

L'ECRAN LCD

Esteban Chevalier-Drevon
Matyas Ricci
Groupe F

Exercices : L'écran LCD

1. Exercice 1: Branchement du LCD:

Réaliser le branchement de votre afficheur LCD, faites-le vérifier par votre enseignant.
La procédure est la suivante : [Le branchement à été vérifié et validé par l'enseignant.](#)

Branchez le module LCD aux broches de l'Arduino comme représenté ci-dessous :

Vérifiez que le cavalier est en place

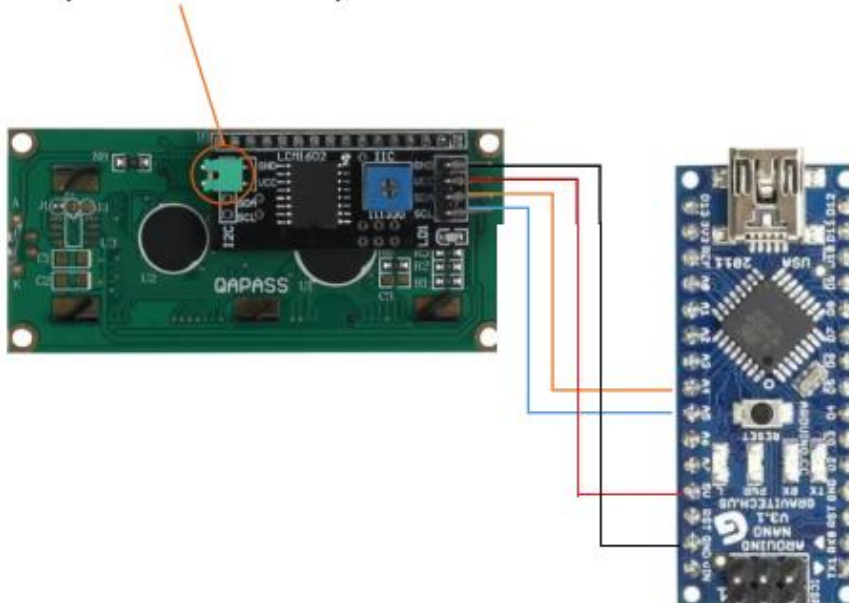


Table de correspondance :

Afficheur LCD I2C	Arduino
GND	GND
VCC	+5V
SDA	A4
SCL	A5

2. Recherche de l'adresse du LCD

Télécharger le programme scanI2C sur :

<https://moodle.univ-tln.fr/mod/folder/view.php?id=166398>

Exécuter le et déterminer quelle est l'adresse de votre afficheur.

Adresse de mon afficheur : [0x27](#)

3. Installation de la bibliothèque LiquidCrystal I2C

Vous devez d'abord vérifier si la bibliothèque n'est pas déjà installée en lançant Arduino et en regardant dans les exemples. Si ce n'est pas le cas, alors :

- Aller sur la page : <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library> puis télécharger le fichier « zip » et suivre les instructions de la rubrique installation.
- Toutes les fonctions que vous pouvez exécuter sont décrites dans le fichier d'en tête.

L'afficheur LCD utilisé est composé de deux lignes de 16 caractères rétro-éclairé se raccordant via le bus I²C sur un microcontrôleur (Arduino par exemple). [Bibliothèque installée.](#)

4. Programme n°1: Hello World

Modifier puis exécuter le programme HelloWorld dans le répertoire exemple de la bibliothèque LiquidCrystal_I2C. [Le programme d'exemple fonctionne l'écran LCD est opérationnel.](#)

5. Programme n°2: J'écris où je veux

Modifier ce programme de telle sorte à écrire sur la deuxième ligne à partir du 3ème caractère.

Programme :

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27,16,2); // Addressage de notre écran LCD à l'adresse 0x27
5  // en définissant les dimensions (16 colonnes, 2 lignes)
6
7  void setup()
8  {
9      lcd.init();           // Initialisation de l'écran LCD
10     lcd.backlight();       // Allumage de l'éclairage de l'écran.
11     lcd.setCursor(2,1);    // Positionnement du curseur (3ème caractère; 2ème ligne)
12     lcd.print("Hello World"); // Affichage de notre message à la position actuelle du curseur
13 }
14
15 void loop()
16 {
17 }
18
```

Réalisation d'un robot autonome détecteur d'obstacles

6. Programme n°3: Combien de temps

Afficher le nombre de secondes (entier) écoulées depuis le lancement du programme.

Recommandations :

- On affichera une seule fois les chaînes constantes TEMPS et secondes.
- Pas d'accès inutiles au bus I²C, c-à-d un seul accès donc une seule écriture par secondes.

					T	E	M	P	S	:					
S	e	c	o	n	d	e	s	:	1	9	2	4			

Programme :

```

1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27,16,2); // Addressage de notre écran LCD à l'adresse 0x27
5                                     // en définissant les dimensions (16 colonnes, 2 lignes)
6
7  void setup()
8  {
9      lcd.init(); // Initialisation de l'écran LCD
10     lcd.backlight(); // Allumage de l'éclairage de l'écran.
11     lcd.setCursor(5,0); // Positionnement du curseur à la 6ème colonne à la 1ère ligne
12     lcd.print("TEMPS:"); // Affichage du message "TEMPS" sur l'écran LCD
13     lcd.setCursor(0,1); // Positionnement du curseur à la 1ère colonne à la 2ème ligne
14     lcd.print("Secondes:"); // Affichage du message "Secondes:" sur l'écran LCD
15 }
16
17 void loop()
18 {
19     lcd.setCursor(9,1); // positionnement du curseur à la 10ème colonnes à la 2 ème ligne.
20     lcd.print(millis() / 1000); // Affichage du nombre de secondes écoulées
21     // depuis le démarrage du programme.
22     delay(1000); // Attente de 1 seconde entre chaque rafraichissement
23 }
24

```

Réalisation d'un robot autonome détecteur d'obstacles

7. Programme n°4: Le même avec la précision en plus

En respectant toujours les mêmes recommandations, modifier le programme précédent pour afficher les heures, min, secondes et dixièmes de seconde.

					T	E	M	P	S	:					
				1	2	:	3	2	:	2	4	:	8		

Programme :

```

1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27, 16, 2); // Addressage de notre écran LCD à l'adresse 0x27
5                                     // en définissant les dimensions (16 colonnes. 2 lignes)
6  void setup()
7  {
8      lcd.init(); // Initialisation de l'écran LCD
9      lcd.backlight(); // Allumage de l'éclairage de l'écran.
10     lcd.setCursor(5, 0); // Positionnement du curseur à la 6ème colonne à la 1ère ligne
11     lcd.print("TEMPS:"); // Affichage du message "TEMPS" sur l'écran LCD
12     lcd.setCursor(8, 1);
13     lcd.print("M:");
14     lcd.setCursor(4, 1);
15     lcd.print("S:");
16     lcd.setCursor(9, 1);
17     lcd.print("D:");
18 }
19
20 void loop()
21 {
22     unsigned long int t = millis();
23     static unsigned long int t0;
24     static unsigned int D, S, M;
25     if (t - t0 > 1000) {
26         t0 = t;
27         D++;
28         if (D == 10) {
29             D = 0;
30             S++;
31             if (S == 60) {
32                 S = 0;
33                 M++;
34                 lcd.setCursor(2, 1);
35                 lcd.print(M);
36             }
37             lcd.setCursor(6, 1);
38             if (S < 10) {
39                 lcd.print(" ");
40             }
41             lcd.print(S);
42         }
43         lcd.setCursor(11, 1);
44         lcd.print(D);
45     }
46 }
47

```