STM32 I2C

Objectifs : Mise en œuvre du bus I2C

Matériel : Ce TP utilise une NUCLEO-F411RE, mais n'importe quelle autre carte NUCLEO convient.
Logiciel : MBED

Le microcontrôleur STM32 dispose de bus de communication synchrones SPI et I2C.
Une première approche permet de découvrir le bus I2C à travers la mise en œuvre d'un « expander » de bus PCF8574A.
Ce TP propose ensuite la mise en œuvre du bus I2C dans le cadre d'une liaison STM32 vers un afficheur LCD.

Ressources I2C
https://www.planete-sciences.org/robot/ressources/electronique/protocoleI2C/index.html
https://fr.wikipedia.org/wiki/I2C

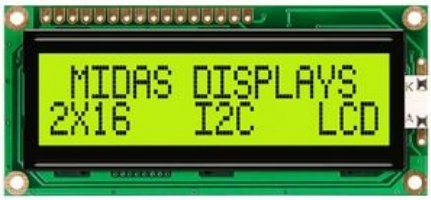
1 Exercices de découverte

(utiliser comme ressource documentaire le site wikipedia)

Que signifie "I2C" ? Quelle entreprise est à l'origine de ce bus de communication ?
Le bus I2C est half-duplex, expliquer ce terme.
Le bus I2C est synchrone, expliquer ce terme.
Expliquer la signification de : Le bit 1 est récessif , le bit 0 est dominant
Comment cela est il rendu possible ?
Quel est l'intérêt de la structure drain ouvert et résistance de pullup ?
Comment choisir la valeur des résistances de pullup ?
Combien d'esclaves peuvent cohabiter sur un bus I2C ?
Quelle est le débit maximum théorique (en octets/seconde)
Qui initie un échange de données ?
Expliquer ce qu'est :
- Une condition start
- Une condition stop
- Un acknowledge

Comment est reconnue une demande de lecture, une demande d'écriture ?
Combien de bus I2C sont disponibles sur le STM32 équipant votre carte Nucleo ?
Repérer les numéros des broches correspondantes sur le connecteur Arduino

2 Mise en œuvre de l'afficheur LCD



Afficheur LCD 2x16 I2C
http://fr.farnell.com/midas/mc21605c6w-sptlyi-v2/afficheur-alphanumerique-16x2/dp/2748647?st=lcd%20i2c

MBED regorge de pilotes de périphériques et en particulier de périphériques I2C . Cette première partie du TP vous fera découvrir comment rechercher des ressources logicielles sur MBED.
Vous allez mettre en œuvre une afficheur LCD communiquant par I2C à l'aide d'une bibliothèque MBED.
L'usage d'une bibliothèque rend transparent l'usage du bus I2C, les échanges de données s'effectuant par l'appel de méthodes d'écriture et de lecture déjà définies.
Le TP montre la simplicité de la mise en œuvre d'un bus I2C.

Sur le site de Midas www.midasdisplays.com rechercher le datasheet de l'afficheur
Réaliser à la main un schéma structurel d'interconnexion entre la carte Nucleo et l'afficheur.
- Alimentation VDD / VSS
- bus I2C et résistances de pullup
- potentiomètre de réglage du contraste
- SA0, SA1et CSB seront connectés à VSS

L'adresse I2C de l'afficheur est fixée par le fabriquant mais dépend également des niveaux sur les broches SA1, SA0. Le contrôleur LCD est un circuit rw1063. Rechercher le datasheet de ce circuit ici :
https://www.crystalfontz.com/controllers/RockWorks/RW1063-0B-002/227
A la page 28 est indiqué comment calculer l'adresse I2C.

Réaliser le câblage suivant le schéma réalisé.
Mettre sous tension et ajuster ensuite le potentiomètre de manière à rendre légèrement grise la ligne du haut de l'afficheur.

3 Logiciel pilote et test

L'afficheur est alphanumérique, il peut afficher les codes ASCII « imprimables ».
Il est organisé en deux rangées de 16 caractères.
Indiquer d'après la documentation quelles sont les adresses internes de ces deux rangées

Rechercher le programme de démonstration sur MBED dans cookbook paragraphe LCDs and Displays sélectionner Text LCD Enhanced, ajouter cette démonstration (TextLCD_HelloWold2) au compilateur, en cochant library update.

Le programme doit être adapté à l'afficheur et à la carte NUCLEO.

Modifier l'instanciation de la classe I2C en modifiant les noms des ports SDA et SCL que vous recherchez dans PinNames.h (voir premier TP STM32).

Le bus étant I2C l'instanciation SPI sera supprimée

L'instanciation de la classe Text_LCD_…… permet de configurer l'interface I2C vers l'afficheur. La syntaxe est la suivante :

TextLCD_I2C_N lcd(adresse bus I2C,adresse I2C de l'afficheur, format de l'afficheur, type de contrôleur) ;
Les paramètres de TextLCD_I2C_N se trouvent dans TextLCD_Config.h

L'adresse du bus I2C du STM32 est celle de l'instance I2C donc &i2c_lcd
Le contrôleur est compatible AC780, nous avons vu que son adresse est 0x78 le paramètre adresse sera AC780_SA0
L'afficheur est sur 2 lignes de 16 caractères, le paramètre sera TextLCD::LCD16x2
Le contrôleur est compatible AC70, le paramètre sera TextLCD::AC780

L'instanciation de la classe TextLCD_I2C_N se fera par :

```
TextLCD_I2C_N lcd(&i2c_lcd, AC780_SA0, TextLCD::LCD16x2, NC, TextLCD::AC780);
```

Le programme de démonstration doit maintenant fonctionner. Testez le…
De nombreuses lignes sont maintenant inutiles, remplacez le programme par celui ci :

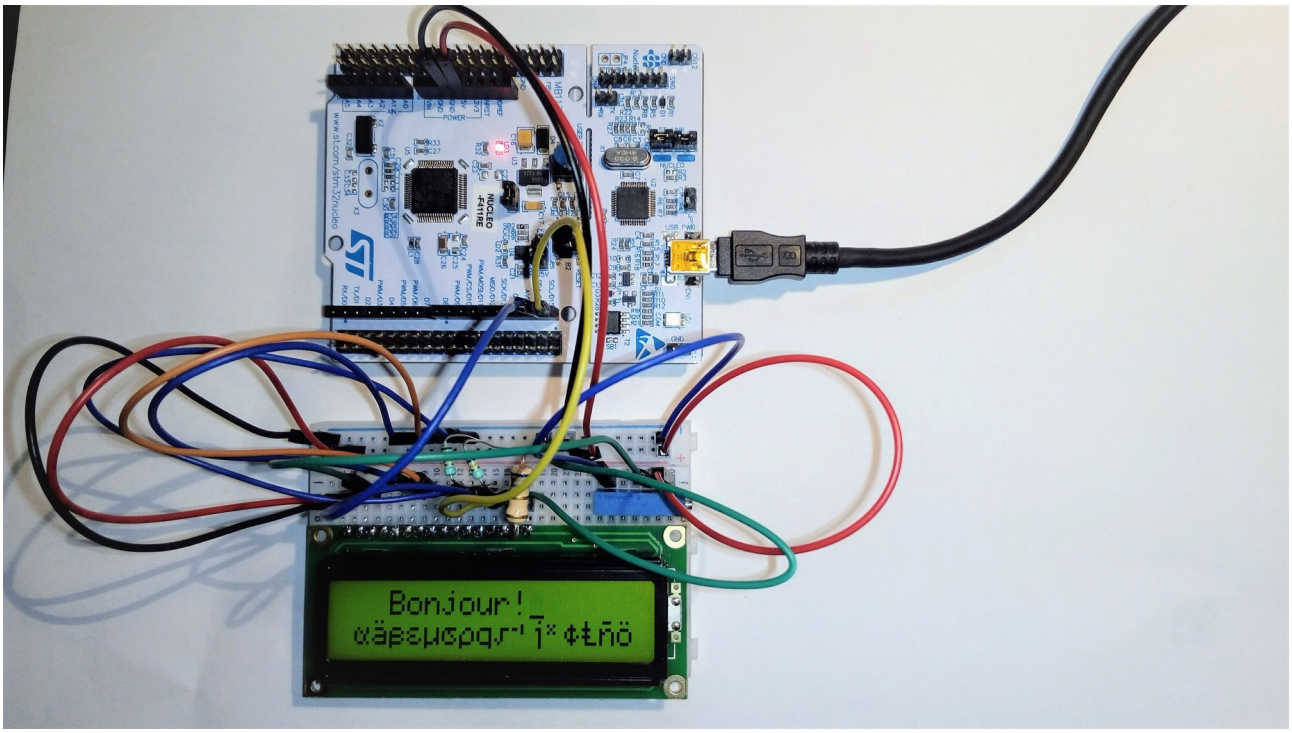
```
#include "mbed.h"
#include "TextLCD.h"

I2C i2c_lcd(I2C_SDA,I2C_SCL); // SDA, SCL
TextLCD_I2C_N lcd(&i2c_lcd, AC780_SA0, TextLCD::LCD16x2, NC, TextLCD::AC780);

int main() {
    lcd.setBacklight(TextLCD::LightOn);     // backlight
    lcd.setCursor(TextLCD::CurOn_BlkOn);    // cursor blink
    lcd.cls();                              // clear screen
    lcd.setAddress(0,1);                    // second line, left
    for(int i=0;i<16;i++) lcd.putc(224+i);  // for greeks and stranges caracters
    lcd.setAddress(2,0);                    // first line, third place
    lcd.printf("Bonjour!");
    while(1);
```

```
}
```

Dans la bibliothèque comment sont organisées les adresses des caractères (lignes/colonnes) ?
Expliquer le rôle de chaque méthode appelée dans ce programme.



3.1 Exercice :

Réaliser un voltmètre affichant la tension sur l'entrée analogique A0.

3.2 Exercice :

Réaliser un programme recopiant sur l'afficheur un texte saisi dans un terminal lors de l'appui sur la touche enter du clavier du PC.

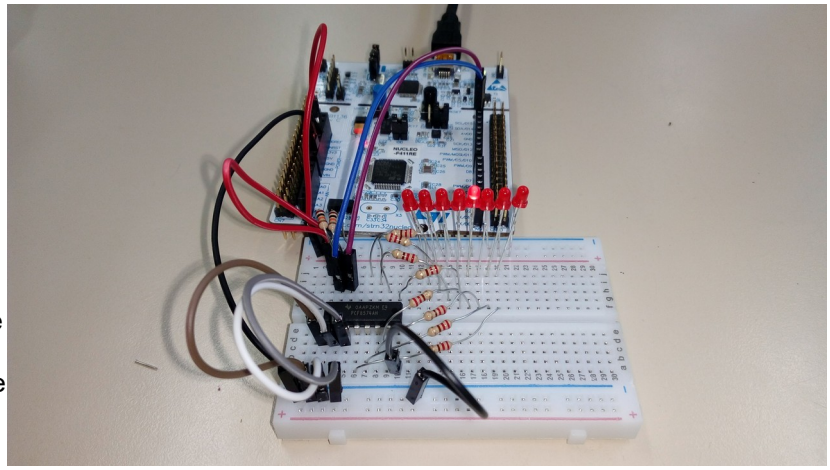
3.3 Exercice :

Réaliser un chronomètre heure, minutes, secondes, 1/10 secondes.
L'affichage sera déclenché par une interruption Ticker toutes les 100mS.
Dans un deuxième temps, le chronométrage sera déclenché et arrêté par une interruption sur le bouton bleu.

4 Etude approfondie du bus I2C

La deuxième partie du TP propose une étude approfondie du protocole I2C et la réalisation d'un driver pour un composant I2C simple.

Expander I2C Texas-Instruments PCF8574A
http://www.ti.com/product/PCF8574A



Mise en œuvre le bus I2C1 du STM32 connecté à un PCF8574A. Huits LEDs et leurs résistances seront connectées sur les sorties de l'expander.
Le courant dans chaque LED sera de 6mA, la tension VF des diodes est de 2v. Définir une valeur normalisée dans la série E12 pour les résistances.

Expliquer ce qu'est un expander de bus.
Réaliser sur papier un schéma de câblage entre votre carte Nucleo, le PCF8574A et les composants discrets.

4.1 Classe I2C de MBED

MBED dispose d'une classe I2C. Rechercher dans le manuel de référence de MBED, dans Docs/Mbed OS/ API, drivers, la documentation de la classe I2C.
*Le constructeur d'une classe est une méthode portant **le même nom** que la classe qui sera exécuté automatique lors de l'instanciation de chaque objet.Une fois affichée la référence de la classe I2C, on accède à la description des méthodes en cliquant sur leur nom.*

I2C Class Reference

- Décrire les paramètres du constructeur de la classe I2C.
- Indiquer le rôle des méthodes frequency, start et stop
- La méthode write est surchargée, décrire les différentes manières de l'utiliser.
- Faire de même pour la méthode read.

A partir de la documentation du PCF8574A et de la classe I2C.

- Indiquer l'adresse I2C du PCF8574A lorsque les entrées A0 ,A1, A2 sont reliées à VSS (0v) (voir page 12)
- Réaliser l'algorigramme d'une fonction

```
void ecrit_I2C(unsigned char add,unsigned char data) ;
```

qui émet data sur le bus I2C à adresse add: ex : ecrit_I2C(0x12,23) ; emet 23 à l'adresse 0x12
Afin de tester cette fonction, réaliser un programme faisant clignoter simultanément les huit LEDs avec une période de 1S (on pourra utiliser une méthode wait(1) pour la temporisation).

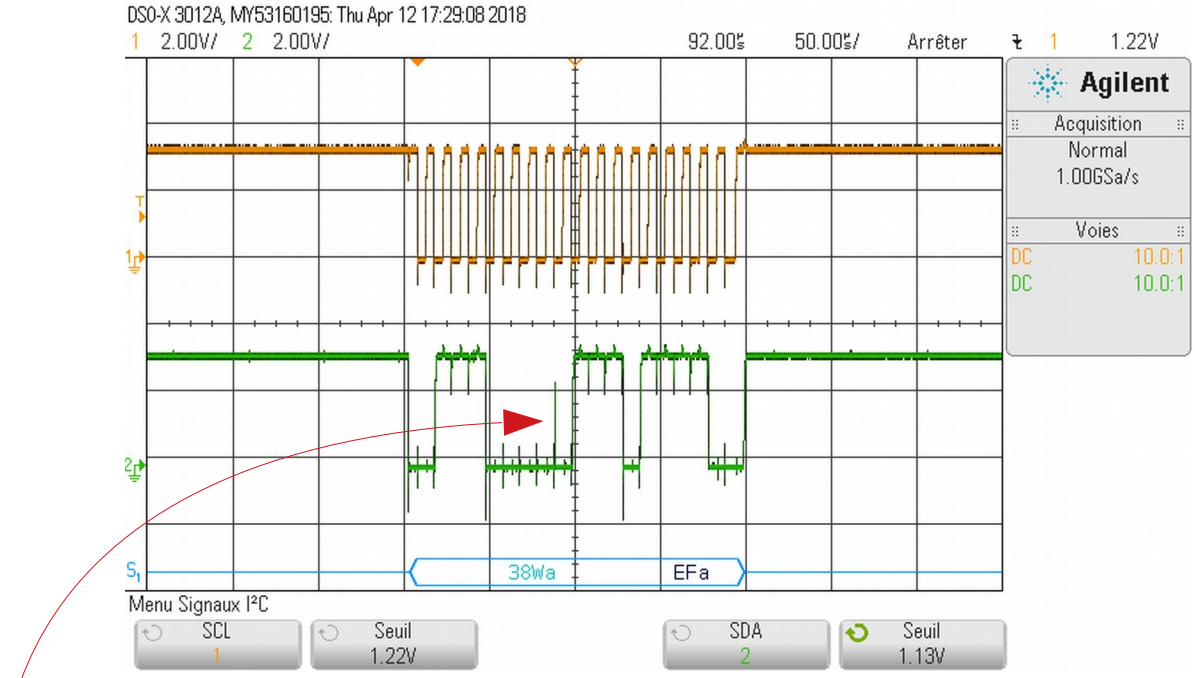
4.2 Exercice

Réaliser un chenillard allez-retour, une seule LED allumée à la fois avec une temporisation de 500mS par allumage.

4.3 Exercice

A l'aide d'un oscilloscope avec analyseur de trame I2C ou d'un analyseur logique, relever les trames I2C lors d'une écriture sur PCF8574A. Repérer les conditions start et stop, les bits d'acquittement , l'adresse I2C ainsi que la donnée transmise.

Résultat attendu :



Start est repéré par le front descendant sur SDA suivi d'un front descendant sur SCL
Stop est repéré par le front montant sur SCL suivi d'un front montant sur SDA
L'adresse 0x38 et la donnée (ici 0xEF) sont facilement identifiables en binaire sur SDA
Il y a bien un 9ieme coup d'horloge après l'émission de l'adresse pour l'acquittement. Lorsque le maître fait monter le signal d'horloge il place un bit récessif sur SDA, l'esclave place alors un bit dominant. On voit très bien apparaître une impulsion très courte.

4.4 Exercice

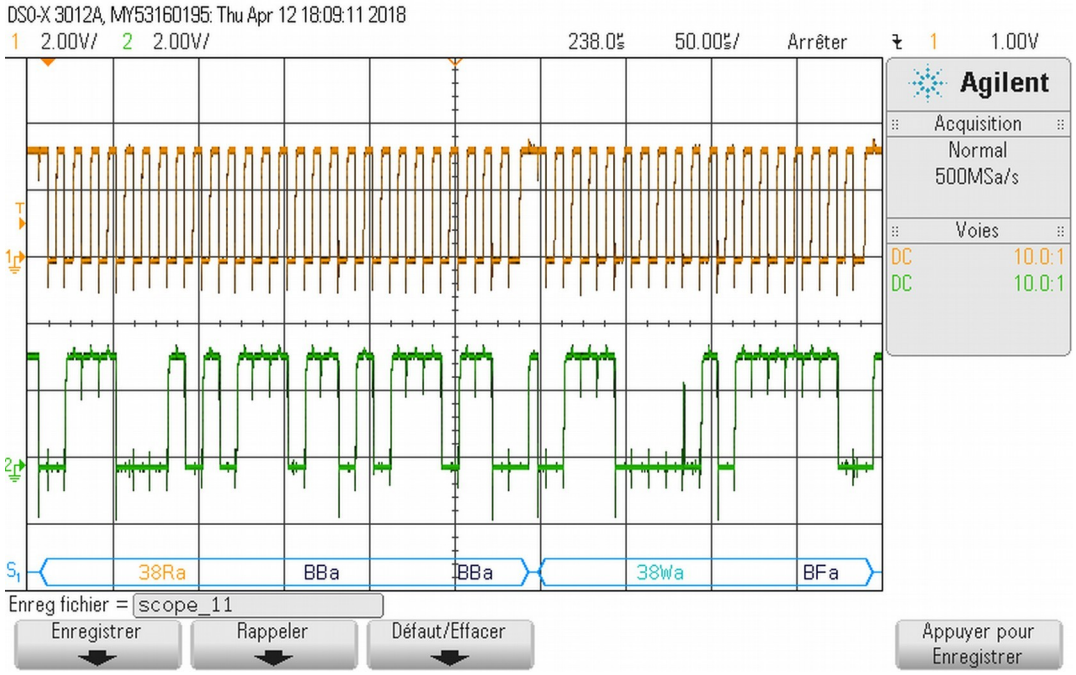
Les bits de poids faible du PCF8574A seront configurés en entrée et connectés manuellement à VSS ou VDD.
Réaliser un programme qui recopie les bits D3-D0 sur les bits D7-D4 du PCF8574A. Vous utiliserez le programme précédent comme base de travail en ajoutant une fonction

```
unsigned char lit_I2C(unsigned char add) ;
```

qui retourne la valeur en entrée du PCF8574A à l'adresse add.

Relever à l'aide de l'oscilloscope un échange complet, lecture/écriture entre le STM32 et le PCF8574A

Résultat attendu :



4.5 Exercice

A partir des exercices précédents, réaliser une classe PCF8574A et proposer un programme de test de cette classe.