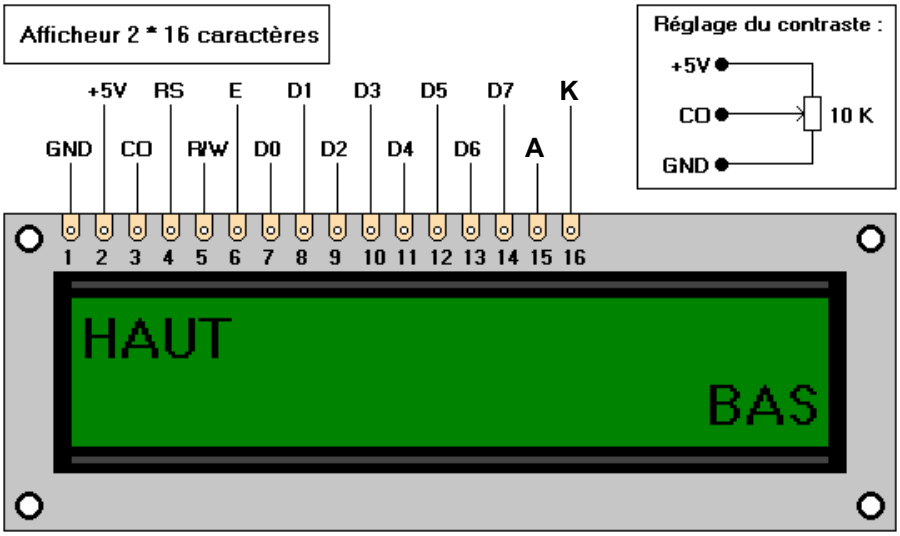
LES AFFICHEURS LCD

Les afficheurs LCD sont devenus incontournables dans toutes applications qui demandent la visualisation de paramètres. Auparavant onéreux et difficiles à mettre en œuvre, ils sont maintenant bon marchés et l'interface parallèle, au standard Hitachi, permet un pilotage facile.

On rencontre aussi de plus en plus d'afficheur pilotable avec un port série ou I2C.

Les afficheurs LCD se ressemblent tous, à part le nombre de lignes et le nombre de colonnes, le fonctionnement et le brochage est standard et identique. Un des points intéressant est de pouvoir contrôler l'afficheur en mode 8bits ou en mode 4bits.

Le brochage



L'afficheur LCD a 14 broches en standard et souvent 16, les broches 15 et 16 servent au rétro-éclairage (une option).

Broche	Nom	Description
1	Vss	Masse
2	Vdd	Alimentation 5v
3	CO	Variables de 0 à 5v permet de modifier le contraste de l'afficheur
4	RS	Indique une commande ou une donnée à afficher (0: Commande / 1: Donnée)
5	R/W	Indique une écriture ou une lecture (0: Écriture / 1: Lecture)
6	E	Indique une validation (Le niveau Haut doit être maintenu 500µs)
7	D0	Bus de données bidirectionnel.
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	A	Anode rétroéclairage (+5V)
16	K	Cathode rétroéclairage (masse)

Le fonctionnement

On envoie deux types d'information à l'afficheur :
- les **commandes** qui permettent de l'initialiser : positionnement du curseur, effacement écran, etc. ;
- les **données** à afficher.

L'entrée **RS** permet de spécifier si on envoie une commande ou une donnée :
- RS=0 : instruction (commande) ;
- RS=1 : caractère (donnée).

L'afficheur dispose d'une entrée **R/W** pour spécifier une lecture ou une écriture :
- R/W=0 : écriture vers l'afficheur ;
- R/W=1 : lecture de l'afficheur.

Pour valider tous les échanges sur le bus de données (D7-D0) on utilise l'entrée **E** de l'afficheur. Un **front descendant** sur cette entrée valide la donnée. En programmation, il faudra placer un court instant **E** à l'état haut puis à l'état bas.

Il est possible d'utiliser l'afficheur LCD en mode 8 bits normal ou en mode 4 bits pour économiser les broches de son µContrôleur par exemple, c'est assez pratique :

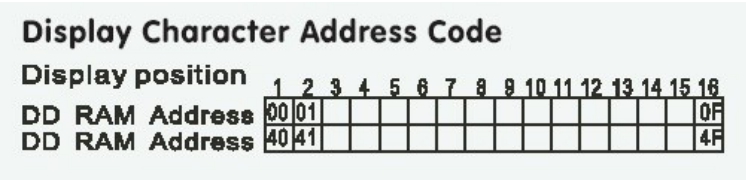
Mode 8 bits:
En mode 8 bits on place la donnée ou la commande sur le bus **D7** à **D0** et on valide avec **E**

Mode 4 bits:
En mode 4 bits on place déjà les poids forts de la donnée ou la commande sur les bits de **D7** à **D4** et on valide une première fois avec **E**. Puis on va mettre le poids faibles sur les bits de **D7** à **D4** et on valide une seconde fois avec **E**.
L'envoie ou la lecture d'un octet s'effectue donc en 2 temps dans ce mode

Les mémoires

Les afficheur LCD possède 2 types de mémoires :
- la **DD RAM** qui mémorise les caractères affichés à l'écran ;
- la **CG RAM** qui contient la table des caractères affichables.

DD RAM (Display Data RAM)
La DD RAM commence à l'adresse 0x00 et dans le cas d'un afficheur 16x2 lignes, elle termine à 0x4F. C'est une mémoire d'affichage dont l'adresse contient le caractère affiché à l'écran à une certaine position.
La première ligne commence en 0x00 jusqu'à 0x0F inclue.
La seconde ligne commence en 0x40 jusqu'en 0x4F inclue.



Exemple : on utilise un afficheur LCD pour visualiser la température d'une pièce (voir ci-dessous). Donner les adresses des 2 cases où devront être envoyés les chiffres de la température.

CG RAM (Character generator RAM)

Cette mémoire stocke le code des caractères affichables (voir la table ci-dessous). Une partie de la CG RAM est modifiable à volonté par le développeur, les 8 premiers caractères. Si un caractère de la CG RAM qui est actuellement sur l'afficheur est changé, alors le changement est immédiatement apparent sur l'afficheur.

La table des caractères

Utilisation avec une librairie

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
    lcd.begin(16, 2);
}

void loop() {
  lcd.setCursor(3, 0);
  lcd.print("Coucou");
  }
```

Pour utiliser une librairie, il faut la déclarer en début de programme.
La ligne LiquidCrystal lcd(rs, en, d4, d5, d6, d7) ; permet de tenir compte du câblage de l'afficheur. Donner les valeurs correctes à rs , en, d4, d5, d6, d7. On peur avaoir :

```
LiquidCrystal lcd(rs, enable, d4, d5, d6, d7); // mode 4 bits - RW non connectée (le plus utilisé)
LiquidCrystal lcd(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7); //mode 8 bits - RW utilisée
```

lcd.setCursor(colonne, ligne) permet de commencer un texte à l'endroit voulu.

lcd.print("texte")	affiche un texte
lcd.print(data)	affiche une donnée (nombre ou texte dans une variable)

Il existe d'autre fonctions, pour les connaître, voir la référence : https://www.arduino.cc/en/Reference/LiquidCrystal .