurbrute, angle;
```

```
float temperature;
```

```
void setup() {
```

```
  // on accroche notre servomoteur branché sur la broche 9
```

```
  servomoteur.attach(9); // utilise la broche 9 pour le contrôle du servomoteur
```

```
  servomoteur.write(0); // positionne le servomoteur à 0°
```

```
  //on initialise le port série
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
  // Lecture temperature et affichage
```

```
  valeurbrute = analogRead(A0);
```

```
  temperature = valeurbrute * 5.0 / 1023.0 * 100.0;
```

```
  Serial.print("Temperature : ");
```

```
  Serial.print(temperature);
```

```
  Serial.println(" C");
```

```
  angle = map(temperature, 0, 50, 0, 180); // mise à l'échelle : 0 <= température <= 50
```

```
  servomoteur.write(angle);
```

```
  Serial.print("Angle : ----- ");
```

```
  Serial.print(angle);
```

```
  Serial.println(" degres");
```

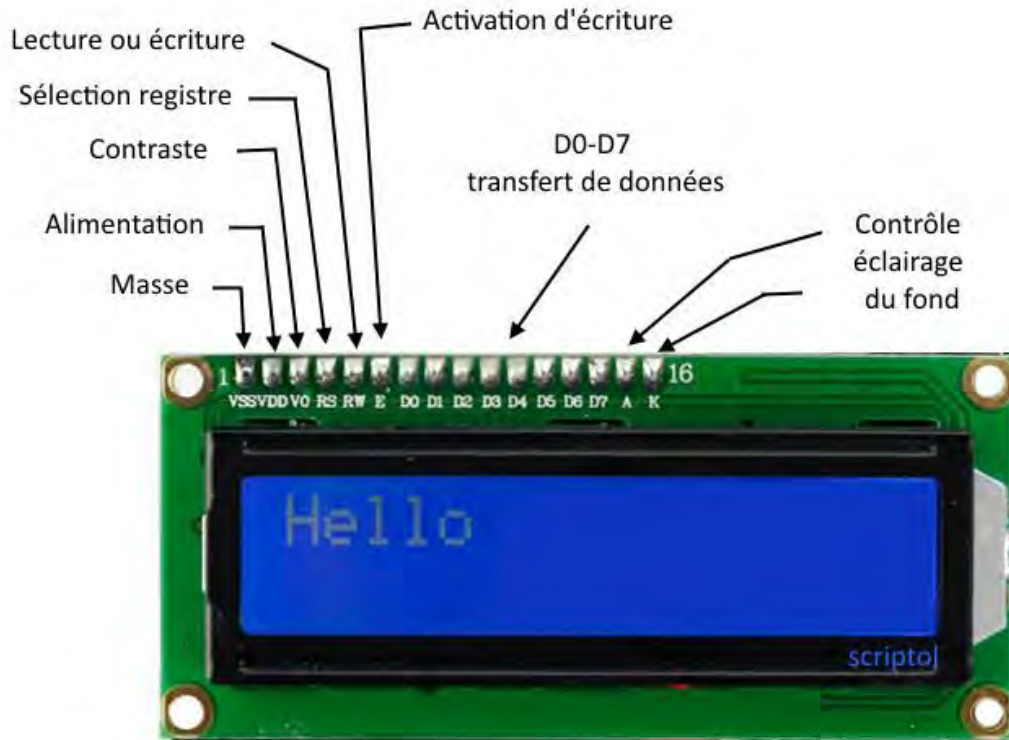
```
  // Délai entre mesures
```

```
  delay(2000);
```

```
}
```

## Câblage d'un écran LCD XXX

LCD est l'abréviation anglaise de "liquid crystal display" qui veut dire : afficheur à cristaux liquides.

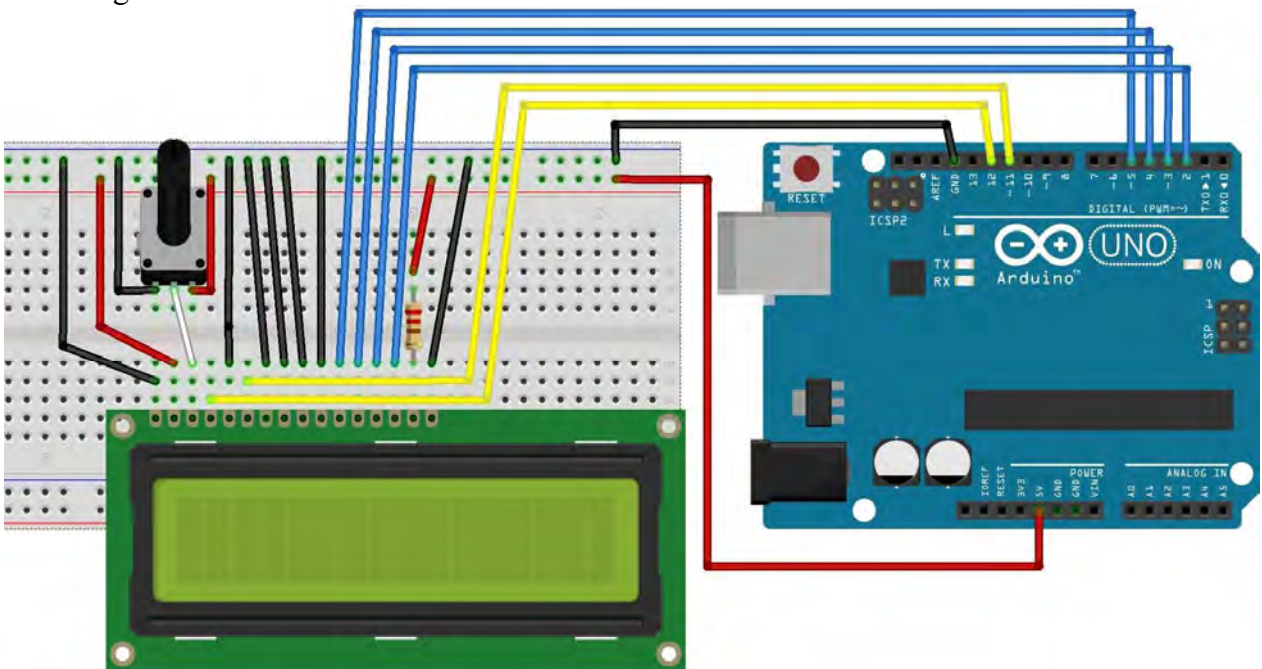


<https://www.scriptol.fr/montages/salut-le-monde.php>

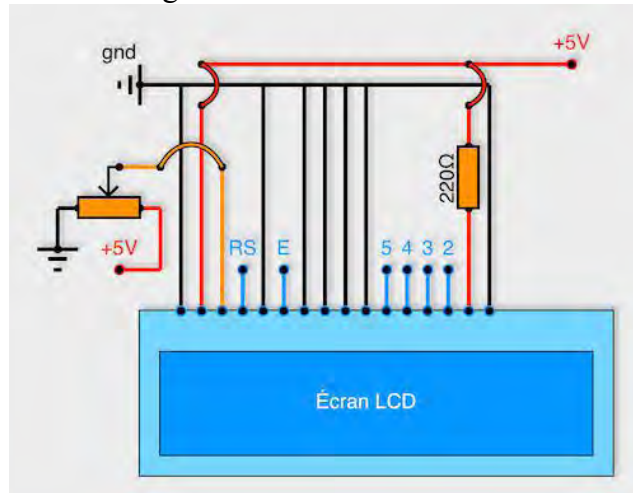
Il y a 2 façons de câbler un afficheur LCD : en utilisant 4 broches (mode 4 bits) ou 8 broches (mode 8 bits), en plus de 2 broches pour l'alimentation. L'avantage du mode 8 bits est une plus grande vitesse de communication, mais l'inconvénient est d'utiliser 8 broches.

Dans la suite, nous câblons l'écran LCD en mode 4 bits.

Le câblage est le suivant :



Une autre façon de schématiser le câblage est le suivant :



<https://openclassrooms.com/fr/courses/3290206-perfectionnez-vous-dans-la-programmation-arduino/3342221-programmez-un-ecran-lcd>

Le paramétrage de l'écran se fait avec la fonction LiquidCrystal, de la façon suivante :

LiquidCrystal NOM(RS, E, D4, D5, D6, D7);

Dans le câblage du schéma, cela donne LiquidCrystal NOM(12, 11, 5, 4, 3, 2);

### **Montage 3 :**

Il s'agit de faire un montage qui permet d'afficher « FAB LAB » sur la première ligne et « ENSFEA » sur la deuxième ligne d'un écran LCD

Voir le montage de l'écran LCD

### **Programme :**

```
// inclure la librairie :
#include <LiquidCrystal.h>

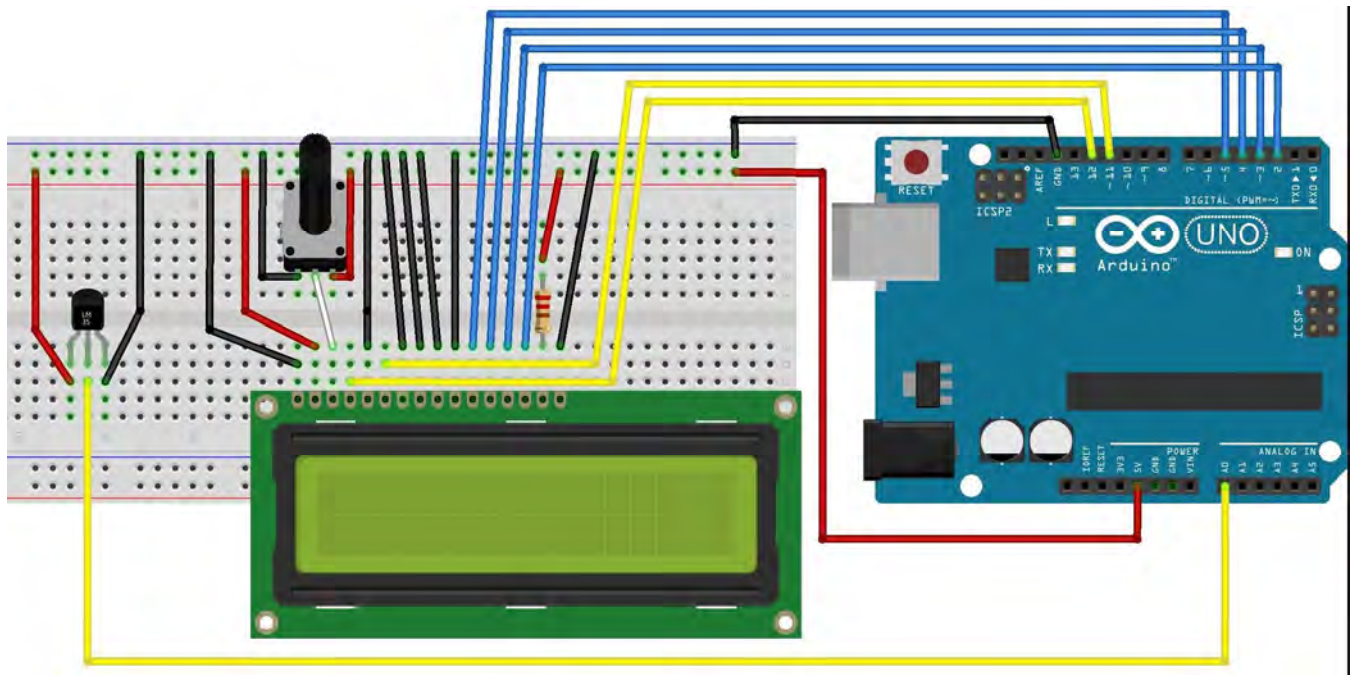
// initialise la librairie avec les numéros de pins de l'arduino, en fonction des branchements
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // initialise l'écran LCD, nombre de colonne et de ligne:
  lcd.begin(16, 2);
  // Ecrit un message sur le LCD (facultatif)
  lcd.print("bonjour");
}

void loop() {
  // Met le curseur sur la colonne 0, ligne 0
  lcd.setCursor(0, 0);
  lcd.print("FAB LAB");// Ecrit le mot FAB LAB sur la première ligne
  lcd.setCursor(0, 1);
  lcd.print("ENSFEA");// Ecrit le mot ENSFEA sur la deuxième ligne
}
```

### Montage 4 :

Il s'agit de faire un montage qui permet d'afficher la température, prise toutes les secondes, sur l'écran LCD.



### Programme :

```
// inclure la librairie :
#include <LiquidCrystal.h>

// initialise la librairie avec les numéros de pins de l'arduino, en fonction des branchements
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
// initialise l'écran LCD, nombre de colonne et de ligne:
lcd.begin(16, 2);
// Ecrit un message sur le LCD (facultatif)
lcd.print("bonjour");
delay(1000);
//efface les caractères de l'écran
lcd.clear();
}

void loop() {

int valeur_brute = analogRead(A0);
float temperature_celcius = valeur_brute * 5.0 / 1023.0 * 100.0;
// Met le curseur sur la colonne 0, ligne 0
lcd.setCursor(0, 0);
lcd.print(temperature_celcius);// Ecrit le mot FAB LAB sur la première ligne
delay(1000);
}
```



# 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.

