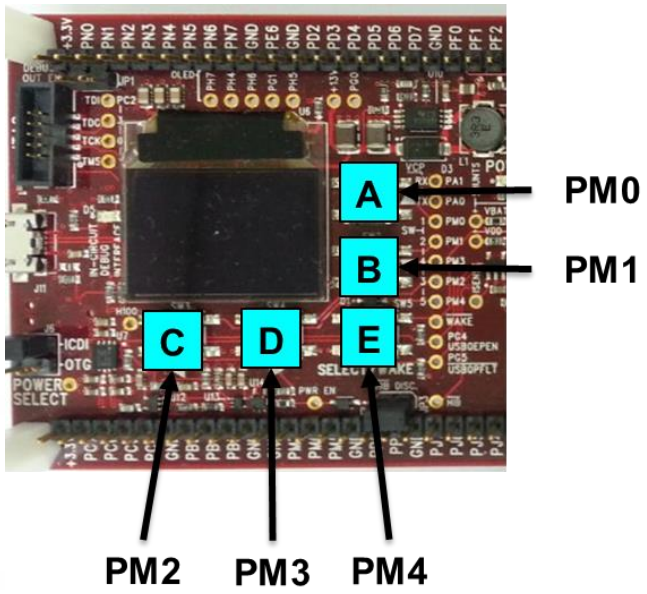
ARM HW4

2019. 04. 05.

Embedded System LAB
SKKU

Implementation Topic

- **Interrupt**
  - Using switch
  - C : LED ON
  - D : LED OFF
  - E : Push switch → LED ON
    Release switch → LED OFF



Implementation Conditions

- Use Hardware interrupt

- You must coding
  - Vector table
  - Switch Initialize
  - LED Initialize
  - Interrupt Configuration
  - Interrupt enable
  - Unmasking
  - Interrupt default handler
  - Handler code
  - Interrupt Clear

Program flowchart 1

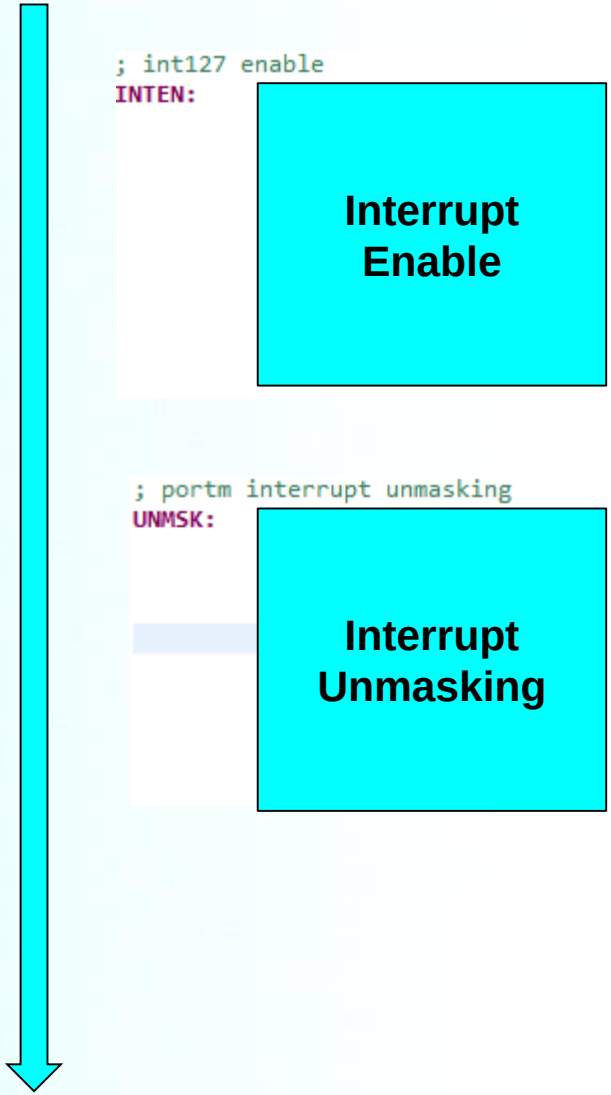
```
          .global __stack
__stack:

;-------------------------------------------------------------------------------
                        ; Interrupt Vectors
;-------------------------------------------------------------------------------
                        .sect   ".intvecs"
                        .align  4
                        .field  IntDefaultHandler,32        ; g_pfnVectors[0] @ 0
                        .field  IntDefaultHandler,32        ; g_pfnVectors[1] @ 32
                        .field  IntDefaultHandler,32        ; g_pfnVectors[2] @ 64
                        .field  IntDefaultHandler,32        ; g_pfnVectors[3] @ 96
                        .field  IntDefaultHandler,32        ; g_pfnVectors[4] @ 128
                        .field  IntDefaultHandler,32        ; g_pfnVectors[5] @ 160
                                        ⋮
                        .field  IntGPIOm,32     ; g_pfnVectors[127] @ 4064
                                        ⋮
```

```
196 ;-----------------------------------------
197             .text                       ;
198 ;-----------------------------------------
199
200 SWITCH:
201             mov r0, #GPIO_BASE  ;RCGC : Gener
202             mov r1, #0xFE000
203             add r1, r1, r0
204             mov r0, #RCGCGPIO
205             add r1, r1, r0
206
207             ldr r0, [r1]
208             orr r0, r0, #0x800
209             str r0, [r1]
210             nop
211             nop
```

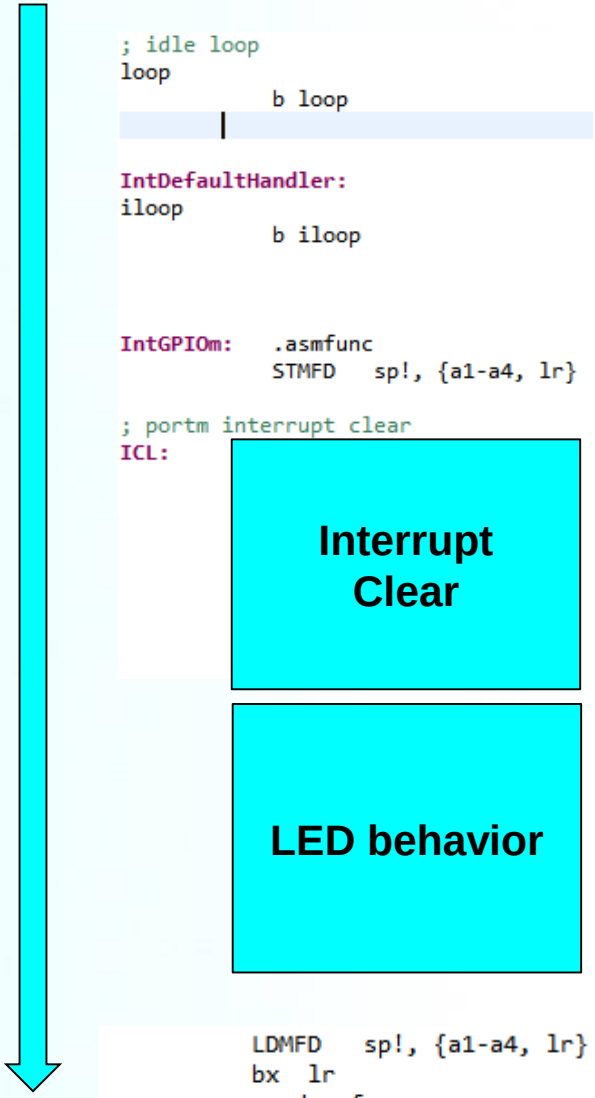
⋮

Interrupt Configuration

Program flowchart 2



Program flowchart 3



GPIO PORT M

Vector Number (Interrupt Number)

Table 2-2. User Switches and User LED Signals

GPIO Pin	Pin Function	Feature
PM0	GPIO	SW1 (Up)
PM1	GPIO	SW2 (Down)
PM2	GPIO	SW3 (Left)
PM3	GPIO	SW4 (Right)
PM4	GPIO	SW5 (Select/Wake)
PG2	GPIO	User LED

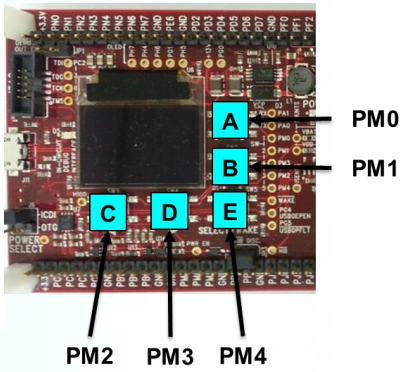


Figure 2-6. Vector Table

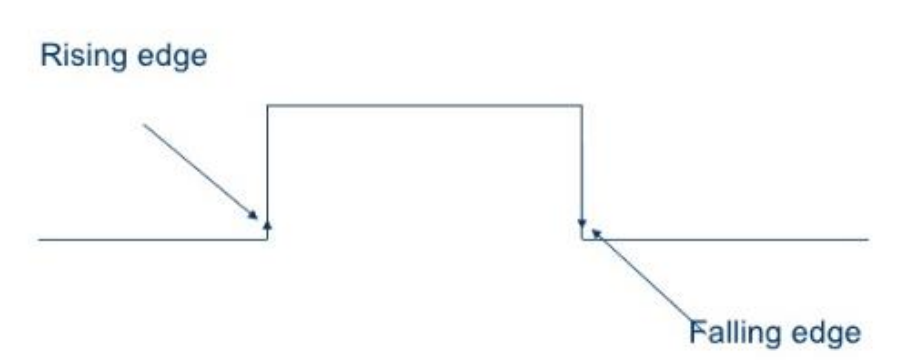
Exception number	IRQ number	Offset	Vector
154	138	0x0268	IRQ 138
.			.
.			.
.			.
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	Systick
14	-2	0x003C	PendSV
13		0x0038	Reserved
12			Reserved for Debug
11	-5		SVCall
10		0x002C	Reserved
9			
8			
7			
6	-10		Usage fault
5	-11	0x0018	Bus fault
4	-12	0x0014	Memory management fault
3	-13	0x0010	Hard fault
2	-14	0x000C	NMI
1		0x0008	Reset
		0x0004	Initial SP value
		0x0000	

Datasheet – TM4C123GH6PGE.pdf

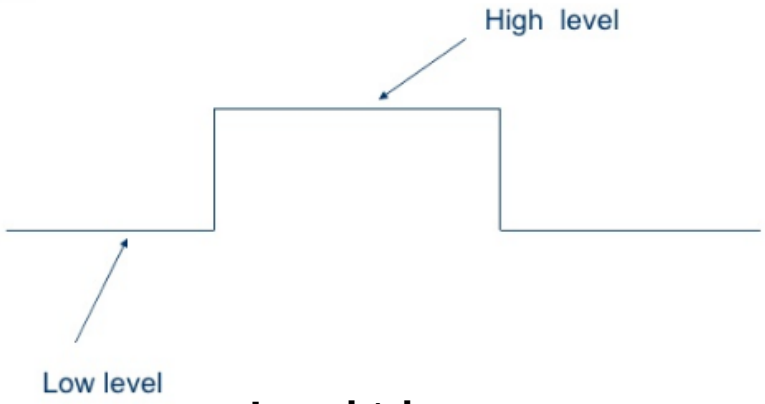
Vector Number	Interrupt Number (Bit in Interrupt Registers)	Vector Address or Offset	Description
75	59	0x0000.012C	UART3
76	60	0x0000.0130	UART4
77	61	0x0000.0134	UART5
⋮			
127	111	0x0000.01FC	GPIO Port M
128	112	0x0000.0200	GPIO Port N

Interrupt Configuration

- **Edge trigger, Level trigger**



Edge trigger



Level trigger

- **Related Register**
  - **GPIOIS (GPIO Interrupt Sense)**
  - **GPIOIEV (GPIO Interrupt Event)**

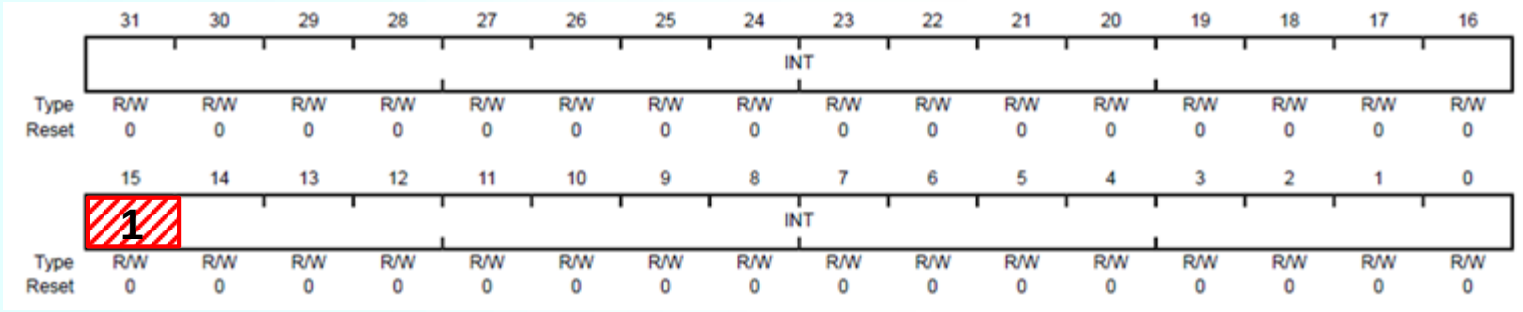
Register setting

- **EN3**
  - Interrupt 96 ~ 127 Set Enable
- **GPIOIM**
  - GPIO Interrupt Mask
- **GPIOICR**
  - GPIO Interrupt Clear

Register setting

- **EN3**
  - **Interrupt 96-127 Set Enable**
  - **This register can only be accessed from privileged mode**

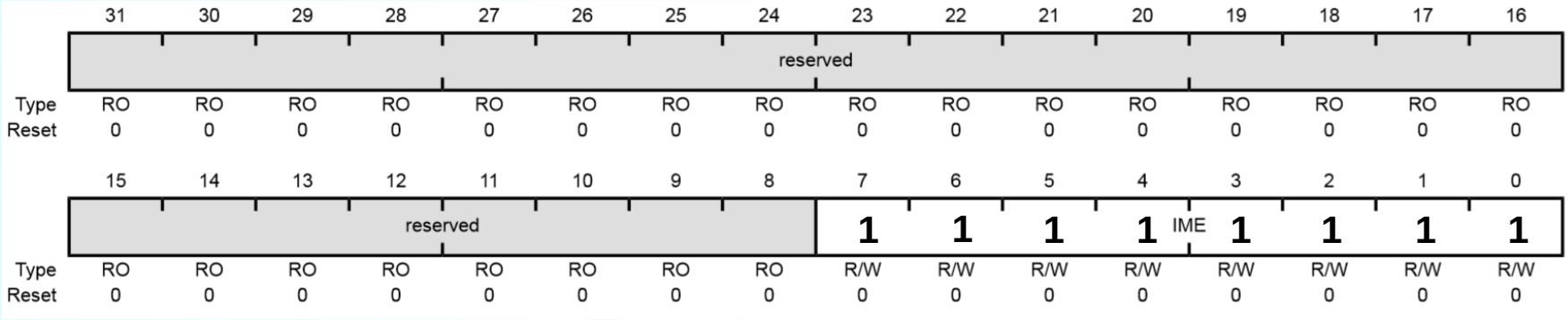
EN3[15] = Set '1' ← IRQ111 is enable



- **Access address: 0xE000.E10C**
- **Base: 0xE000.E000, Offset: 0x10C**
- **Value: 0x8000**
  - **On a read, indicates the interrupt is enabled.**
  - **On a write, enables the interrupt.**

Register setting

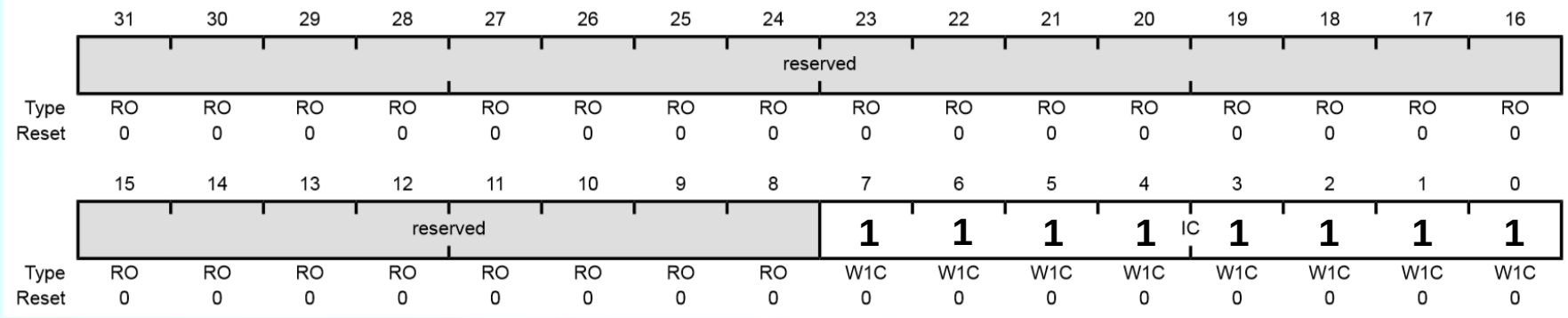
- **GPIOIM**
  - **GPIO Interrupt Mask**



  - **Access address: 0x4006.3410**
  - **Base: 0x4006.3000, Offset: 0x410**
  - **Value: 0xFF**
    - **0** The interrupt from the corresponding pin is masked.
    - **1** The interrupt from the corresponding pin is sent to the interrupt controller.

Register setting

- **GPIOICR**
  - **GPIO Interrupt Clear**



	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								1	1	1	1	IC	1	1	1	1
Type	RO	RO	RO	RO	RO	RO	RO	RO	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

  - **Access address: 0x4006.341C**
  - **Base: 0x4006.3000, Offset: 0x41C**
  - **Value: 0xFF**
    - **0 The corresponding interrupt is unaffected.**
    - **1 The corresponding interrupt is cleared.**

HW4 check

- Time and Place
  - April 12th(Fri) 19:00
  - Semi-conductor building 2 floor computer room
    - 400202, 400212
- How to submit
  - .asm
  - I-Campus, until April 12th 18:59
    - format
      - 2012310000_HW4.asm