

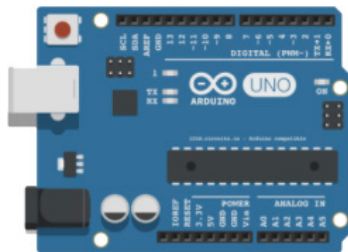
Fiche 12

Affichage cristal liquide (2 lignes)

Quatre fils de branchement mâles - femelles



Circuit Arduino Uno



Afficheur cristal liquide

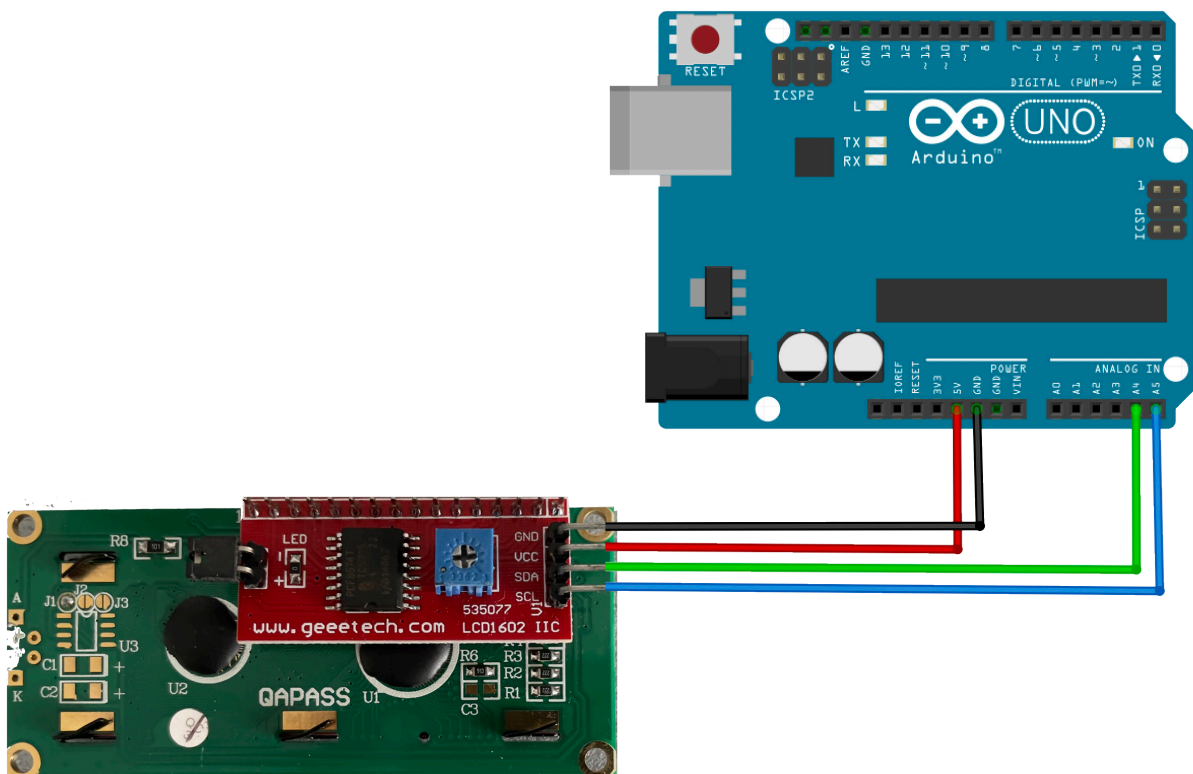
Dessus



Dessous

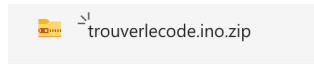


Branchement



Avant d'utiliser un afficheur cristal, il faut trouver son adressage. Cet adressage est différent selon les fabricants d'afficheurs cristal.

1- Télécharger le programme «trouverlecode.ino»



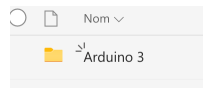
qui se trouve dans l'équipe

«Arduino»



Arduino

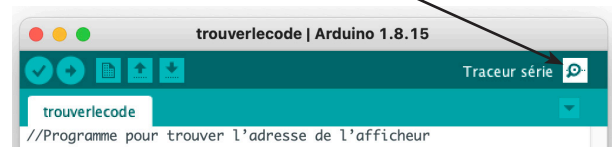
dossier «Arduino 3»



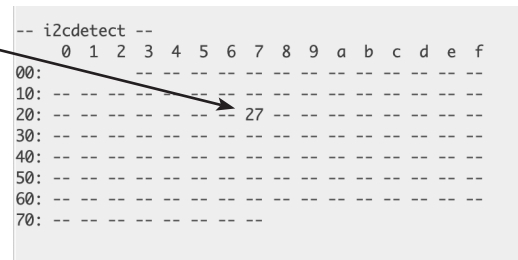
sur la plate-forme Teams.

2- Brancher l'afficheur en respectant le plan de la page précédente.

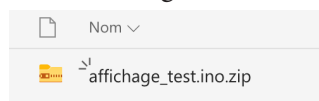
3- Envoyer le programme dans le module Arduino et sélectionnez le «Traceur série».



Sur l'écran du traceur série, le code d'adressage de l'afficheur apparaîtra de cette façon. Dans notre exemple, c'est le chiffre 27 qui est assigné à son adressage.



4- Télécharger le programme «affichage_test.ino» afin de vérifier si l'affichage fonctionne sur votre Afficheur cristal.



Fonctionnement du programme :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
void setup()
{
  lcd.init();           // initialize the lcd
  // on allume l'éclairage de fond.
  lcd.backlight();
  // on place le curseur sur la ligne 0 colonne 1
  lcd.setCursor(1, 0);
  lcd.print("salut les amis");
}
void loop()
{
  // on place le curseur sur la colonne 3 ligne 1
  lcd.setCursor(3, 1);
  // print the number of seconds since reset:
  // n'imprime que le premier chiffre du nombre
  lcd.print("temps «);
  lcd.print(millis() / 100);
}
```

On doit inclure ces deux bibliothèques pour le fonctionnement de l'afficheur.

Le chiffre 27 trouvé précédemment est indiqué de cette façon dans le programme. Le chiffre 16 indique le nombre de colonnes et le chiffre 2 indique le nombre de lignes de l'afficheur.

La commande `lcd.setCursor()` sert à placer le premier caractère de l'impression.

La commande `lcd.print()` imprime le texte ou les données à l'écran.

`lcd.setCursor(colonne, ligne);`

position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ligne 0																
ligne 1																