

마이크로컨트롤러 기초(#514112)

#.6 Interrupt

한림대학교
전자공학과 이선우

Contents

- ▶ **Interrupt**
 - ▶ Interrupt mechanism
 - ▶ MSP430 IRQ Introduction
 - ▶ P1/P2 Interrupt Capability
 - ▶ Example program

인터럽트(Interrupt)란?

- ▶ MCU가 여러 사건(event)의 발생을 인식하는 방법
 - ▶ Polling:
 - ▶ 사건의 발생 유무(예, 입력 변화)를 지속적인 감시를 통해 알아냄
 - ▶ 장점: 동작이 직관적이라 이해/프로그래밍 용이
 - ▶ 단점: 입력 감시에 집중하므로 효율 저하
 - ▶ Interrupt: 하드웨어가 자동적으로 사건 발생을 감지하여 CPU에 알려주는 방법
 - ▶ 장점: 사건이 발생 때만 처리, 고효율
 - ▶ 단점: 보다 복잡한 하드웨어/프로그래밍 필요
- ▶ 여기서 event란?
 - ▶ MCU 내부 또는 외부의 다양한 사건 발생원(interrupt sources)이 만들어 내는 특정 상태
 - ▶ 예, 외부 스위치 입력(전원SW), 타이머 타임아웃, ADC의 변환 완료 등등

Interrupt Mechanism

- ▶ 어떤 작업을 하는 중간(synchronous)에 예기치 않은 사건이 발생(asynchronous events)하여 이를 먼저 처리하는 방식
- ▶ 일상 생활 예: 독서 중 전화가 걸려온 경우
 - ▶ 전화 걸려 옴 : 인터럽트 소스가 인터럽트 요청(**interrupt request**)
 - ▶ 책의 현재 페이지 표시: 현재까지 수행했던 작업을 보관 (이를 **context saving**이라 함. 기본적으로 PC(program counter), SR(status register)을 stack memory에 저장함)
 - ▶ 전화 응대: 사건에 적합한 처리를 수행. 이를 **Interrupt Handler, Interrupt Service Routine(ISR**로 표시)이라함.
 - ▶ 다시 책 읽음: 원래 작업(task)으로 돌아감. Return from IRQ. 저장되었던 정보(PC, SR)를 이용.

MSP430 Interrupts procedures

- ▶ 인터럽트 처리 방법/순서 (manual 2.2.3)
 1. 현재 수행 중인 명령어 완료
 2. PC & SR 스택에 저장(Push to stack)
 3. (여러 인터럽트가 발생했다면) 최우선 사건(highest priority event)을 먼저 처리함.
 4. 단일소스(single source)이면 자동적으로 인터럽트 요청 플래그(request flag) reset.
 5. SR clear됨 (SCGo bit 제외)→GIE=0이 되므로 다른 IRQ disable.
 6. PC ← 인터럽트 벡터 내용 (content of interrupt vector)
- ▶ 인터럽트 벡터란?
 - ▶ 특정 사건 (즉, 특정 IRQ. Source)들 각각에 대하여 미리 정해져 있는 프로그램 메모리 상의 주소(address in program memory).
 - ▶ Ex. PIC16 시리즈: 0x0004, MSP430: 0xFFFFE ~ 0xFFC0
 - ▶ 위의 처리 마지막 과정ISR/Irq. Handler 루틴이 작성되어 있는 시작 주소

MSP430FG461x Irq. sources

Irq. vector

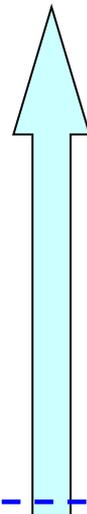
Table 3. Interrupt Sources, Flags, and Vectors of MSP430xG461x Configurations

INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-Up External Reset Watchdog Flash Memory	WDTIFG KEYV (see Note 1 and 5)	Reset	0FFFEh	31, highest
NMI Oscillator Fault Flash Memory Access Violation	NMIIFG (see Notes 1 and 3) OFIFG (see Notes 1 and 3) ACCVIFG (see Notes 1, 2, and 5)	(Non)maskable (Non)maskable (Non)maskable	0FFFCh	30
Timer_B7	TBCCR0 CCIFG0 (see Note 2)	Maskable	0FFFAh	29
Timer_B7	TBCCR1 CCIFG1 ... TBCCR6 CCIFG6, TBIFG (see Notes 1 and 2)	Maskable	0FFF8h	28
Comparator_A	CAIFG	Maskable	0FFF6h	27
Watchdog Timer+	WDTIFG	Maskable	0FFF4h	26
USCI_A0/USCI_B0 Receive	UCA0RXIFG, UCB0RXIFG (see Note 1)	Maskable	0FFF2h	25
USCI_A0/USCI_B0 Transmit	UCA0TXIFG, UCB0TXIFG (see Note 1)	Maskable	0FFF0h	24
ADC12	ADC12IFG (see Notes 1 and 2)	Maskable	0FFEEh	23
Timer_A3	TACCR0 CCIFG0 (see Note 2)	Maskable	0FFEC h	22
Timer_A3	TACCR1 CCIFG1 and TACCR2 CCIFG2, TAIFG (see Notes 1 and 2)	Maskable	0FFEAh	21
I/O Port P1 (Eight Flags)	P1IFG.0 to P1IFG.7 (see Notes 1 and 2)	Maskable	0FFE8h	20
USART1 Receive	URXIFG1	Maskable	0FFE6h	19
USART1 Transmit	UTXIFG1	Maskable	0FFE4h	18
I/O Port P2 (Eight Flags)	P2IFG.0 to P2IFG.7 (see Notes 1 and 2)	Maskable	0FFE2h	17
Basic Timer1/RTC	BTIFG	Maskable	0FFE0h	16
DMA	DMA0IFG, DMA1IFG, DMA2IFG (see Notes 1 and 2)	Maskable	0FFDEh	15
DAC12	DAC12.0IFG, DAC12.1IFG (see Notes 1 and 2)	Maskable	0FFDCh	14
Reserved	Reserved (see Note 4)		0FFDAh	13
		
			0FFC0h	0, lowest

RESET

NMI

Maskable



[2] 13 page

Port 1/2에 의한 Interrupt request

- ▶ 각각의 GPIO 핀을 디지털 입력으로 사용할 때 이 포트(핀)의 값의 변화 (H→L, L→H)를 자동으로 감지하여 인터럽트를 발생시킬 수 있다.
 - ▶ 즉 P1/P2의 8개 포트들은 각각 인터럽트 소스가 될 수 있고 (설정해야 됨) 이 포트에 값이 변하면 IRQ Request
- ▶ 관련 레지스터 이름 및 역할

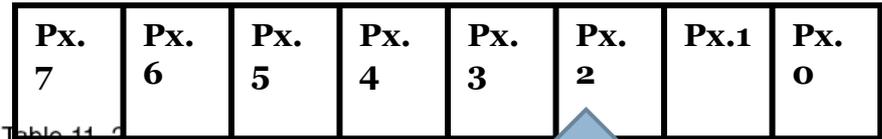
11.3 Digital I/O Registers

The digital I/O registers are listed in Table 11-1 and Table 11-2.

Table 11-1. Digital I/O Registers, P1-P6

Port	Register	Short Form	Address	Register Type	Initial State
P1	Input	P1IN	020h	Read only	-
	Output	P1OUT	021h	Read/write	Unchanged
	Direction	P1DIR	022h	Read/write	Reset with PUC
	Interrupt Flag	P1IFG	023h	Read/write	Reset with PUC
	Interrupt Edge Select	P1IES	024h	Read/write	Unchanged
	Interrupt Enable	P1IE	025h	Read/write	Reset with PUC
	Port Select	P1SEL	026h	Read/write	Reset with PUC
	Resistor Enable	P1REN	027h	Read/write	Reset with PUC
P2	Input	P2IN	028h	Read only	-
	Output	P2OUT	029h	Read/write	Unchanged
	Direction	P2DIR	02Ah	Read/write	Reset with PUC

Reg.: Px[IN/OUT/..], x=1~10



모두 8bit

해당 자리의 비트값(0/1)
에 따라 해당 포트(핀)을
제어!

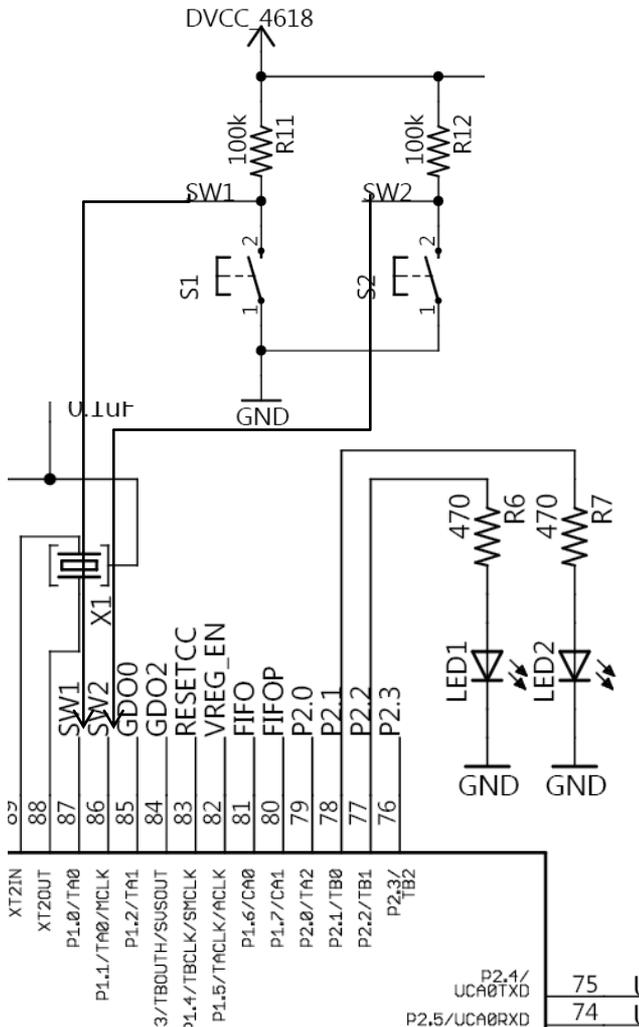
[1] 11-7 page

Control registers

▶ 관련 control reg.

- ▶ P1(2)IE: Interrupt Enable: 인터럽트를 사용할 지 말지 결정.
각 비트 1=enable, 0=disable
- ▶ P1(2)IES: Interrupt Edge Select: 반응하는 신호 타입 결정
 - ▶ Bit=0: L to H (rising edge라 부름), Bit=1: H to L(falling edge)
- ▶ P1(2)IFG: Interrupt Flag: 인터럽트 발생을 알리는 신호
 - ▶ Bit=1: 지정된 비트의 핀의 변화를 감지하면 자동으로 SET됨.
 - ▶ 반드시 software에서 RESET(o)해야 함
- ▶ Status Reg. 안에 있는 GIE(Global Interrupt Enable)
 - ▶ 모든 인터럽트 요청을 거부할지 말지를 결정하는 비트
 - ▶ GIE=1 이어야 CPU가 인터럽트 동작을 수행함.

Example : SW1/2를 P1 IRQ source로 사용



▶ 설정 사항

- ▶ P1.0, P1.1 포트를 디지털 입력으로 설정(P1DIR)
- ▶ P1.0/1을 인터럽트 사용하도록 설정(P1IE)
- ▶ 어떤 종류 신호(rising/falling edge)에 반응하는지 결정(P1IES)
- ▶ GIE=1

▶ 작동 순서

- ▶ S1 누름 → falling edge 신호 발생
→ P1.0 IRQ request → IRQ process 수행 (지정된 ISR 실행)
→ 원래 하던 일을 다시 수행함.

C program에서 IRQ 다루는 법

- ▶ 원하는 인터럽트 소스에 대해 초기화(즉 Irq. Enable) 하는 법
 - ▶ IAR C compiler에서 제공하는 내장형 함수(intrinsic functions) 사용 (EW430 Help에서 C/C++ compiler reference guide 문서 215쪽 참조할 것!)
 - ▶ `__enable_interrupt()`, `__disable_interrupt()` : 앞에 1개 `underbar` 있음.
- ▶ Irq. Handler (ISR) 함수 작성
 - ▶ 각 Irq. Request에 대한 응답(대책)을 별도 함수로 만들어야 함.
 - ▶ 이를 위해 특수 지시어(directives) `#pragma`와 `__interrupt` 사용함.

데모용 예제 프로그램

P1 Irq. Handler

```
#pragma vector=PORT1_VECTOR
__interrupt void port1_handler(void)
{
    P2OUT &= ~(0x06); //turn off LEDs

    //clear P1IFG
    if(P1IFG & 0x01) { //SW1 pushed
        P1IFG &= ~(0x01); //clear P1.0 IFG
        P2OUT |= 0x02;
    }
    else if(P1IFG & 0x02) {
        P1IFG &= ~(0x02);
        P2OUT |= 0x04;
    }
}
```

```
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;
    // Stop watchdog timer

    P2DIR |= 0x06;

    //setup P1.0 P1.1 input and
    irq. enable
    P1DIR &= ~(0x03);
    P1IES |= 0x03; //H to L
    P1IE |= 0x03; //irq.
    enable
    _enable_interrupt();
    //intrinsic ftn.

    while(1); //foreground
    routine
}
```

- 원래 하던 일 (여기서는 아무 일도 하지 않음)
- ISR로 갔다 옴.