

Exercice 1

L'objectif de cet exercice est d'écrire un programme Arduino qui permet d'afficher « Hello World ! » sur l'écran LCD de la carte ISEN. Pour cela, il faudra d'abord télécharger et installer les compléments librairies de la carte ISEN. Les fichiers ainsi que la procédure à suivre sont sur le portail du cours. Ensuite :

- Inclure les fichiers suivants dans votre programme:

```
// Include ISEN libraries
#include <ISEN.h>
#include <LabIsen-2009.h>
#include <ISL12026.h>
#include <MCP23X08.h>

// Include Arduino libraries
#include <LiquidCrystal.h>
```

- Déclarer une variable globale :

```
ISENLiquidCrystal lcd;
```

- Appeler les fonctions de la variable « lcd » (elles ne sont pas dans le bon ordre) :
 - `lcd.clear()` : permet d'effacer le contenu de l'écran.
 - `lcd.print("Hello World!")` : permet d'afficher le message sur l'écran.
 - `lcd.setCursor(2,1)` : permet de positionner le curseur à la colonne 2 et la ligne 1 (les indexes commencent à 0).
 - `lcd.begin(16,2)` : permet d'initialiser le fonctionnement de l'écran avec 16 colonnes et 2 lignes
- Modifier le programme pour qu'il affiche « Hello » et « World ! » sur deux lignes séparées.
- A quelle catégorie de langages de programmation appartient cette syntaxe ?

Exercice 2 :

Ecrire un programme Arduino qui affiche un compteur sur l'écran LCD.

- La valeur initiale est 0.
- Le bouton associé à l'interruption 0 permet d'incrémenter la valeur de ce compteur et d'actualiser ainsi l'affichage sur l'écran. Utiliser la fonction `attachInterrupt()`.

Exercice 3 :

Ecrire un programme Arduino qui simule un chronomètre précis au dixième de seconde en vous basant sur la fonction « delay »:

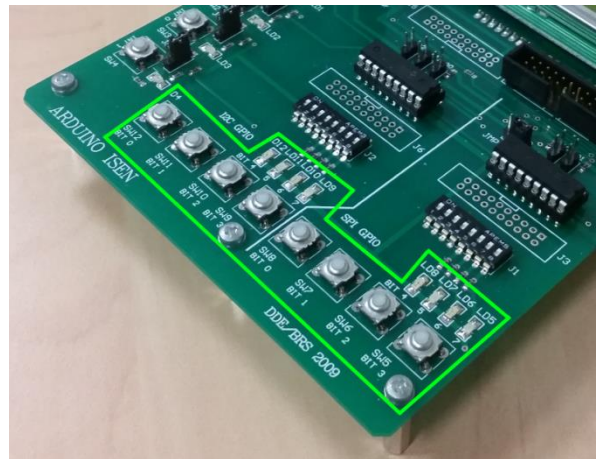
- La valeur est affichée au format : HH:MM:SS :C.
- Le bouton 1 (rouge) permettra de démarrer le chrono
- Le bouton 2 (orange) permettra de mettre le chrono en pause
- Le bouton 3 (vert) permettra de réinitialiser le temps à zéro.

Est-ce que ce chrono mesure le temps correctement ? Justifiez.

Exercice 4 :

La carte ISEN inclut un ensemble de Leds et boutons (numérotés de 5 à 12) programmables en INPUT/OUTPUT simultanément et manipulables avec les bibliothèques fournies.

En effet, le nombre de pins numériques sur la carte Arduino limite le nombre de branchements et d'opérations réalisables avec ces derniers. Ils sont en plus programmables en INPUT ou en OUTPUT mais pas les deux simultanément.



- Compiler le programme suivant et le transférer vers l'Arduino :

```
// Include ISEN libraries
#include <ISEN.h>
#include <LabIsen-2009.h>
#include <ISL12026.h>
#include <MCP23X08.h>

// Include Arduino libraries
#include <Wire.h>

IOstd io;

void setup() {
  io.begin();
  io.clear();
}

void loop() {
  for(int i=5; i<=12; i++)
  {
    io.digitalWrite(i, io.digitalRead(i));
  }
  delay(50);
}
```

- Que fait ce programme? Commenter les différentes déclarations et instructions.

On souhaite maintenant réaliser un convertisseur « Binaire » vers « Décimal ». Pour cela :

- La valeur décimale sera stockée sur 8 bits (1 octet). Sa valeur initiale est 0.
- L'état des boutons est stocké dans un tableau de 8 booléens. Il représente l'état des 8 bits de la valeur binaire. Les 8 cellules sont initialisées à 0.
- Ecrire une fonction qui convertit le tableau en valeur décimale. Ex : $00001101_2 = 13_{10}$
- Les boutons serviront à allumer/éteindre les Leds associées et de mettre à 0 ou 1 le bit associé dans le tableau binaire.
- L'écran LCD affiche à chaque fois les deux valeurs, binaire et décimale.
Exemple : L'écran LCD Affiche $00001101 = 13$
Si l'utilisateur appuie sur le bouton associé au 5^{ème} bit, l'affichage est réactualisé $00011101 = 29$