

```

1  /*****
2  *
3  * HEIG-VD
4  * Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud
5  * School of Business and Engineering in Canton de Vaud
6
7  *****/
8
9  *
10 * File           : defines.h
11 * Author        : Sébastien Masle
12 * Date          : 16.02.2018
13 *
14 * Context       : SOCF class
15 *
16 *****/
17 * Brief: some definitions
18 *
19 *****/
20 * Modifications :
21 * Ver    Date      Engineer    Comments
22 * 0.0    16.02.2018 SMS         Initial version.
23 * 1.1    06.05.20  Isaia Spinelli : Refactor
24 *****/
25 /
26 #include "exceptions.h"
27
28 // Déclaration de fonction
29 void pushbutton_ISR(void);
30
31 // Defines
32
33 #define EDGE_TRIGGERED      0x1
34 #define LEVEL_SENSITIVE    0x0
35 #define CPU0                0x01 // bit-mask; bit 0 represents cpu0
36 #define ENABLE              0x1
37
38 #define USER_MODE          0b10000
39 #define FIQ_MODE            0b10001
40 #define IRQ_MODE            0b10010
41 #define SVC_MODE            0b10011
42 #define ABORT_MODE          0b10111
43 #define UNDEF_MODE          0b11011
44 #define SYS_MODE            0b11111
45
46 #define INT_ENABLE          0b01000000
47 #define INT_DISABLE        0b11000000
48
49 // Valeur des keys
50 #define KEY0 0x01
51 #define KEY1 0x02
52 #define KEY2 0x04
53 #define KEY3 0x08
54
55 // Typedef
56 typedef volatile unsigned char vcint;
57 typedef volatile unsigned short vsint;
58 typedef volatile unsigned int vuint;
59

```

```
60 // Adresses
61 #define FPGA_BASE_ADDR_IO 0xFF200000
62 #define AXI_LIGHT_BASE_ADDR FPGA_BASE_ADDR_IO
63
64
65 #define AXI_REG_CONST_CHAR *(vcint *) (AXI_LIGHT_BASE_ADDR + 0x0)
66 #define AXI_REG_CONST_SHORT *(vsint *) (AXI_LIGHT_BASE_ADDR + 0x0)
67 #define AXI_REG_CONST *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x0)
68
69 #define AXI_REG_TEST *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x4)
70
71 #define AXI_LEDS *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x100)
72
73 #define AXI_KEYS *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x200)
74 // Lecture de la source d'int. + acquitement
75 #define AXI_INT_SRC *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x204)
76 // 1 = interruption masquée
77 #define AXI_INT_MASK *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x208)
78
79 #define AXI_SWITCHES *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x300)
80
81 #define AXI_HEX0 *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x400)
82 #define AXI_HEX1 *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x410)
83 #define AXI_HEX2 *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x420)
84 #define AXI_HEX3 *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x430)
85 #define AXI_HEX4 *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x440)
86 #define AXI_HEX5 *(vuint *) (AXI_LIGHT_BASE_ADDR + 0x450)
87
```