

NOM :

CLASSE :

PRÉNOM :

Condition :

- Travail individuel ; durée 3 heures

Matériel :

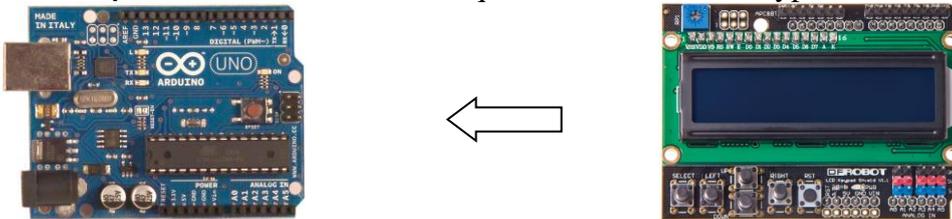
- un ordinateur avec le logiciel Arduino installé
- une maquette Arduino Uno avec son shield « *clavier-lcd* »

Documents :

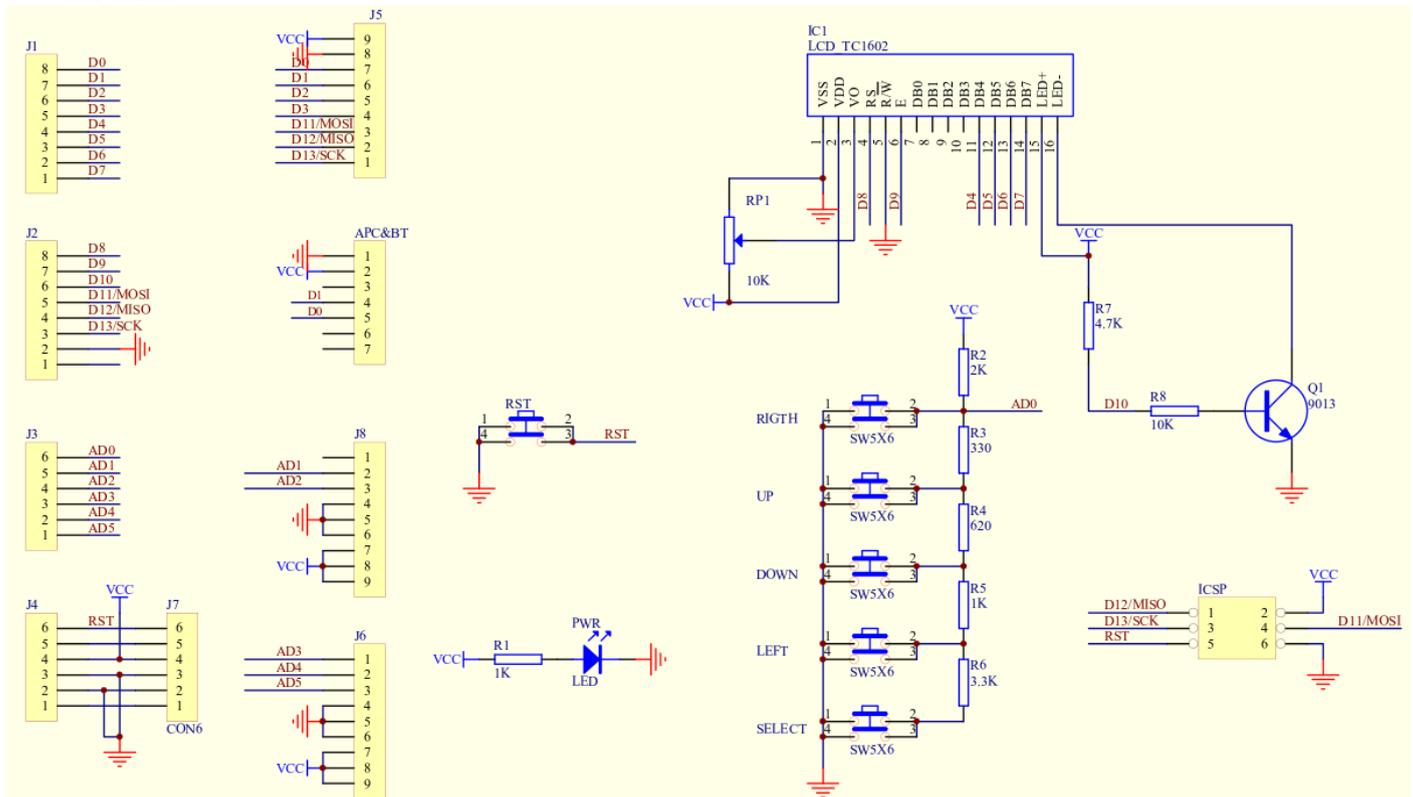
- le sujet du TP
- le cours sur la carte Arduino et sa programmation
- le cours sur l'afficheur LCD

1. Présentation

On utilise la maquette Arduino Uno et la maquette shield « LCD Keypad » de chez DFROBOT.



Cette carte shield est composée d'un afficheur LCD standard, de 5 boutons poussoirs et de quelques picots pour des entrées-sorties analogiques ou numériques. Le schéma de la carte shield LCD est le suivant :



Hors tension connecter la carte shield « interrupteurs leds » sur la carte Arduino. Prenez soin de ne pas casser des broches ! Connecter ensuite l'Arduino au PC avec un câble USB.

2. L'afficheur LCD

- ✓ A partir du schéma structurel de la carte shield LCD, indiquer si l'afficheur LCD est utilisé en 4 bits ou en 8 bits.

3. Utilisation d'une bibliothèque

Il est long de créer des fonctions permettant d'utiliser un afficheur. Il est plus simple d'utiliser une **bibliothèque** ou **librairie** (library en anglais). Les bibliothèques sont développées par les constructeurs de carte Arduino ou encore par des particuliers qui mettent à disposition leur programme afin de faciliter la programmation.

Sous le logiciel Arduino, cliquez sur **File, Example, Liquid Crystal** puis **HelloWorld**. Le programme suivant, nettoyé des commentaires anglais, s'ouvre :

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,                ); // brochage de RS, E, D4, D5, D6, D7

void setup() {
  lcd.begin(16, 2); // afficheur de 16 colonnes et 2 lignes
  lcd.print("hello, world!");
}
void loop() {
  lcd.setCursor(0, 1); // curseur première colonne deuxième ligne
  lcd.print(millis()/1000);
}
```

La première ligne du programme, `#include <LiquidCrystal.h>`, prévient le compilateur que vous allez utiliser la bibliothèque `LiquidCrystal.h`.

La ligne `LiquidCrystal lcd(8,);` crée un objet appelé `lcd` de la classe `LiquidCrystal` en passant en paramètres le brochage de l'afficheur : RS, E, D4, D5, D6, D7.

- ✓ Modifiez les numéros de broche affectées à l'afficheur dans le programme exemple. Compilez le, programme et testez-le.
- ✓ Quelle instruction permet d'écrire sur l'afficheur ?
- ✓ Quelle instruction permet de sélectionner la case où s'affiche le message sur l'afficheur ?
- ✓ Enregistrez le programme sous un autre nom et modifiez-le afin d'écrire « BIENVENUE » centré sur la première ligne et « EN STI2D SIN » centré sur la deuxième ligne de l'afficheur.

Tester et faire valider par le professeur.

Le contenu complet des fonctions écrites dans la bibliothèque `LiquidCrystal` est disponible à l'adresse <http://arduino.cc/en/Reference/LiquidCrystal> .

En utilisant ce site, modifiez le programme précédent afin de faire défiler à l'infini, de gauche à droite puis de droite à gauche, le message sur l'afficheur.

Tester et faire valider par le professeur.

4. Lire les Boutons-Poussoirs

L'objectif est d'écrire un programme de test du bon fonctionnement des boutons poussoirs de la carte shield. Le programme écrit sur l'afficheur LCD :

- « TEST BOUTONS » centré sur la première ligne de l'afficheur ;
- « SELECT » ou « LEFT » ou « UP » ou « DOWN » ou « RIGHT » centrés sur la deuxième ligne si un bouton est appuyé ;
- « AUCUN APPUI » centré sur la deuxième ligne si aucun des boutons n'est appuyé ;

Les boutons poussoirs sont câblés sur l'entrée analogique A0. Le potentiel sur cette broche varie en fonction du bouton appuyé.

- ✓ Écrire un premier programme qui affiche le résultat de la conversion Analogique Numérique de l'Arduino en cas d'appui sur chacun des boutons (Notez ces valeurs).
- ✓ Écrire le programme complet de test des boutons. Tester et faire valider par le professeur.

5. Programmer une horloge

L'objectif est d'écrire un programme permettant d'afficher l'heure sous la forme heure : minute : seconde.

Une première version utilisant la fonction delay() est la suivante :

```
int seconde=0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  seconde = seconde + 1;
  Serial.println(seconde);
  delay(1000);
}
```

- ✓ Tester et préciser combien de fois par seconde l'affichage est modifié.

.....
Le code suivant utilise la fonction millis()

```
unsigned long refTemp = 0;
unsigned long top;
int seconde;

void setup() {
  Serial.begin(9600);
}

void loop() {
  top = millis(); //lecture du temps actuel
  if (top - refTemp >= 1000) {
    refTemp = millis(); //lecture du temps de référence
    seconde = seconde + 1;
  }
  Serial.println(seconde);
  delay(100);
}
```

- ✓ Comparer le fonctionnement des deux programmes. Donner l'avantage de millis() si on voulait lire l'état d'un bouton poussoir toutes les 200ms.

- ✓ Modifier le programme pour que l'affichage des secondes soit sur le lcd et évolue de 0 à 59.



Tester et faire valider par le professeur.

- ✓ Rajouter maintenant le comptage des minutes de 0 à 59.

Rajouter le comptage des heures de 0 à 23. **Tester et faire valider par le professeur.**