FACULTE DES SCIENCES DE TUNIS
Département d'Informatique
Licence Computer Engineering
Parcours : Systèmes Embarqués & IoT
LCE2

Année Universitaire 2022-2023

T.D N°3: **Système sur puce (SoC) et Technologies d'interfaçage**

LIS302DL 3-axis accelerometer

1. In **STM32CubeMX**, you must enable **SPI1** to Full-Duplex Master mode and disable Hardware NSS Signal. You will see pin PA5, PA6 and PA7 will be enabled as SPI clock, SPI output and SPI input respectively. Then, you must enable **USART2** to asynchronous mode. You will see pin PA2 and PA3 will be enabled as USART2_Tx and USART2_Rx. Finally, you can generate file.
2. In main.c, before while loop coding, you need to write **CS** (Pin PE3) to low in order to transmit SPI. (set **CTRL_REG1**(20h) register by this value 0x67)

DR	0	100 Hz output data rate
PD	1	active mode
FS	1	enable
STP	0	normal mode
STM	0	normal mode
Zen	1	enable
Yen	1	enable
Xen	1	enable

Then, you need to write **CS** (Pin PE3) to high again.

3. In while loop, you need to read value of **Out_X**, **Out_Y** and **Out_Z** register from accelerometer. So you need to set **CS** to low in order to read from these **SPI** read of X axis value, Y axis value and Z axis value.
   X axis value = 0x29 OUT_X address + 0x80 to set READ bit high
   Y axis value = 0x2B OUT_X address + 0x80 to set READ bit high
   Z axis value = 0x2D OUT_X address + 0x80 to set READ bit high
   When you read value finish, you will set **CS** to high again.
4. After you have value of X, Y and Z, you need to convert these values into -+ 2g format. The formula is "(value_of_each_axis * 4000 *9.8) / 127" and you will keep answer of this formula to integer type. After that, you will use the function **itoa** to keep those integer into array of character and send these array of character to show on your display by **UART** transmit. Then while loop will repeat step 3 to 4 again.

```
#include <stdlib.h>
char * itoa(int n, char * buffer, int radix);
```

example : *int i;*
*char buffer [sizeof(int)*8+1];*
itoa (i,buffer,HEX);

EXTRAIT DATASHEET

7.2 CTRL_REG1 (20h)

Table 18. CTRL_REG1 (20h) register

DR	PD	FS	STP	STM	Zen	Yen	Xen

Table 19. CTRL_REG1 (20h) register description

DR	Data rate selection. Default value: 0 (0: 100 Hz output data rate; 1: 400 Hz output data rate)
PD	Power Down Control. Default value: 0 (0: power down mode; 1: active mode)
FS	Full Scale selection. Default value: 0 (refer to *Table 3* for typical full scale value)
STP, STM	Self Test Enable. Default value: 0 (0: normal mode; 1: self test P, M enabled)
Zen	Z axis enable. Default value: 1 (0: Z axis disabled; 1: Z axis enabled)
Yen	Y axis enable. Default value: 1 (0: Y axis disabled; 1: Y axis enabled)
Xen	X axis enable. Default value: 1 (0: X axis disabled; 1: X axis enabled)

7.7 OUT_X (29h)

Table 28. OUT_X (29h) register

XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0

X axis output data.

7.8 OUT_Y (2Bh)

Table 29. OUT_Y (2Bh) register description

YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0

Y axis output data.