Ecran LCD texte

D.Moreaux

2 novembre 2018

1 L'écran

Les écran LCD texte permettent de 1, 2 ou 4 lignes de texte allant de 10 à 20 caractères par ligne. A peu de choses près, les commandes sont les mêmes quel que soit le nombre de lignes ou de colonnes.

La commande se fait en écrivant dans des registres de contrôle ou dans la mémoire. La sélection entre les deux se fait à l'aide de la broche `RS` de l'écran.

La lecture en mode registre de contrôle permet de connaître le status de l'écran (busy et l'adresse mémoire pour la prochaine écriture).

La mémoire est décomposée en deux zones : CGRAM qui contient les 8 caractères définis par l'utilisateur et DDRAM qui contient l'écran.

Le choix entre lecture et écriture se fait à l'aide de la broche `RW`. Un niveau bas indique une écriture. Une impulsion sur la broche `E` validera la lecture ou l'écriture.

Il est possible de se passer de la lecture. En effet, les datasheet indiquent le temps maximum pris par chaque opération. Il suffit d'attendre suffisemment longtemps que pour que l'opération se termine. Cela permet d'économiser un signal au détriment d'un peu de vitesse [1]

L'écran permet également un mode 4-bits. Dans ce mode, seuls D4 à D7 sont connectés, D3 à D0 sont inutilisés et n'ont pas besoin d'être connectés au microcontroleur, ce qui permet d'économiser 4 broches.

Lorsque l'on écrit un message sur l'écran, les caractères de 0 à 0 et de 08 à 0F correspondent aux 8 caractères redéfinissables (définis dans CGRAM). Les caractères de 10 à FF contiennent les caractères ASCII (à quelques exceptions près), les katakana (alphabet Japonais) et quelques autres caractères.

1. A noter que la librairie Arduino n'utilise pas le status et donc, le signal RW est inutilisé et peut être relié à la masse

L'écran intègre un rétro-éclairage à LED. La résistance de limitation est incluse dans l'écran, il suffit donc de fournir l'alimentation de 5V directement à l'écran.

2 Brochage

1	VSS	Alimentation négative (0V)
2	VDD	Alimentation positive (5V)
3	V0	Réglage contraste, potentiel entre 0 et 5V à régler
4	RS	Sélection de registre (mémoire/contrôle)
5	RW	Lecture/écriture (à relier à la masse)
6	E	Enable : une impulsion pour valider les données
7	D0	Data bit 0
8	D1	Data bit 1
...	...	
14	D7	Data bit 7
15	A	Anode : alimentation positive rétroéclairage (5V)
16	K	Kathod : alimentation négative rétroéclairage (0V)

Le plus souvent, V0 est le point milieu d'une résistance ajustable de minimum 10k dont les bornes extérieurs sont reliées au 0V et au 5V.

On utilisera généralement RS, E et D4 à D7 (mode 4 bits) ce qui ne demandera que 6 fils sur l'arduino.

3 Librairie Arduino

L'utilisation de l'écran se fait à l'aide de la librairie LiquidCrystal.

On déclare l'écran à l'aide des lignes suivantes à mettre avant le `setup`

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(RS, E, D4, D5, D6, D7);
```

Les mentions `RS`, `E` et `D4` à `D7` seront remplacées par les numéros des broches où les dits signaux sont connectés. A noter que l'on peut utiliser le mode 8 bits en précisant les valeurs de `D0` à `D7`

Dans le setup, on utilisera la commande

```
lcd.begin(col, lig);
```

où `col` sera le nombre de colonnes (caractères par ligne) et `lig` le nombre de lignes. Ces nombres sont présents au niveau de l'écran, par exemple, un 1602 est un écran de 2 lignes de 16 caractères.

Une fois l'écran initialisé, on pourra utiliser les commandes suivantes

lcd.clear()	Efface l'écran et repositionne en (0,0)
lcd.home()	Positionne en (0,0)
lcd.setCursor(c,l)	Positionne à la ligne l et la colonne c
lcd.print(x)	Affiche la valeur x. x peut être une variable numérique, une chaîne, ...
lcd.write(x)	Affiche le caractère x

Des commandes permettent l'affichage du curseur (masqué ou visible, fixe ou clignotant), le choix du sens d'écriture, l'allumage ou l'extinction de l'écran ou de faire défiler l'écran vers la gauche ou la droite.

Pour redéfinir un caractère utilisateur (code de 00 à 07 ou 08 à 0F), on utilisera

```
byte symbole[8] = {
  B00110,
  B11001,
  B01010,
  ...
};

lcd.createChar(n,symbole);
```

Le tableau `symbole` contient 8 lignes avec des nombres sur 5 bits qui forme le tableau 8x5 du nouveau caractère (un bit à 1 pour une case noire). La valeur `n` correspond au numéro du caractère de 0 à 7.