Écran LCD I2C- Arduino

1. Objectifs :

- Écrire sur un écran à cristaux liquides
- Utiliser une bibliothèque externe
- Utiliser les fonctions d'une librairie
- Afficher une valeur lue sur l'écran LCD

2. Matériel :

- Arduino UNO
- Écran LCD
- Ordinateur + fil de branchement pour la carte
- Quelques fils
- Plaque d'essai (« breadboard »)
- Potentiomètre

3. Ce que fait le programme

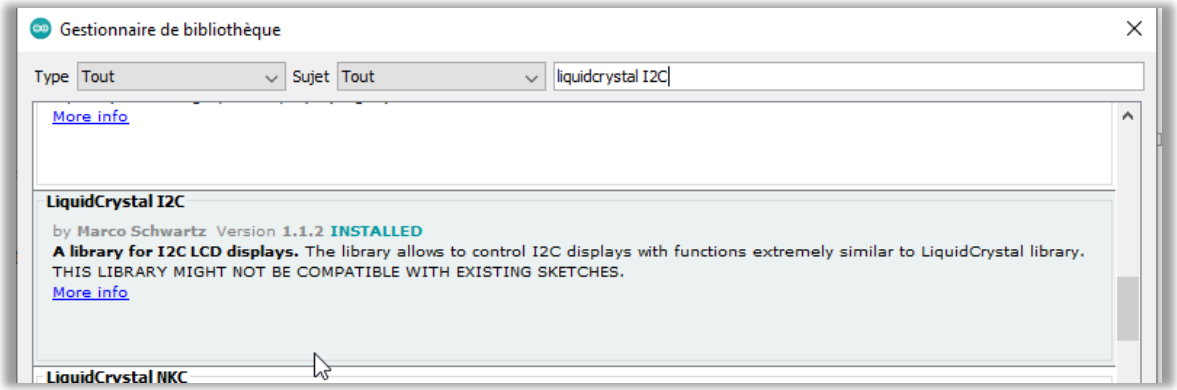
Ce programme permet d'écrire sur un écran à cristaux liquides. La première ligne affiche « Hello, world! » et la deuxième ligne affiche « Arduino ».

4. Étapes de réalisation

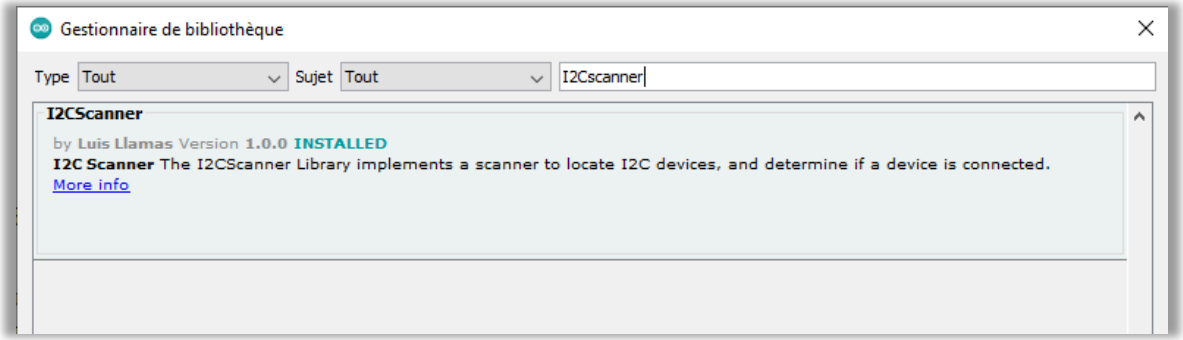
1. Procéder de la manière habituelle pour le circuit, l'application et le programme.

2. Il faut installer une bibliothèque externe pour utiliser ses fonctions. Pour ce faire, dans l'application Arduino, cliquer sur l'onglet « *Croquis* », choisir « *Inclure une bibliothèque* » et « *Gérer les bibliothèques* ». La fenêtre suivante s'ouvrira :

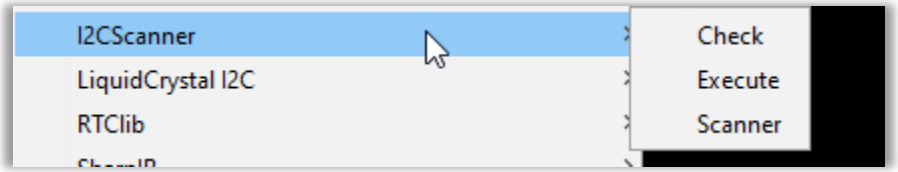
3. Rechercher « LiquidCrystal i2C ». Dérouler la liste des bibliothèques disponibles jusqu'à LiquidCrystal I2C. Si « *Installed* » est indiqué sur la première ligne de la description, la bibliothèque est installée. On peut fermer cette fenêtre. Sinon, cliquer sur « *More info* » et poursuivre l'installation.



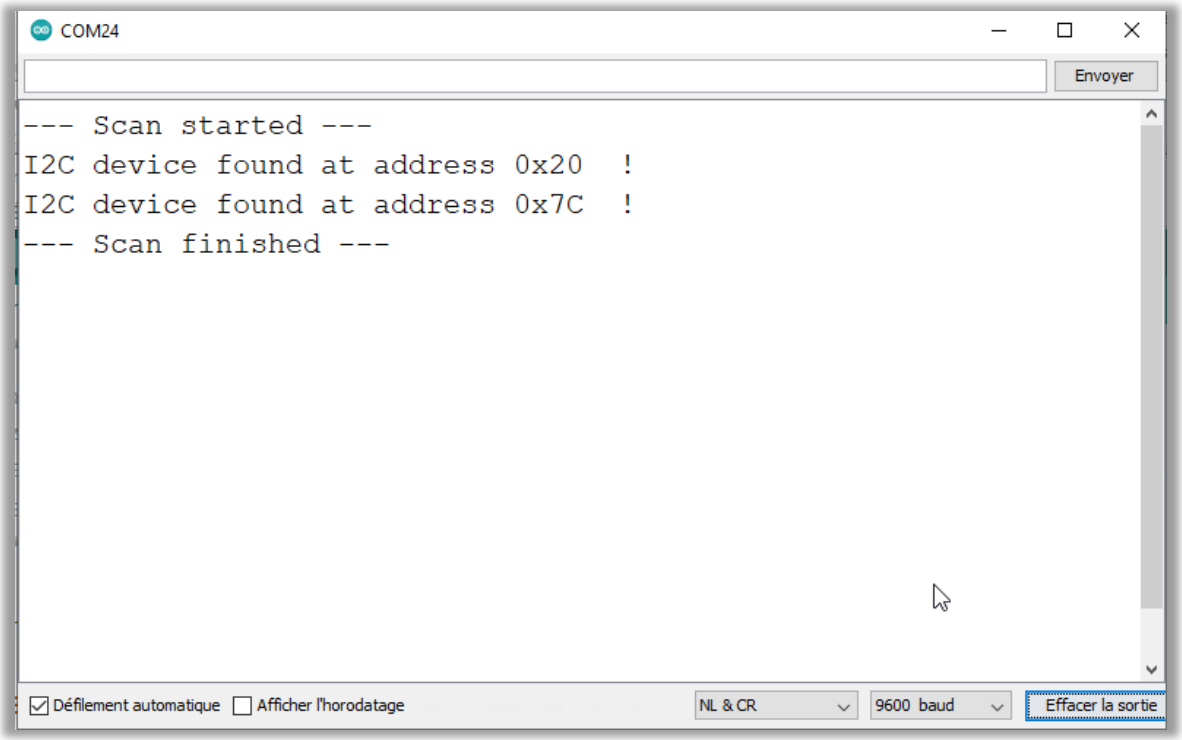
4. Si l'adresse I2C de l'écran n'est pas connue, on peut utiliser la bibliothèque suivante pour la trouver (I2CScanner en un seul mot) :



5. On va ensuite dans *Fichier*, *Exemples*, *I2CScanner* et on sélectionne *Scanner*



On téléverse ce programme dans la carte Arduino une fois que l'écran LCD est bien branché dans la carte et en ouvrant le moniteur série, on obtient l'adresse I2C. Utiliser la première.



L'adresse des modules I2C peut varier d'un fabricant à l'autre, mais elle reste la même pour un module. L'adresse I2C ne changera donc pas pour un même écran LCD. Il n'est donc pas nécessaire de refaire cette partie.

5. Bibliothèques

Lorsque l'on programme pour les cartes Arduino, on utilise déjà plusieurs fonctions prédéfinies et intégrées dans le logiciel. Cependant, il est possible de créer ses propres fonctions et de les partager. Pour ce faire, un contributeur (fabricant ou autre) regroupe les fonctions sous la forme d'une bibliothèque. En incluant une bibliothèque, on peut alors accéder à ces fonctions.

La bibliothèque utilisée dans ce programme est « LiquidCrystal_I2C ». Elle permet d'écrire sur un écran LCD (*Liquid Crystal Display* ou écran à cristaux liquides) en utilisant le protocole de communication I2C. il est plus pratique d'utiliser cette ce protocole de communication car l'on utilise alors que 2 broches sur la carte Arduino, au lieu de 6 avec la librairie « LiquidCrystal » préinstallée d'Arduino. L'écran LCD doit pouvoir le permettre par contre.

Voici une liste de certaines des fonctions de la bibliothèque « LiquidCrystal_I2C »[1] :

- **print()** : Affiche une chaîne de caractères ou des chiffres à l'écran à partir de la position du curseur.
- **clear()** : efface ce qui est écrit à l'écran
- **home()** : replace le curseur à la position (0,0)
- **noBlink()** : L'affichage du curseur est fixe (si le curseur est visible).
- **blink()** : Le curseur clignote (si le curseur est visible)
- **noCursor()** : désactive le curseur qui devient invisible.
- **cursor()** : active le curseur qui devient invisible.
- **scrollDisplayLeft()** :Décale le contenu affiché d'une colonne sur la gauche
- **scrollDisplayRight()** : Décale le contenu affiché d'une colonne sur la droite
- **leftToRight()** : Active l'affichage normal (annule l'effet introduit par rightToLeft())
- **rightToLeft()** : Affiche le texte de gauche à droite (à l'envers) depuis la position du curseur. "Hello!" est donc affiché "!olleH" depuis la position du curseur.
- **noBacklight()** : Désactive le rétro-éclairage
- **backlight()** : Active le rétro-éclairage
- **autoscroll()** : "Justifie le texte à droite" depuis la position du curseur
- **noAutoscroll()** : Justifie le texte à gauche depuis la position du curseur
- **setCursor(uint8_t colonne, uint8_t ligne)** : Modifie la position du curseur à l'écran pour le prochain affichage (de 0 à N-1). Attention, le premier argument est la **colonne** et le deuxième la ligne! Exemple : setCursor(2, 1); mettra le curseur à la troisième colonne de la 2e ligne (on commence toujours à 0!).

[1] https://wiki.mchobby.be/index.php?title=LCD-I2C-Fonctions

- **createChar(uint8_t, uint8_t[])** :Voir l'exemple disponible dans le menu *Fichier -> Exemples -> LiquidCrystal_I2C*. Permet de créer son propre caractère à afficher!

6. Hello, world!

Dans la plupart des livres de programmation en C, le premier programme qui est fait est généralement d'afficher le message : Hello, world !. Cela permet de voir si tout fonctionne bien. C'est un grand classique de la programmation

7. Schéma du circuit

Retourner l'écran LCD. Il y a 4 broches permettant de se brancher sur la carte Arduino. Les branchements à effectuer sont comme suit :

Écran LCD	Arduino
GND	N'importe quel GND de la carte
VCC	5V
SDA	A4
SCL	A5

Les broches A4 et A5 sont celles utilisées pour la communication I2C avec Arduino UNO.

8. Code

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x20,16,2);

void setup() {
  lcd.init();         // initialise l'écran LCD
  lcd.backlight();    //Ouvre la lumière  du rétro-éclairage
}
void loop() {
   lcd.setCursor(0,0);
   lcd.print("Hello, world!");    // affiche "Hello, world!"
    delay(500);
   lcd.setCursor(0,1);
   lcd.print("Arduino ");  //afficher « Arduino »
   delay(2000);
  }
```

Fonctions utilisées

```
#include <LiquidCrystal_I2C.h>
```

Cette ligne de code permet d'utiliser les fonctions de la bibliothèque *LiquidCrystal_I2C.*

```
LiquidCrystal_I2C lcd(0x20,16,2);
```

On crée un objet de type « LiquidCrystal_I2C » dont le nom est « lcd » qui possède 16 colonnes et 2 lignes. Le terme « 0X20 » est l'adresse hexadécimale utilisée par le protocole I2C pour ce type de composante. Il n'est pas nécessaire de le changer. Si l'écran ne fonctionne pas, on peut essayer de scanner pour l'adresse comme expliqué plus haut, ou voir la page https://forum.arduino.cc/index.php?topic=128635.0 pour trouver l'information pour trouver l'adresse exacte.

```
lcd.init();
```

Lorsque l'on utilise les fonctions d'une bibliothèque, il faut souvent indiquer d'abord le nom de l'objet, suivi d'un point et ensuite de la fonction.

Pour les autres fonctions, voir la description plus haut!

Défis :

- Modifier le texte écrit pour que «Arduino» soit centré (deux façons).
- Modifier le texte pour écrire autre chose! Question : que se passe-t-il avec les accents?
- Faire clignoter le texte avec des rythmes différents, ou pas du tout.
- Utiliser les autres fonctions de la bibliothèque pour :
  - Faire dérouler le texte
  - Afficher un nombre avec 5 décimales
  - Afficher du texte, puis un nombre sur la même ligne
- Ajouter un potentiomètre, lire sa valeur et l'afficher en pourcentage (100% = 255) sur l'écran!
- Créer son propre caractère !
- Tout autre chose que vous voudriez essayer!