

TD 7 STM32

Dans cette exercice, nous allons reprendre l'exemple du cours sur le capteur de pression LPS22H et l'appliquer su LSM6DSL.

Un bus I2C est composé de 2 fils :

- ✓ Un bus d'horloge (fil CSL)
- ✓ Un bus de données (fil SDA)

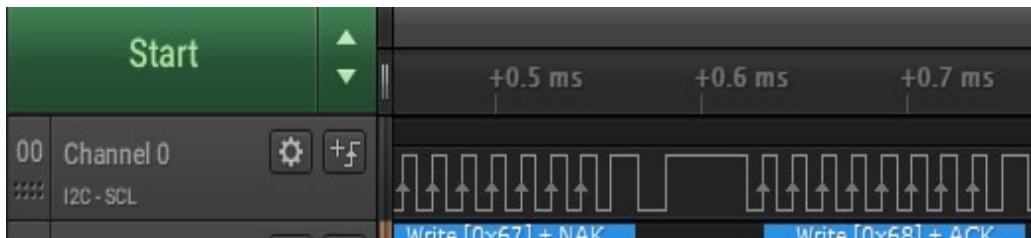
Le bus I2C est un bus Half-duplex, c'est-à-dire que l'information est bi-directionnelle, mais pas en même temps. En effet, comme il n'existe qu'un seul fil de données, l'information ne peut circuler que dans un sens. Soit le maître envoie de l'information à l'esclave, soit l'esclave envoie de l'information au maître.

Un maître peut contrôler plusieurs esclaves. Le bus I2C est très populaire pour cette raison. Avec seulement deux broches, un microcontrôleur peut commander de nombreux capteurs, ce qui rend le système très peu coûteux.

Toutes ces contraintes donnent lieu à des solutions qu'il faut comprendre afin de configurer et programmer le périphérique I2C.

Adresse du composant :

Afin de pouvoir connecter plusieurs composants ensemble et de reconnaître les composants, chaque composant a une adresse différente. Le maître peut scanner avec toutes les adresses possibles. L'esclave répond par un acknowledge quand il reconnaît son adresse.



Dans l'exemple ci-dessus, à l'adresse 0x68, un composant a fait un acknowledge, alors qu'aux adresses 0x67 et 0x69, il y a un No-Acknowledge, donc pas de composant avec ces adresses.

Si 2 composants ont la même adresse, le système ne pourra pas fonctionner car il y aura conflit entre les 2 composants. C'est la raison pour laquelle la plupart des composants offrent le choix entre plusieurs adresses.

Un capteur de pression LPS22H a les spécifications suivantes :

The slave address (SAD) associated to the LPS22HB is 1011
be used to modify the less significant bit of the device address:
to voltage supply, LSb is '1' (address 1011101b), otherwise if t
around the 1.0V value is 10' (address 1011100b). This solution

Table 11. SAD+Read/Write patter

Command	SAD[6:1]	SAD[0] = SA0	
Read	101110	0	
Write	101110	0	

La trame de l'I2C est formée de 9 bits. 7 bits pour l'adresse plus un bit pour informer si le maître veut écrire ou lire la donnée à / de l'esclave. Le 9^{ème} bit sert à l'acknowledge ou No-acknowledge. Par conséquent, on parle d'une adresse 7-bit, mais la documentation peut prêter parfois à confusion quand elle parle d'une adresse sur un octet (8 bits), incluant le write dans l'adresse.

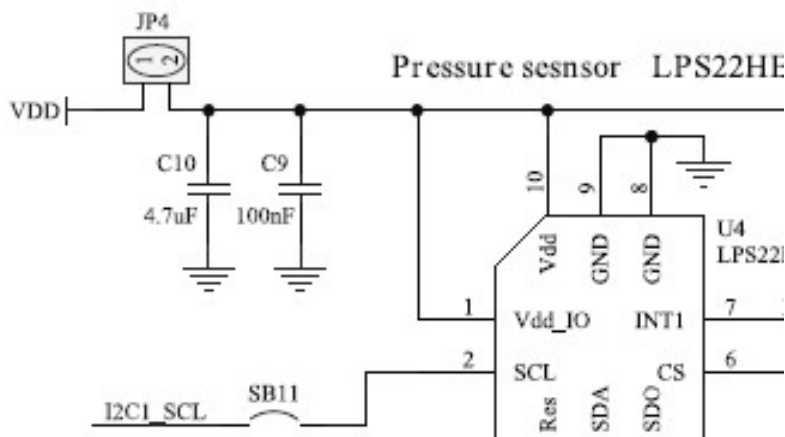
Avec les indications ci-dessus, l'adresse 0xBA se lit comme suit :

0xBA = bx1011 1010

Int	93
Hex	0x5D= bx0101 1101

Dans un logic analyzer, le composant sera repéré avec l'adresse 0x5D. C'est alors une adresse sur 7 bits. De nouveau, il est très important de lire et comprendre les spécifications.

Le choix d'une adresse se fait souvent avec une connection matérielle. Une broche à 3V ou 0V change un bit et donc une adresse. Dans le cas du LPS22H, quand SDO est à 3V, l'adresse vaut 0xBA et quand SDO est à la masse, l'adresse vaut 0xB8



Question :

Donner les adresses possibles pour le LSM6DSL. On donnera les adresses sur 7 et 8 bits

L'intérieur d'un composant est constitué de registres et donc de mémoires. Comme un fichier excel, les registres sont repérés par un registre d'adresse (à ne pas confondre avec l'adresse du composant !!!)

Pour savoir si la communication avec un composant est établie, on lit souvent en premier un registre appelé WHO_AM_I.

Un exemple de la datasheet du LPS22H montre que l'adresse du registre WHO_AM_I est à l'adresse 0x0Fh.

Table 16. Registers address map

Name	Type	Register Address	Default	Function and comment
		Hex	Binary	
Reserved		00 - 0A	-	Reserved
INTERRUPT_CFG	R/W	0B	00000000	Interrupt register
THS_P_L	R/W	0C	00000000	Pressure threshold registers
THS_P_H	R/W	0D	00000000	
Reserved		0E	-	Reserved
WHO_AM_I	R	0F	10110001	Who am I

Question :

Dans le LSM6DSL, quelles sont les adresses de registre des registres :

- ✓ WHO_AM_I
- ✓ CTR1_XL
- ✓ OUTZ_H_XH

Le protocole exact est expliqué dans la DataSheet du composant. Le protocole est différent si on écrit plusieurs ou un octet, si on lit un ou plusieurs octets.

Si un seul octet est lu, le protocole est le suivant :

Table 14. Transfer when master is receiving (reading) one byte of data from slave

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

La logique est la suivante :

- On prévient l'esclave qu'on va écrire (envoyer de l'information) (SAD+W)
- On dit à l'esclave que l'adresse du registre concernée (SUB)
- On informe l'esclave qu'on veut lire une donnée (SAD+R)
- L'esclave envoie alors la valeur stockée à l'adresse du registre SUB.



La cellule I2C du microcontrôleur prend soin de la partie matérielle, c'est-à-dire des acknowledge, no acknowledge...

Question :

La valeur de l'accélération se fait sur 2 octets.

- ✓ Repérer l'adresse de ces 2 registres
- ✓ En supposant que les valeurs des 2 registres soient dans un tableau uint8_t receivedData Avec le LSB dans receivedData[0] et le MSB dans receivedData[1], comment conserver cette valeur dans un entier 16-bit