



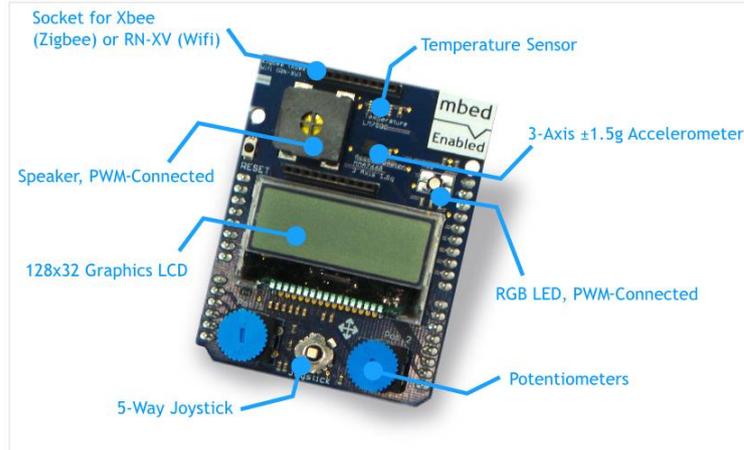
Carte Application Shield TP N°1 Création de menus

Table des matières

I.	Présentation de la carte Application Shield.....	1
II.	L'afficheur LCD	2
a)	Test du fonctionnement de l'afficheur LCD.....	2
b)	Utilisation du formatage de la fonction printf :.....	3
III.	Le joystick.....	3
a)	Recherche des correspondances entre le joystick et les broches	3
b)	Affichage sur l'écran LCD	3
IV.	Un menu avec le joystick	3
a)	Questions préliminaires.....	4
V.	Cahier des charges	4
b)	Affichage de menus déroulants sur l'écran LCD.....	4
VI.	Diagramme états/transitions.....	4
VII.	Code sources.....	6
	Ecran	6
	Joystick.....	6
	Menus	6

I. Présentation de la carte Application Shield

C'est une carte compatible avec le site mbed.



Les caractéristiques sont :

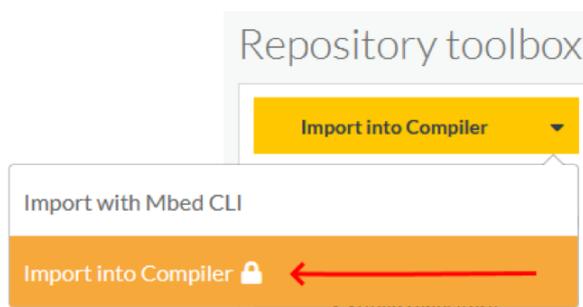
- Un écran LCD de 128x32 Graphics LCD
- un joystick 5 positions
- 2 x Potentiomètres
- Un Haut-Parleur connecté en PWM
- un accéléromètre 3 axes 1.5g Accelerometer
- une led trois couleurs RGB LED, connecté en PWM
- un capteur de température
- une Socket pour (Zigbee) ou RN-XV (Wifi)

II. L'afficheur LCD

a) Test du fonctionnement de l'afficheur LCD

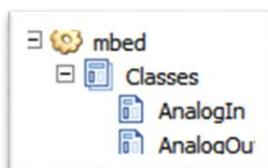
Ce deuxième TP C++ va permettre de tester le fonctionnement de l'afficheur LCD sur la carte de développement adaptée à la « Nucléo-board ».

- Importer le projet : <https://developer.mbed.org/users/rtk/code/app-shield-LCD/> (cliquer sur *Import Into Compiler*)

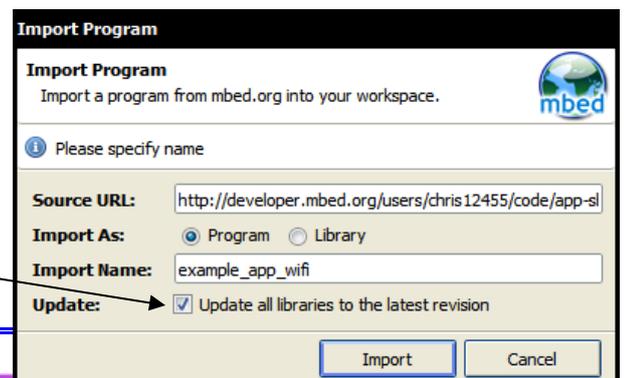


Attention : comment importer correctement les exemples ?

Pour obtenir les classes sous mbed comme ceci :



Il faut cocher Update all libraries to the latest revision





- Ouvrir le projet app_shield_lcd, compiler et tester le fonctionnement.
- Modifier le programme pour faire apparaître votre nom sur une première ligne et votre prénom sur la ligne du dessous, déterminer le rôle de la fonction « locate ».
- Importer et tester pas à pas avec la carte mbed

b) Utilisation du formatage de la fonction printf :

Développer un programme permettant d'écrire sur une première ligne de l'afficheur : « Secondes écoulées » et sur une seconde ligne le nombre de secondes qui va s'incrémenter à chaque boucle (une temporisation de valeur appropriée sera incluse dans la boucle pour assurer le comptage des secondes).

III. Le joystick

a) Recherche des correspondances entre le joystick et les broches

Au dos de la carte, il est indiqué la liste des ressources utilisables.

- Remplissez ce tableau (sur papier libre)

Touche	Nom de la broche Nucleo

- Les ports correspondant aux touches sont-elles analogiques ou numériques ?
- Sont-elles en entrée ou en sortie ?
- Quelle devrait-être alors la classe de ces broches ?

Vous prendrez malgré tout, DigitalIn.

b) Affichage sur l'écran LCD

- Importer le projet : https://os.mbed.com/users/rtk/code/TP3_Partage_app-shield-joystick-2021/ (cliquer sur )
- Déclarer les quatre autres boutons.
- Modifier le programme pour afficher le nom de la touche sur l'écran LCD
- Importer et tester pas à pas avec la carte mbed1);

IV. Un menu avec le joystick



a) Questions préliminaires

- Que veut dire `C12832 lcd(D11, D13, D12, D7, D10) ; ?`
- Quelle est la taille en pixel d'un pixel en hauteur ?

En déduire le nombre de pixels à partir du haut, pour écrire sur la 2^{ème} ligne.

Quel est alors la commande `locate(x,y)` pour écrire au début de la 2^{ème} ligne (attention le point le plus en haut est à la position 0 et non 1)

- Que fait les fonctions `cls()` et `locate()` ?

V. Cahier des charges

b) Affichage de menus déroulants sur l'écran LCD

Le joystick devra permettre de se déplacer verticalement dans trois menus :

- Matin
- Midi
- Soir

A l'intérieur de chaque menu on pourra se déplacer horizontalement dans les sous-menus :

- Petit déjeuner
- Déjeuner
- Diner

Dans chaque sous-menu un message sera envoyé à l'écran dans les conditions ci-dessous quand on appuiera sur le bouton de validation (bouton central) :

Menu matin :

- Sous_menu : Petit déjeuner on affichera « OK »
- Sous_menu : Déjeuner on affichera « Trop tôt »
- Sous_menu : Diner on affichera « Trop tôt »

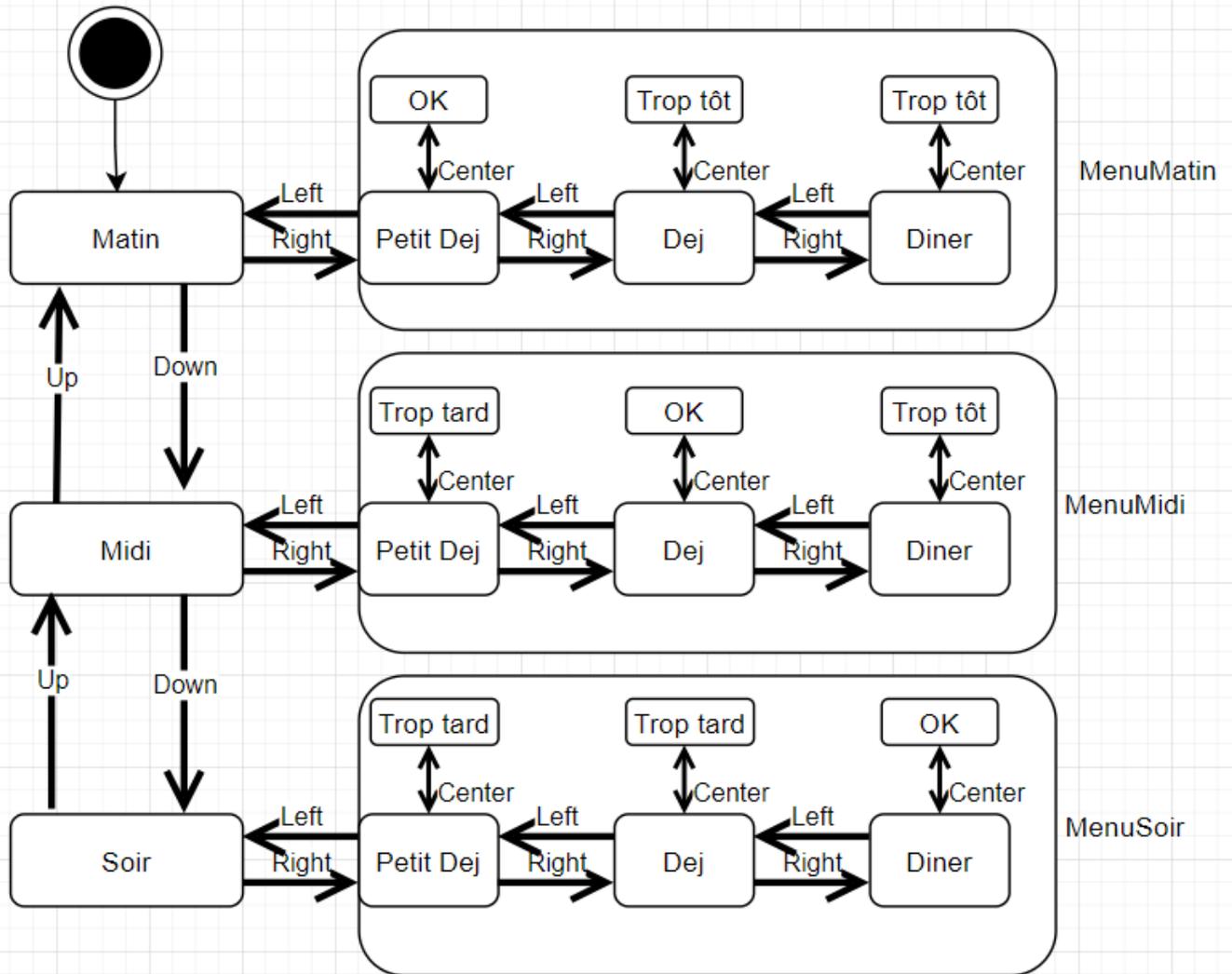
Menu midi :

- Sous_menu : Petit déjeuner on affichera « Trop tard »
- Sous_menu : Déjeuner on affichera « OK »
- Sous_menu : Diner on affichera « Trop tôt »

Menu soir :

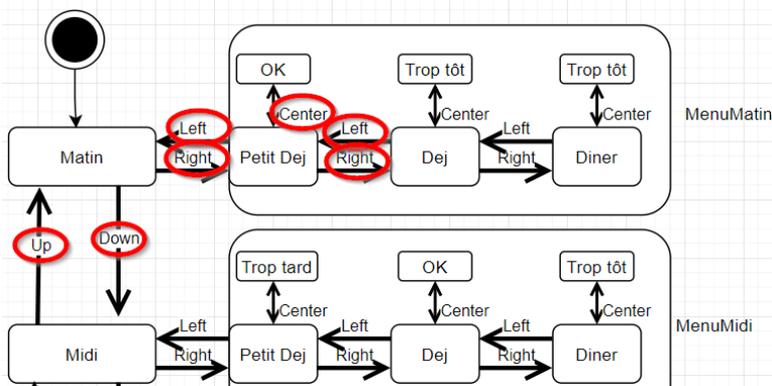
- Sous_menu : Petit déjeuner on affichera « Trop tard »
- Sous_menu : Déjeuner on affichera « Trop tard »
- Sous_menu : Diner on affichera « OK »

VI. Diagramme états/transitions



Le début de l'implémentation se trouve à https://os.mbed.com/users/rtk/code/TP3_2021-Menu/ :

Attention : les événements uniquement en rouge sont



Travail à faire :

- Testez ce code
- Modifiez-le pour qu'il gère le menu du matin (sous-menu DEJ et DINER)
- Terminez avec les autres menus



VII. Code sources

Ecran

```
#include "mbed.h"
#include "C12832.h"

// Using Arduino pin notation
C12832 lcd(D11, D13, D12, D7, D10);

int main()
{
    int j=0;
    lcd.cls();
    lcd.locate(0,3);
    lcd.printf("mbed application shield!");

    while(true) { // this is the third thread
        lcd.locate(0,15);
        lcd.printf("Counting : %d",j);
        j++;
        wait(1.0);
    }
}
```

Joystick

```
1 #include "mbed.h"
2 #include "C12832.h"
3
4 DigitalIn up(A2);
5 C12832 lcd(D11, D13, D12, D7, D10);
6
7 int main()
8 {
9
10     while (1) {
11         lcd.locate(0,8);
12         if (up) lcd.printf("Up");
13         else lcd.printf("Aucune touche");
14
15     }
16 }
17
18
19
20
```

Menus

```
#include "mbed.h"
#include "C12832.h"
DigitalIn up(A2);
DigitalIn down(A3);
DigitalIn left(A4);
```



```

DigitalIn right(A5);
DigitalIn center(D4);
C12832 lcd(D11, D13, D12, D7, D10);
enum {MATIN,MIDI,SOIR, PETIT, DEJ,DINER} menuPrincipal = MATIN;

int menu_precedent = MIDI;
char message_precedent[]="unMessageAuHasard";

void AttendreRelachementBoutons()
{
  if (up) while (up) wait(0.01);
  else if (down) while (down) wait(0.01);
  else if (left) while (left) wait(0.01);
  else if (right) while (right) wait(0.01);
  else if (center) while (center) wait(0.01);
}

void AttendreNouvelAppuiCenter()
{
  AttendreRelachementBoutons();
  while (!center) wait(0.01);
}

void Afficher_Message(char *Message)// Pour ne pas avoir de scintillement
{
  if (strcmp(message_precedent,Message) ==0) return;
  strcpy(message_precedent,Message);
  lcd.cls();
  lcd.locate(0,8);
  lcd.printf(Message);
  AttendreRelachementBoutons();
}

void Afficher_Menu(int Menu)
{
  switch(Menu) {
    case MATIN :
      Afficher_Message("Matin");
      break;
    case MIDI :
      Afficher_Message("Midi");
      break;
    case SOIR :
      Afficher_Message("Soir");
      break;
    case PETIT :
      Afficher_Message("Petit déjeuner");
      break;
    case DEJ :
      Afficher_Message("Dejeuner");
  }
}

```



```

        break;
    case DINER :
        Afficher_Message("Diner");
        break;
    }
}

void Menu_Matin(int SousMenu)
{
    //main
    bool Sortir = false; // On sort du sous-menu
    while (!Sortir) {
        //while
        Afficher_Menu(SousMenu);

        switch (SousMenu) {
            //switch
            case PETIT :
                // évènements
                if (left) {
                    menuPrincipal = MATIN;
                    Sortir=true;
                }
                if (right) SousMenu=DEJ;
                if (center) {
                    Afficher_Message("Ok");
                    AttendreNouvelAppuiCenter();
                }
                break;
            case DEJ :
                // évènements
                if (left) SousMenu=PETIT;
                break;
            case DINER :
                break;
        }
    }
}

    }//while
} //main

int main()
{
    while (1) {
        Afficher_Menu(menuPrincipal);
    }
}

```



```
switch (menuPrincipal) {  
  case MATIN :  
    // évènements  
    if (down) menuPrincipal = MIDI;  
    if (right) Menu_Matin(PETIT);  
    break;  
  case MIDI :  
    if (up) menuPrincipal = MATIN;  
    break;  
  case SOIR :  
    break;  
}  
  
}  
  
}
```



BTS SN – EC

Prototypage rapide avec STM32 Nucleo

TP : carte « Application Shield » N°1

