## Interrupt.hpp

```
#ifndef LPC_INT_H__
#define LPC_INT_H__

#include <stdint.h>
#include "LPC17xx.h"
#include "GPIO.hpp"
#include "uart0_min.h"
#include "printf_lib.h"

enum InterruptCondition
{
    RisingEdge,
    FallingEdge,
    BothEdges,
};

typedef void (*IsrPointer)(void);

class LabGpioInterrupts
{
 public:

    void Initialize();
    bool AttachInterruptHandler(uint8_t port, uint32_t pin, IsrPointer
pin_isr, InterruptCondition condition);
    void HandleInterrupt();

 private:

    IsrPointer pin_isr_map[2][32];
};

#endif;
```

## Interrupt.cpp

```
#include "Interrupt.hpp"

void LabGpioInterrupts :: Initialize()
{
    NVIC_EnableIRQ(EINT3_IRQn);
}

bool LabGpioInterrupts :: AttachInterruptHandler(uint8_t port,
uint32_t pin, IsrPointer pin_isr, InterruptCondition condition)
{
    bool f;
    pin_isr_map[port][pin] = pin_isr;
    uart0_puts("ISR sup");
    f=1;
    if(port==0 && pin<=31)
    {
        f=1;
        LPC_GPIOINT->IO0IntClr |= (1 << pin);
        LPC_PINCON->PINSEL0 &= ~(1 << pin);
        LPC_GPIO0->FIODIR &= ~(1 << pin);
        if(condition==RisingEdge)
        {
            LPC_GPIOINT->IO0IntEnR |= (1 << pin);
            uart0_puts("0RE Complete");
        }
        else if(condition==FallingEdge)
        {
            LPC_GPIOINT->IO0IntEnF |= (1 << pin);
            uart0_puts("0FE Complete");
        }
        else if(condition==BothEdges)
        {
            LPC_GPIOINT->IO0IntEnF |= (1 << pin);
            LPC_GPIOINT->IO0IntEnR |= (1 << pin);
            uart0_puts("0BE Complete");
        }
        else{f=0;}
    }

    else if(port==2 && pin<=31)
    {
        f=1;
        LPC_GPIOINT->IO2IntClr |= (1 << pin);
        LPC_GPIO2->FIODIR &= ~(1 << pin);
        if(condition==RisingEdge)
        {
            LPC_GPIOINT->IO2IntEnR |= (1 << pin);
            uart0_puts("2RE Complete");
        }
        if(condition==FallingEdge)
        {
            LPC_GPIOINT->IO2IntEnF |= (1 << pin);
            uart0_puts("2FE Complete");
        }
        if(condition==BothEdges)
        {
            LPC_GPIOINT->IO2IntEnF |= (1 << pin);
            LPC_GPIOINT->IO2IntEnR |= (1 << pin);
            uart0_puts("2BE Complete");
        }
        else{f=0;}
        }return f;
    }

void LabGpioInterrupts :: HandleInterrupt()
{
    uint32_t intpin=1, pin_ext, pin_value=0;
    uint32_t intport;
    bool p=0;
    uart0_puts("HandleIntr");

    if(((pin_ext = LPC_GPIOINT->IO0IntStatR) || (pin_ext =
LPC_GPIOINT->IO0IntStatF)) >= 1)
    {
        intport = 0;
        while(p != 1)
        {
            p = (pin_ext & intpin);
            intpin = (intpin << 1);
            pin_value++;
        }
        pin_value -= 1;
        u0_dbg_printf("Pin Val - ");
        u0_dbg_printf("%d", pin_value);
        LPC_GPIOINT->IO0IntClr |= (1 << (pin_value));
    }

    else if(((pin_ext = LPC_GPIOINT->IO2IntStatR) || (pin_ext =
LPC_GPIOINT->IO2IntStatF)) >= 1)
```

```
    {
        intport = 2;
        while(p != 1)
        {
            p = (pin_ext & intpin);
            intpin = (intpin << 1);
            pin_value++;
        }
        pin_value -=1;
        u0_dbg_printf("Pin Val - ");
        u0_dbg_printf("%d", pin_value);
        LPC_GPIOINT->IO2IntClr |= (1 << (pin_value));
    }

    else if(((pin_ext = LPC_GPIOINT->IO0IntStatR) && (pin_ext = 
LPC_GPIOINT->IO0IntStatF)) >= 1)
    {
        intport = 0;
        while(p != 1)
        {
            p = (pin_ext & intpin);
            intpin = (intpin << 1);
            pin_value++;
        }
        pin_value -= 1;
        u0_dbg_printf("Pin Val - ");
        u0_dbg_printf("%d", pin_value);
        LPC_GPIOINT->IO0IntClr |= (1 << (pin_value));
    }
    else if(((pin_ext = LPC_GPIOINT->IO2IntStatR) && (pin_ext = 
LPC_GPIOINT->IO2IntStatF)) >= 1)
    {
        intport = 2;
        while(p != 1)
        {
            p = (pin_ext & intpin);
            intpin = (intpin << 1);
            pin_value++;
        }
        pin_value -=1;
        u0_dbg_printf("Pin Val - ");
        u0_dbg_printf("%d", pin_value);
        LPC_GPIOINT->IO2IntClr |= (1 << (pin_value));
    }
    pin_isr_map[intport][pin_value]();
}
```

**main.cpp**

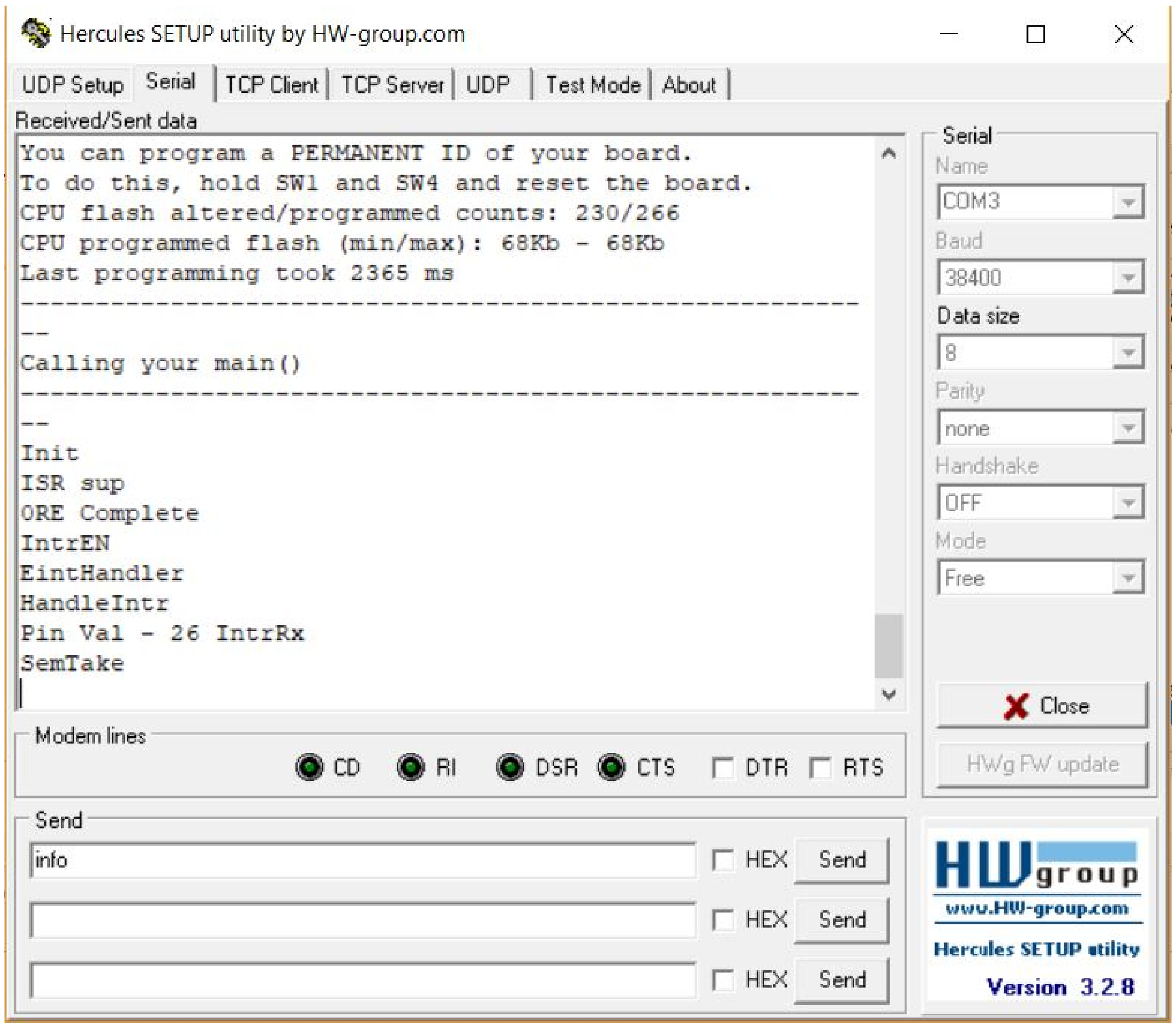
```
#include <FreeRTOS.h>
#include <stdint.h>
#include "GPIO.hpp"
#include "task.h"
#include "Interrupt.hpp"
#include "uart0_min.h"
#include "semphr.h"

SemaphoreHandle_t LED_Semaphore;

LabGpioInterrupts gpio_interrupt;
GPIO_Lab LED(1);

void user_callback(void)
{
    xSemaphoreGiveFromISR(LED_Semaphore, 0);
    uart0_puts(" IntrRx");
}

void Eint3Handler(void)
{
    uart0_puts("EintHandler");
    gpio_interrupt.HandleInterrupt();
}

void vControlLED (void * pvParameters)
{
    uint32_t param = (uint32_t)(pvParameters);
    LED.setAsOutput();
    while(1)
    {
        if(xSemaphoreTake(LED_Semaphore, portMAX_DELAY))
               {
                uart0_puts("SemTake");
                if(LED.getLevel()==1)
                    LED.setLow();
                else
                    LED.setHigh();
               }
    }
}
```

```
int main(void)
{

    isr_register(EINT3_IRQn, Eint3Handler);
    gpio_interrupt.Initialize();
    uart0_puts("Init");
    LED.setAsOutput();
    gpio_interrupt.AttachInterruptHandler(2, 7, user_callback, 
BothEdges);
    uart0_puts("IntrEN");
    LED_Semaphore = xSemaphoreCreateBinary();

    xTaskCreate(vControlLED, (const char*) "LED", 512, NULL, 1, NULL);

    vTaskStartScheduler();
    return 0;
}
```

*Output terminal screenshots on the next page

## Output Terminal Screenshots

`Port 0.26, Rising Edge`



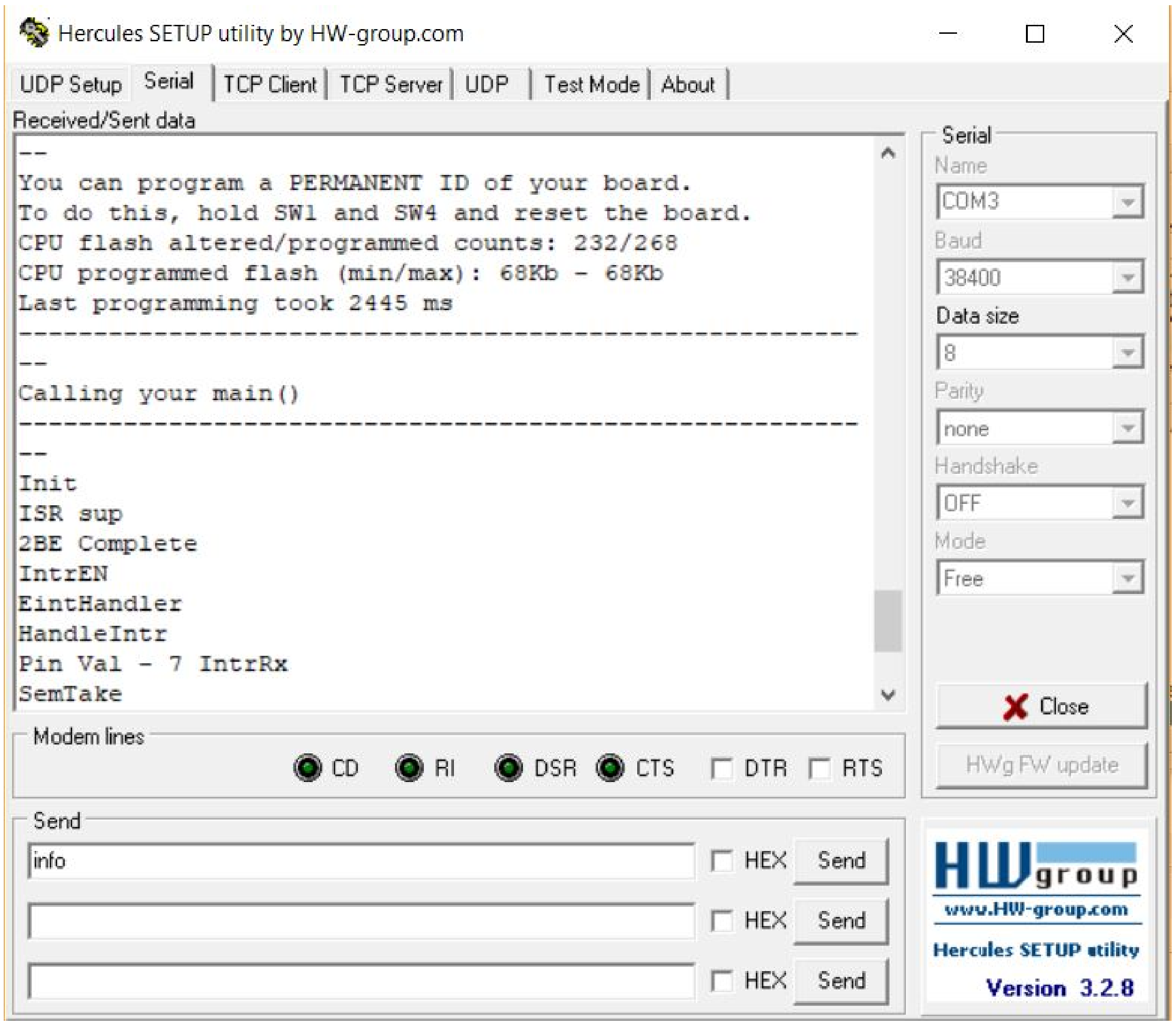
Hercules SETUP utility by HW-group.com

```
You can program a PERMANENT ID of your board.
To do this, hold SW1 and SW4 and reset the board.
CPU flash altered/programmed counts: 230/266
CPU programmed flash (min/max): 68Kb - 68Kb
Last programming took 2365 ms
---------------------------------------------------------
--
Calling your main()
---------------------------------------------------------
--
Init
ISR sup
ORE Complete
IntrEN
EintHandler
HandleIntr
Pin Val - 26 IntrRx
SemTake
```

Serial
Name: COM3
Baud: 38400
Data size: 8
Parity: none
Handshake: OFF
Mode: Free

Send: info

Port 2.7, Falling Edge

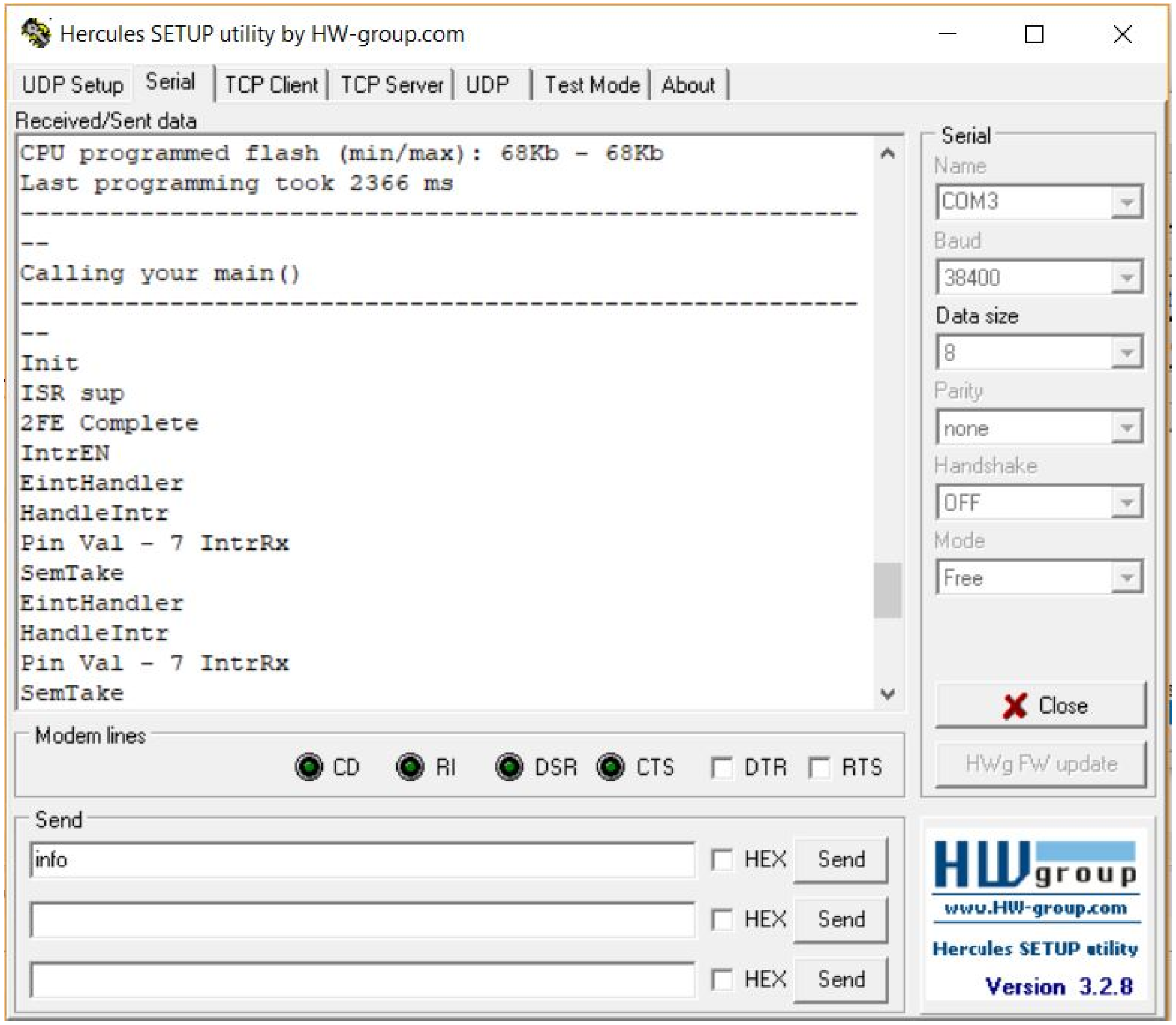


```
--
You can program a PERMANENT ID of your board.
To do this, hold SW1 and SW4 and reset the board.
CPU flash altered/programmed counts: 232/268
CPU programmed flash (min/max): 68Kb - 68Kb
Last programming took 2445 ms
-------------------------------------------------------
--
Calling your main()
-------------------------------------------------------
--
Init
ISR sup
2BE Complete
IntrEN
EintHandler
HandleIntr
Pin Val - 7 IntrRx
SemTake
```

Serial
Name: COM3
Baud: 38400
Data size: 8
Parity: none
Handshake: OFF
Mode: Free

Send: info

Port 2.7, Both Edges



```
CPU programmed flash (min/max): 68Kb - 68Kb
Last programming took 2366 ms
---------------------------------------------------------
--
Calling your main()
---------------------------------------------------------
--
Init
ISR sup
2FE Complete
IntrEN
EintHandler
HandleIntr
Pin Val - 7 IntrRx
SemTake
EintHandler
HandleIntr
Pin Val - 7 IntrRx
SemTake
```

Serial
Name: COM3
Baud: 38400
Data size: 8
Parity: none
Handshake: OFF
Mode: Free

Send: info