TP n°7 - Le microcontrôleur STM32F411 - le bus I2C

Le programme principal permet de définir une interface mminimale sur la console série (115200 bauds, 8 bits de données, pas de parité, 1 bit de stop) qui reconnaît les commandes suivantes :

– `t` : interrogation du registre de température du capteur LM75,

– `m` : interrogation de l'accéléromètre MMA7660

Les deux périphériques sont connectés sur le bus interfacé via le coupleur `_I2C1`. Les fonctions permettant d'échanger des octets avec un capteur ont été étudiées en TD. Le code source est disponible dans les fichiers `lib/i2c.h` / `lib/i2c.c`.

1. Communication avec le capteur de température

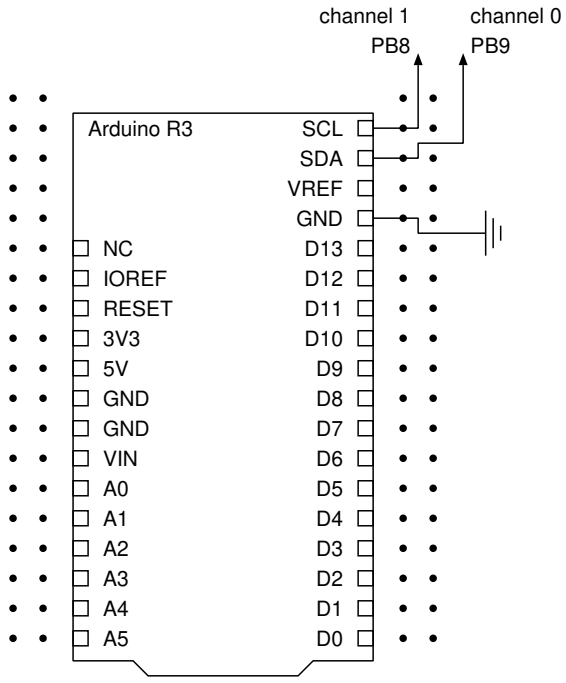
a. Lire les paragraphes dont les titres sont surlignés en jaune dans la documentation `docs/LM75B.pdf` et étudier les figures 11 et 12 p 15 qui définissent le protocole de communication entre le capteur et le processeur.

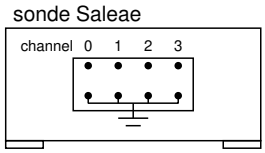
b. Dans le fichier `libshield/lm75.c`, compléter la fonction

int *lm75_read_temp*(**int** ⋆*temp*);

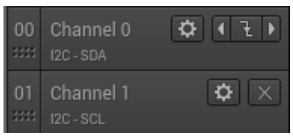
qui permet de lire le registre de température et renvoie une valeur signée codée en virgule fixe au format [29:3].

c. Câbler la sonde de l'analyseur logique Saleae conformément à la figure ci-dessous.





configuration des voies



configuration de l'analyseur de protocole

i. Lancer le logiciel Saleae et relever la trame de communication I2C (utiliser également la fonction de décodage de trame).

ii. Vérifier que la trame est conforme à la trame dans la documentation, ainsi qu'aux valeurs obtenue dans le buffer.

d. Compléter la fonction `main` (dans le fichier `src/main_i2c_irq.c`) pour afficher correctement la température dans le terminal série.

2. Communication avec l'accéléromètre

a. Lire les paragraphes dont les titres sont surlignés en jaune dans la documentation `docs/MMA7660FC.pdf` et étudier les figures 12, 13 et 15 p 23 et 24 qui définissent le protocole de communication entre le capteur et le processeur.

b. Compléter la fonction

int *mma7660_read_XYZT*(*int32_t *data*)

du fichier `libshield/mma7660.c` qui permet de lire les registres `X`, `Y`, `Z` et `Tilt` de l'accéléromètre. Les valeurs seront renvoyées par le tableau data passé en paramètre. On s'assurera que les valeurs renvoyées pour X, Y et Z soient des valeurs signées codées sur 32 bits.

Tester la communication avec l'accéléromètre et tester la cohérence entre les valeurs présentes dans la trame de communication et celles affichées dans le terminal série.

c. Fermer le terminal série. Modifier le symbole MMA0 défini en haut du fichier `src/main_i2c_irq.c` en MMA1. Cette modifcation permet d'envoyer sur la liaison série quatre octets, chaque octet représentant une des valeurs X, Y, Z et Tilt.

i. Justifier que la transmission d'un seul octet signé pour les composantes X, Y et Z est suffisant pour afficher correctement les informations.

ii. Lancer dans le terminal du PC le script *tcl*

```
tclsh dialog.tcl
```

qui attend la réception de quatre octets et les affiche dans une fenêtre. Vérifier que l'appui sur le bouton update (qui envoie la commande 'm') permet de rafraîchir l'affichage.

iii. Le script

```
tclsh dialog2.tcl
```

permet de relancer à intervalle de temps régulier la demande d'information auprès de l'accéléromètre.

Le capteur est un accéléromètre : il mesure l'accélération de pesanteur "g" suivant les trois axes x, y et z. Modifier les expressions dans les lignes 10 à 12 du script `dialog2.tcl` pour afficher la proportion de "g" mesurée suivant les 3 axes (voir la datasheet pour la table de correspondance).

iv. Exemple d'utilisation : lancer le script

```
tclsh 3display.tcl
```

et tester le fonctionnement.

v. Modifier le symbole MMA1 en MMA2. Recompiler et relancer le projet sur le microcontrôleur, puis relancer le script précédent. Expliquer ce qui a changé et comment ce changement est implémenté dans le code principal.