



AVR - HW8

임베디드스쿨1기

Lv1과정

2021. 07. 16

이충재

13.4.3 DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

DDRB는 핀의 입출력 모드를 결정합니다.

입력으로 사용시 0, 출력으로 사용시 1로 놓습니다.

별도로 설정하지 않을시 자동으로 0이 됩니다.

ex) PB6을 출력으로 두고 싶을 때 DDB6을 1로 둡니다.

DDRB = 0x40; // 0x40 = 0b01000000

Atmega328p 에는 PB0 ~ 7, PC0 ~ PC6, PD0 ~ PD7 까지 있다.

PC와 PD도 PB와 마찬가지로 DDRC 와 DDRD를 설정하므로 입출력을 결정할 수 있다.

13.4.2 PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PORTB는 출력을 High로 둘지 Low로 둘지 결정한다.
1로두면 high(5V), 0로두면 low(0V) 이다.
별도로 출력을 지정하지 않을시 자동으로 low가 됩니다.

ex) PB5를 high로 두고 싶을때

DDRB = 0x20;

PORTB = 0x20;

*PORTB 사용시 주의점

반드시 DDRB설정을 하고 PORTB를 설정하여야 한다.

13.4.4 PINB – The Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

PINB는 입력을 받을 때 사용한다.

입력을 high(5V)으로 받을 때 1, 입력을 low(0V)로 받을 때 0

ex) PB0으로 입력이 low일때, PB5 출력을 high로 만들고 싶을 때

```
if(PINB & 0x01 == 0x00)
{
    PORTB = 0x20;
}
```

*PINB 사용시 주의점

반드시 DDRB설정 먼저하고 PINB를 설정한다.

DDRXn 과 PORTXn 의 상태에 따른 동작

Table 13-1. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output low (sink)
1	1	X	Output	No	Output high (source)

내부 풀업저항을 사용하려면 DDXn은 0, PORTB는 1, PUD는 0으로 두어야 한다.
Tri - state (HI - Z) : 0도 아니고 1도 아닌 플로팅상태

< MCUCR 레지스터 >

13.4.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

1. LED 켜고 끄기

```
#include <avr/io.h>
```

← DDRB, PORTB 등이 정의 되어있는 헤더파일

```
#define F_CPU 16000000UL
```

← 클럭속도 16000000HZ 정의,
<util/delay.h> 헤더파일 사용시 반드시 필요

```
#include <util/delay.h>
```

← delay함수가 들어있는 헤더파일

```
int main(void)
```

```
{
```

```
    DDRB = 0x20;
```

← PB5 출력모드, 나머지는 입력모드로 설정

```
    while(1)
```

```
    {
```

```
        PORTB = 0x00;
```

PB0 ~PB7 모두 출력 low

```
        _delay_ms(1000);
```

1000ms 지연

```
        PORTB = 0x20;
```

PB5 출력 High, 나머지 Low

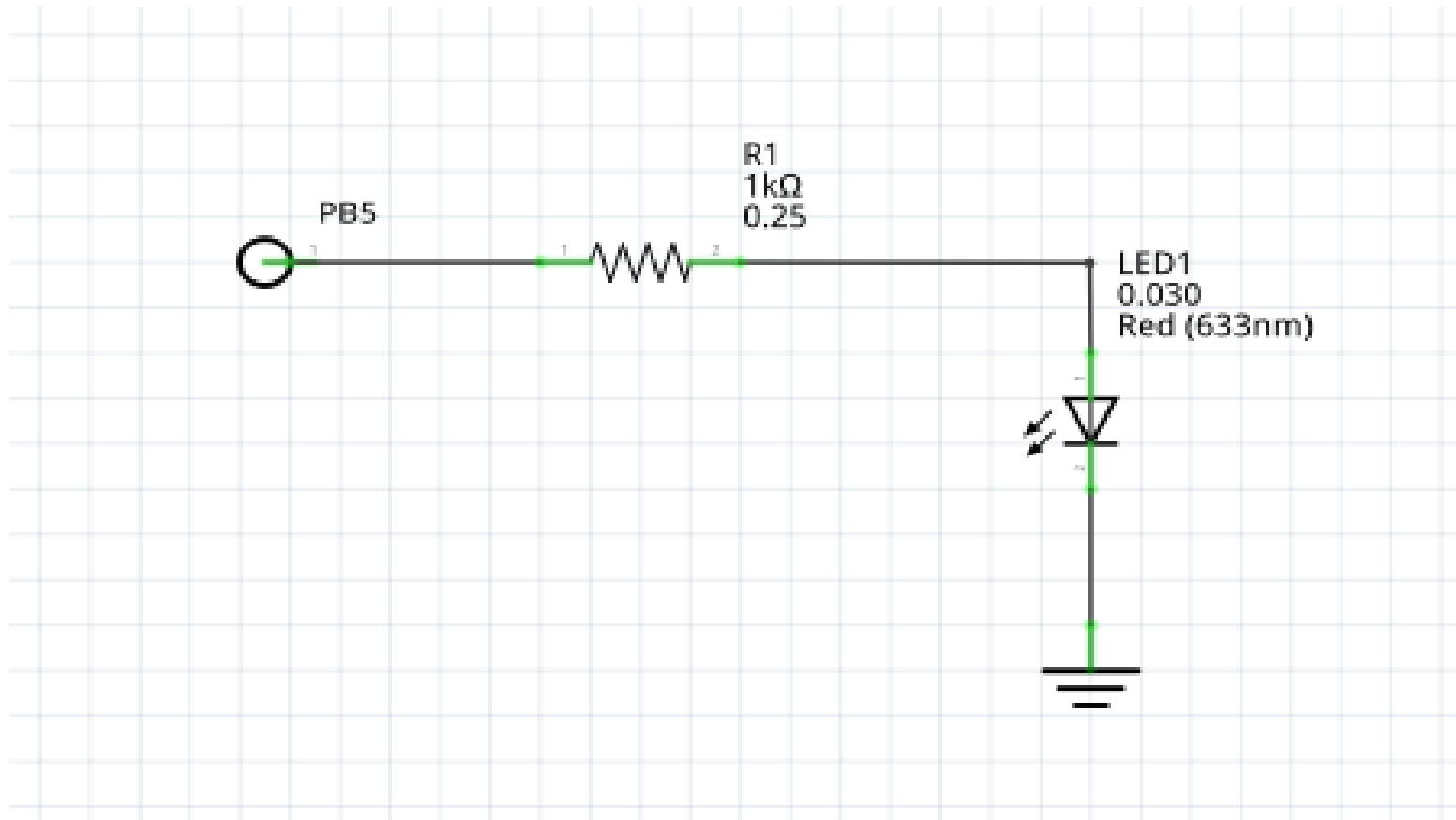
```
        _delay_ms(1000);
```

1000ms 지연

```
    }
```

```
    return 0;
```

```
}
```



PB5에 high신호가 들어오면 led가 켜지는 회로.

인터럽트란?

급한 작업을 먼저 실행하고 작업이 끝나면 다시 하던 작업을 하는 동작.

인터럽트 수행과정

인터럽트 요청 → 인터럽트 허용 여부 판단 → 복귀주소 저장 → 인터럽트 서비스 루틴 실행
→ 인터럽트 서비스 루틴 종료 후 인터럽트 이전 프로그램으로 복귀

예시) 밤 늦게 부모님 몰래 컴퓨터 게임하는 상황

부모님이 방으로 다가오는 발소리 들림 → 도중에 중단해도 되는 게임인지 판단
→ 하던 게임을 일시 정지 → 방 불을 끄고 침대에 누움 → 부모님이 방을 확인하고 돌아가면 다시 게임시작

6.3.1 SREG – AVR Status Register

The AVR status register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

SREG 레지스터는 avr의 상태를 나타내는 레지스터이다.

여기에서 I비트는 인터럽트를 허용해주는 비트이다.

인터럽트를 사용하기 위해서는 반드시 I비트가 set되어야 한다.

일단 인터럽트가 발생하면 ISR프로그램이 실행되고

SREG 레지스터의 I비트는 0으로 clear된다. ISR이 끝나면 I비트가 다시 1로 set된다.

SEI명령어와 CLI명령어를 통해 I비트를 set또는 clear시킬수 있다.

12.2.4 PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	-	-	-	-	-	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCICR은 PCINT에서 발생한 인터럽트를 허용해주는 권한을 가지는 레지스터이다.

PCIE 0 ~ 2 는 다음과 같이 권한을 나눠 가진다.

PCIE0 – PCINT0 ~ 7 // PCINT0 벡터에 의하여 인터럽트 발생
PCIE1 – PCINT8 ~ 15 // PCINT1 벡터에 의하여 인터럽트 발생
PCIE2 – PCINT16 ~ 23 // PCINT2 벡터에 의하여 인터럽트 발생

* PCINT는 PCMSK레지스터에 포함된 비트이다.
(다음 장에 설명)

12.2.5 PCIFR – Pin Change Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	-	-	-	-	-	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCIFR 레지스터는 인터럽트 요청을 받았을때 PCIF비트에 표시를 하는 레지스터

PCIF0은 PCINT 0 ~ 7로 부터 인터럽트 요청을 받을 때 1이 된다.

PCIF1은 PCINT 8 ~ 15로 부터 인터럽트 요청을 받을 때 1이 된다.

PCIF2는 PCINT 16 ~ 23으로 부터 인터럽트 요청을 받을 때 1이 된다.

인터럽트가 실행되면 PCIF비트는 다시 0이 된다.

****** 일반적으로 레지스터의 비트에 0을 써서 클리어 하지만
PCIF비트는 직접 1로 세팅하여 0으로 클리어 할 수 있다.

12.2.8 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCINT는 해당하는 핀의 상태변화를 감지하여 인터럽트를 발생시키는 비트이다.
PCINT비트가 1로 set되어 있어야 인터럽트를 발생시킬수 있다.

PCMSK1은 PCINT8 ~ 15, PCMSK2는 PCINT16 ~ 23 비트를 가지고 있다.

** 주의: 인터럽트를 사용하기 위해서는
SREG – I비트와 해당하는 PCICR – PCIE비트가 set 되어야 한다.

12.2.1 EICRA – External Interrupt Control Register A

The external interrupt control register A contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
(0x69)	–	–	–	–	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ISC 0n – INT0핀의 인터럽트 발생조건 결정
ISC 1n – INT1핀의 인터럽트 발생조건 결정

Table 12-1. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

// INT1핀 low 일때
// INT1핀 변화 있을때
// INT1핀 상승엣지
// INT1핀 하강엣지

ISC0n 도 ISC1n과 똑같이 동작한다.

12.2.2 EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	-	-	-	-	-	-	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

INT1 비트에 1을 쓰면 INT1핀으로 부터 발생하는 인터럽트 허용
 INT0 비트에 0을 쓰면 INT0핀으로 부터 발생하는 인터럽트 허용

12.2.3 EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	-	-	-	-	-	-	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

엣지 또는 논리적 변화에 의해 INT1핀 또는 INT0핀이 인터럽트를
 요청하면 INTF1 비트 또는 INTF0 비트는 1로 세팅된다.
 인터럽트 루틴이 실행되면 다시 0으로 돌아간다.

```
#include <avr/io.h>
#include <avr/interrupt.h>
```

```
int main(void)
```

```
{
```

```
    DDRB = 0x10;
```

```
    DDRB = ~ (1 << DDB5);
```

```
    PORTB = 0x10;
```

```
    sei();
```

```
    PCICR |= (1 << PCIE0);
```

```
    PCMSK0 |= (1 << PCINT5);
```

```
    while(1);
```

```
    return 0;
```

```
}
```

PB4를 출력으로 설정

PB5만은 입력으로 설정

PB5 출력 high

SREG 레지스터 I비트 set

PCICR 레지스터의 PCIE0비트 set

PCMSK0레지스터의 PCINT5비트 set

```
ISR(PCINT0_vect)
```

Pcint0 벡터에 의한 인터럽트 서비스 루틴

```
{
```

```
    PORTB ^= 0x10;
```

PB5를 토글한다.

```
}
```

- 인터럽트 서비스 루틴(ISR)

인터럽트 발생시 실행중인 프로그램을 멈추고 ISR을 실행한다.

ISR이 끝나면 다시 원래 진행하던 프로그램으로 되돌아간다.

