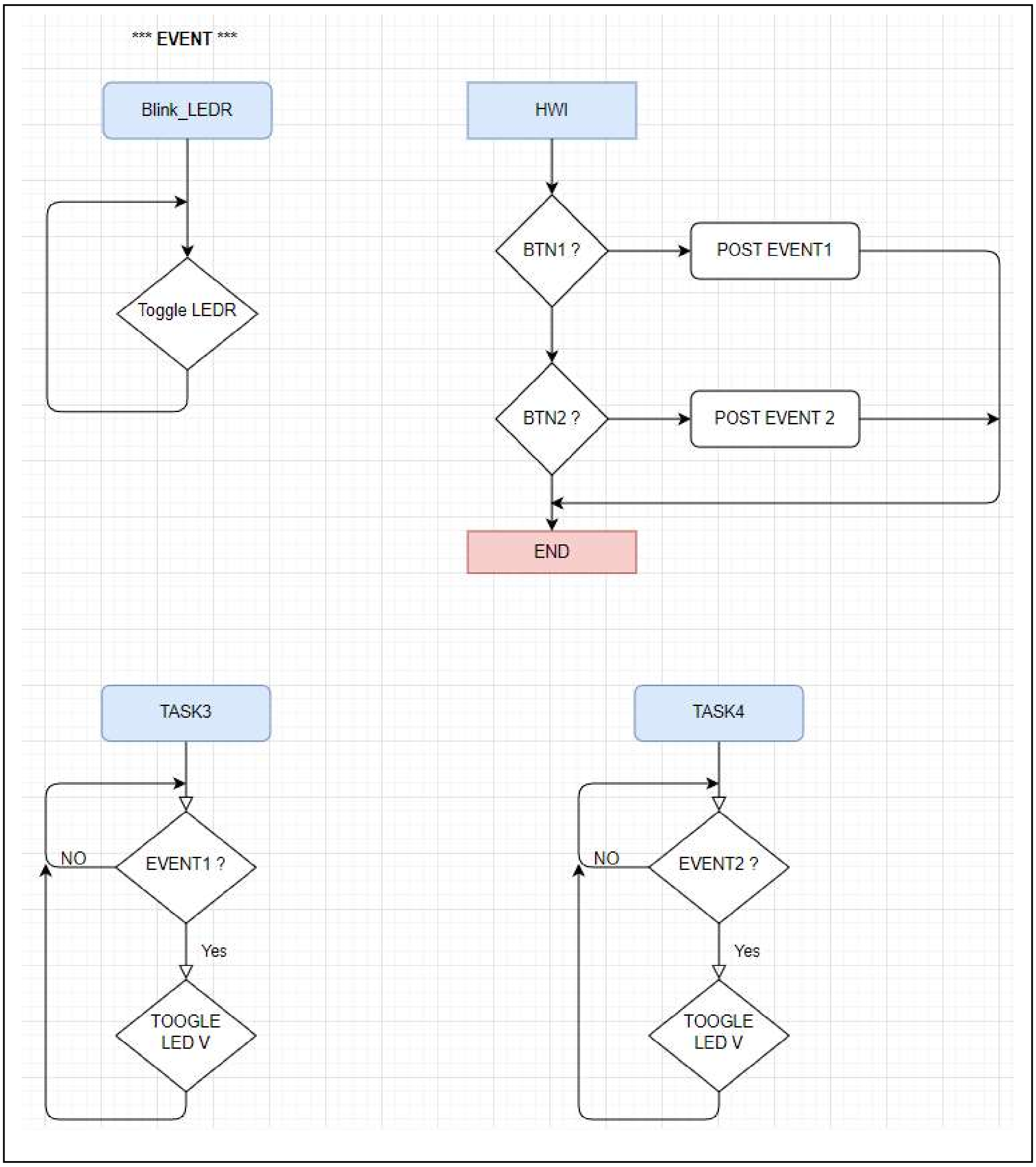
Exemples d’évènement (EVENT)

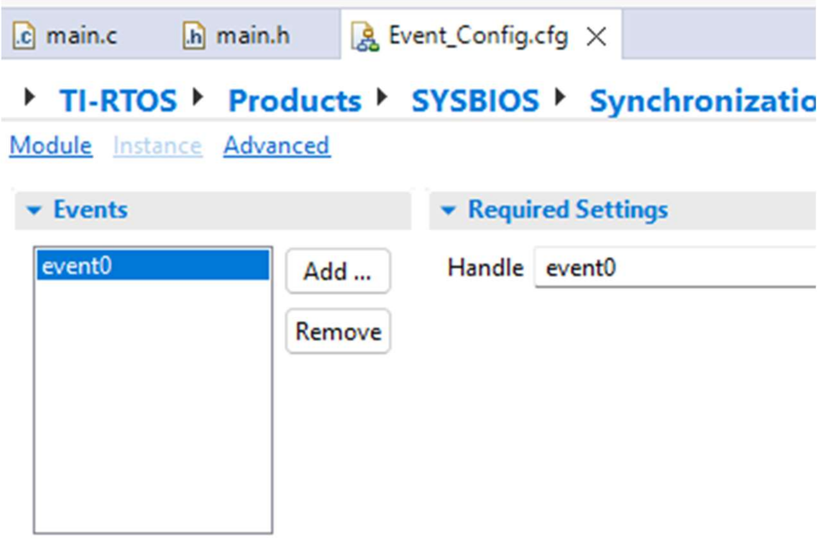


```c
/*******************************************************************
 * main.h
 *
 *  Created on: 18 oct. 2022
 *      Author: beepr
 *******************************************************************/

#ifndef MAIN_H_
#define MAIN_H_

//PORT1
#define LED_ROUGE   GPIO_PIN0
#define BOUTON1     GPIO_PIN1
#define BOUTON2     GPIO_PIN2
//PORT9
#define LED_VERTE   GPIO_PIN7

//EVENT CONSTANT
#define EVENT_TIMEOUT   12
#define BTN1_EVENT      Event_Id_01
#define BTN2_EVENT      Event_Id_02

#endif /* MAIN_H_ */
```

```c
/*******************************************************************
 * Copyright (c) 2015, Texas Instruments Incorporated
 * All rights reserved.
 *
 *  ======== EVENT Example ========
 *
*******************************************************************/
#include <stdint.h>
#include <stdbool.h>
#include <string.h>
/* XDCtools Header files */
#include <xdc/std.h>
#include <xdc/runtime/System.h>
#include <xdc/cfg/global.h>
/* BIOS Header files */
#include <ti/sysbios/BIOS.h>
#include <ti/sysbios/knl/Task.h>
#include <ti/sysbios/knl/Event.h>
/* TI-RTOS Header files */
#include <driverlib.h>
/* Board Header file */
#include "main.h"

/*******************************************************************
 * Prototype de fonction
*******************************************************************/

void Init_GPIO(void);
void Blink_LEDR(UArg arg0, UArg arg1);
void Task_Event_BTN1(UArg arg0, UArg arg1);
void Task_Event_BTN2(UArg arg0, UArg arg1);
void Irq_Port1(unsigned index);

/*******************************************************************
 *  ========  ========
*******************************************************************/
int main(void)
{
    WDT_A_hold(WDT_A_BASE); //Stop WDT
    PM5CTL0 &= ~LOCKLPM5;

    Init_GPIO();

    /* Start BIOS */
    BIOS_start();

    return (0);
}
```

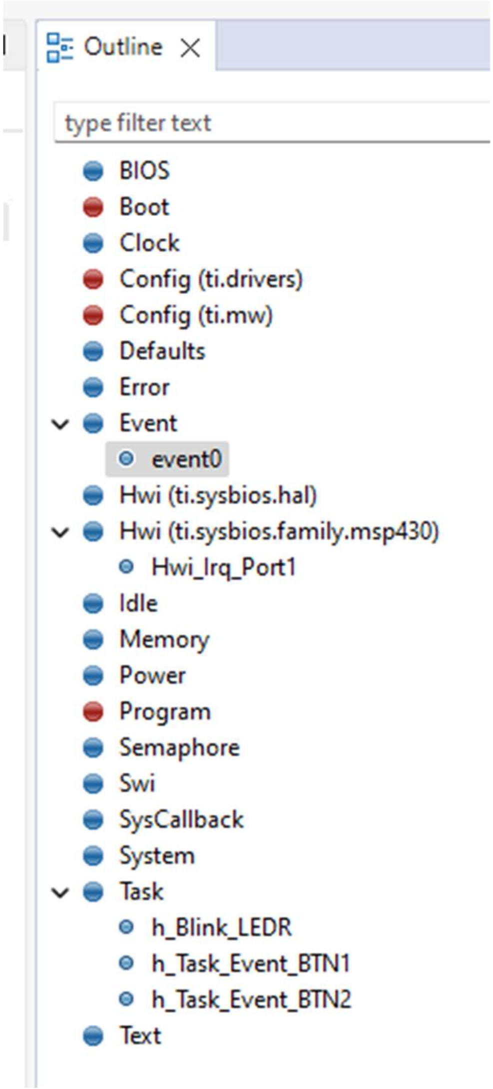
```
/*******************************************************************
 *   ========   ========
*******************************************************************/
void Init_GPIO(void)
{
    //PORT1
    GPIO_setAsOutputPin(GPIO_PORT_P1, LED_ROUGE);
    GPIO_setOutputLowOnPin(GPIO_PORT_P1, LED_ROUGE);
    GPIO_setAsInputPin(GPIO_PORT_P1,BOUTON1 + BOUTON2);
    //PORT9
    GPIO_setAsOutputPin(GPIO_PORT_P9, LED_VERTE);
    GPIO_setOutputLowOnPin(GPIO_PORT_P9, LED_VERTE);
    //IRQ
    GPIO_setAsInputPinWithPullUpResistor(GPIO_PORT_P1,BOUTON1 +
BOUTON2);
    GPIO_selectInterruptEdge(GPIO_PORT_P1, BOUTON1 + BOUTON2,
GPIO_HIGH_TO_LOW_TRANSITION);
    GPIO_enableInterrupt(GPIO_PORT_P1,BOUTON1 + BOUTON2);
    GPIO_clearInterrupt(GPIO_PORT_P1,BOUTON1 + BOUTON2);
}

/*******************************************************************
 *   ======== Blink_LEDR ========
*******************************************************************/
void Blink_LEDR(UArg arg0, UArg arg1)
{
    while (1)
        {
        Task_sleep(1000);
        GPIO_toggleOutputOnPin(GPIO_PORT_P1, LED_ROUGE);
        }
}


/*******************************************************************
 *   ======== Task_Event_BTN1 ========
*******************************************************************/
void Task_Event_BTN1(UArg arg0, UArg arg1)
{
    uint16_t Posted;

    while (1)
        {
        Posted = Event_pend(event0,              // hande du registre
event
                            Event_Id_NONE,       // andMask
                            BTN1_EVENT,          // orMask
                            EVENT_TIMEOUT);
        switch(Posted)
            {
            case BTN1_EVENT:
                GPIO_setOutputHighOnPin(GPIO_PORT_P9, LED_VERTE);
                break;
            }

        Task_sleep(1000);
        }
}

/*******************************************************************
 *  ======== Task_Event_BTN2 ========
*******************************************************************/
void Task_Event_BTN2(UArg arg0, UArg arg1)
{
    uint16_t Posted;

    while (1)
        {
        Posted = Event_pend(event0,         // hande du registre
event
                            Event_Id_NONE,  // andMask
                            BTN2_EVENT,     // orMask
                            EVENT_TIMEOUT);

        switch(Posted)
            {
            case BTN2_EVENT:
                GPIO_setOutputLowOnPin(GPIO_PORT_P9, LED_VERTE);
                break;
            }

        Task_sleep(500);
        }
}

/*******************************************************************
 *  ======== Irq_Port1 ========
 *
 *  Vector: ( .int37 )
*******************************************************************/
void Irq_Port1(unsigned index)
{
    uint16_t Status = GPIO_getInterruptStatus(GPIO_PORT_P1,
BOUTON1+BOUTON2);

    switch(Status)
        {
```

```
        case BOUTON1:
            Event_post(event0, BTN1_EVENT);
            GPIO_clearInterrupt(GPIO_PORT_P1, BOUTON1);
            break;

        case BOUTON2:
            Event_post(event0, BTN2_EVENT);
            GPIO_clearInterrupt(GPIO_PORT_P1, BOUTON2);
            break;
        }
}

/*******************************************************************
```



event0

Handle event0



Outline

type filter text

- BIOS
- Boot
- Clock
- Config (ti.drivers)
- Config (ti.mw)
- Defaults
- Error
- Event
  - event0
- Hwi (ti.sysbios.hal)
- Hwi (ti.sysbios.family.msp430)
  - Hwi_Irq_Port1
- Idle
- Memory
- Power
- Program
- Semaphore
- Swi
- SysCallback
- System
- Task
  - h_Blink_LEDR
  - h_Task_Event_BTN1
  - h_Task_Event_BTN2
- Text