```
/*****************************************************************************
*
 * HEIG-VD
 * Haute Ecole d'Ingenerie et de Gestion du Canton de Vaud
 * School of Business and Engineering in Canton de Vaud

 *****************************************************************************
 *
 * REDS Institute
 * Reconfigurable Embedded Digital Systems

 *****************************************************************************
 *
 *
 * File                 : defines.h
 * Author               : Sébastien Masle
 * Date                 : 16.02.2018
 *
 * Context              : SOCF class
 *

 *****************************************************************************
 *
 * Brief: some definitions
 *

 *****************************************************************************
 *
 * Modifications :
 * Ver    Date        Engineer      Comments
 * 0.0    16.02.2018  SMS           Initial version.
 * 1.1    06.05.20    Isaia Spinelli : Refactor
 * 1.2    08.05.20    Isaia Spinelli : Ajout du paramètre edge des irq
*****************************************************************************
/

#include "exceptions.h"

// Déclaration de fonction
void pushbutton_ISR(void);

// Defines

#define     EDGE_TRIGGERED          0x1
#define     LEVEL_SENSITIVE         0x0
#define     CPU0                    0x01    // bit-mask; bit 0 represents cpu0
#define     ENABLE                  0x1

#define     USER_MODE               0b10000
#define     FIQ_MODE                0b10001
#define     IRQ_MODE                0b10010
#define     SVC_MODE                0b10011
#define     ABORT_MODE              0b10111
#define     UNDEF_MODE              0b11011
#define     SYS_MODE                0b11111

#define     INT_ENABLE              0b01000000
#define     INT_DISABLE             0b11000000

// Valeur des keys
#define KEY0 0x01
#define KEY1 0x02
#define KEY2 0x04
#define KEY3 0x08

// Typedef
typedef volatile unsigned char vcint;
typedef volatile unsigned short vsint;
typedef volatile unsigned int vuint;
```

```
60
61  // Adresses
62  #define FPGA_BASE_ADDR_IO       0xFF200000
63  #define AXI_LIGHT_BASE_ADDR     FPGA_BASE_ADDR_IO
64
65
66  #define AXI_REG_CONST_CHAR      *(vcint *)(AXI_LIGHT_BASE_ADDR + 0x0)
67  #define AXI_REG_CONST_SHORT     *(vsint *)(AXI_LIGHT_BASE_ADDR + 0x0)
68  #define AXI_REG_CONST           *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x0)
69
70  #define AXI_REG_TEST            *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x4)
71
72  #define AXI_LEDS                *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x100)
73
74  #define AXI_KEYS                *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x200)
75  // Lecture de la source d'int. + acquitement
76  #define AXI_INT_SRC             *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x204)
77  // 0 = interruption non masquée (défaut)
78  #define AXI_INT_MASK            *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x208)
79  // 0 = interruption flanc descendant (défaut)
80  #define AXI_INT_EDGE            *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x20C)
81
82
83  #define AXI_SWITCHES            *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x300)
84
85  #define AXI_HEX0                *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x400)
86  #define AXI_HEX1                *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x410)
87  #define AXI_HEX2                *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x420)
88  #define AXI_HEX3                *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x430)
89  #define AXI_HEX4                *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x440)
90  #define AXI_HEX5                *(vuint *)(AXI_LIGHT_BASE_ADDR + 0x450)
91
```