AVR - HW10

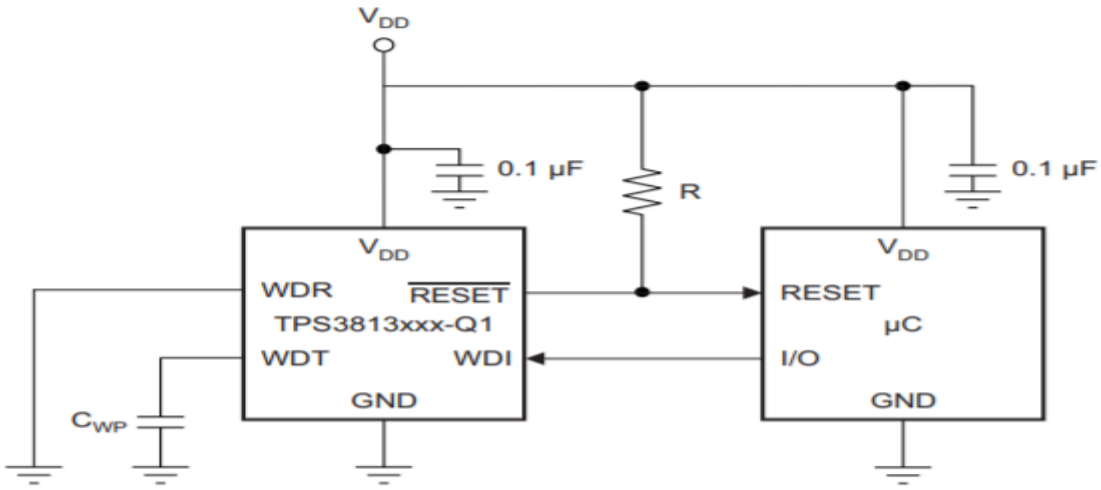
임베디드스쿨1기
Lv1과정
2020. 11. 17
손표훈

1. WATCHDOG(WATCHDOG TIMER)

(1) WATCHDOG? 컴퓨터의 오동작을 감지하고 복구(reset)을 시켜주는 하드웨어 타이머다.

(2) WATCHDOG Timer의 동작

- WATCHDOG Timer는 사용자가 설정한 시간이 될 때까지 타이머를 계수한다.
- Timer에 설정된 시간이 끝나면 WATCHDOG Timer가 "Time Out"신호를 발생한다.
- 이 "Time Out"신호가 MCU내부의 Reset 라인으로 입력되어 MCU를 Reset시킨다.
- 정상적으로 MCU가 동작 할 때 WATCHDOG이 "Time Out"되지 않게 WATCHDOG에 설정된 "WATCHDOG KICK" 신호주기 만큼 MCU -> WATCHDOG으로 KICK신호를 보내야한다.
- 만약 오류가 발생하여 MCU가 KICK신호를 보내지 못하면 WATCHDOG은 "Time Out"신호를 보내 MCU를 Reset시킨다.



1. WATCHDOG(WATCHDOG TIMER)

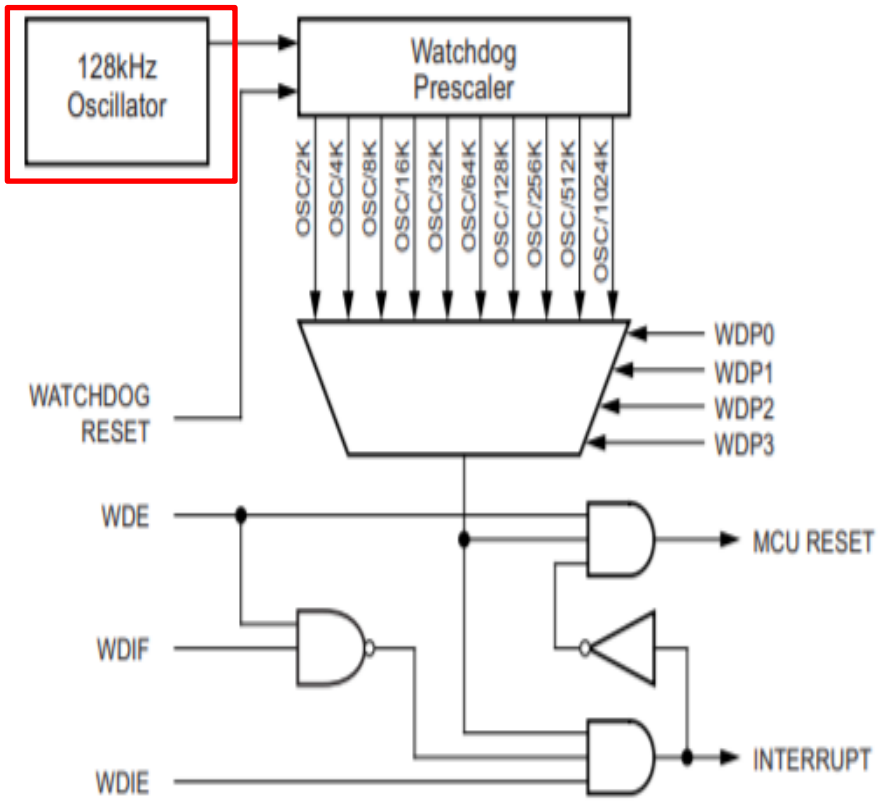
(3) ATmega328P의 내부 WATCHDOG

Features

별도의 oscillator가 내장되어 있음

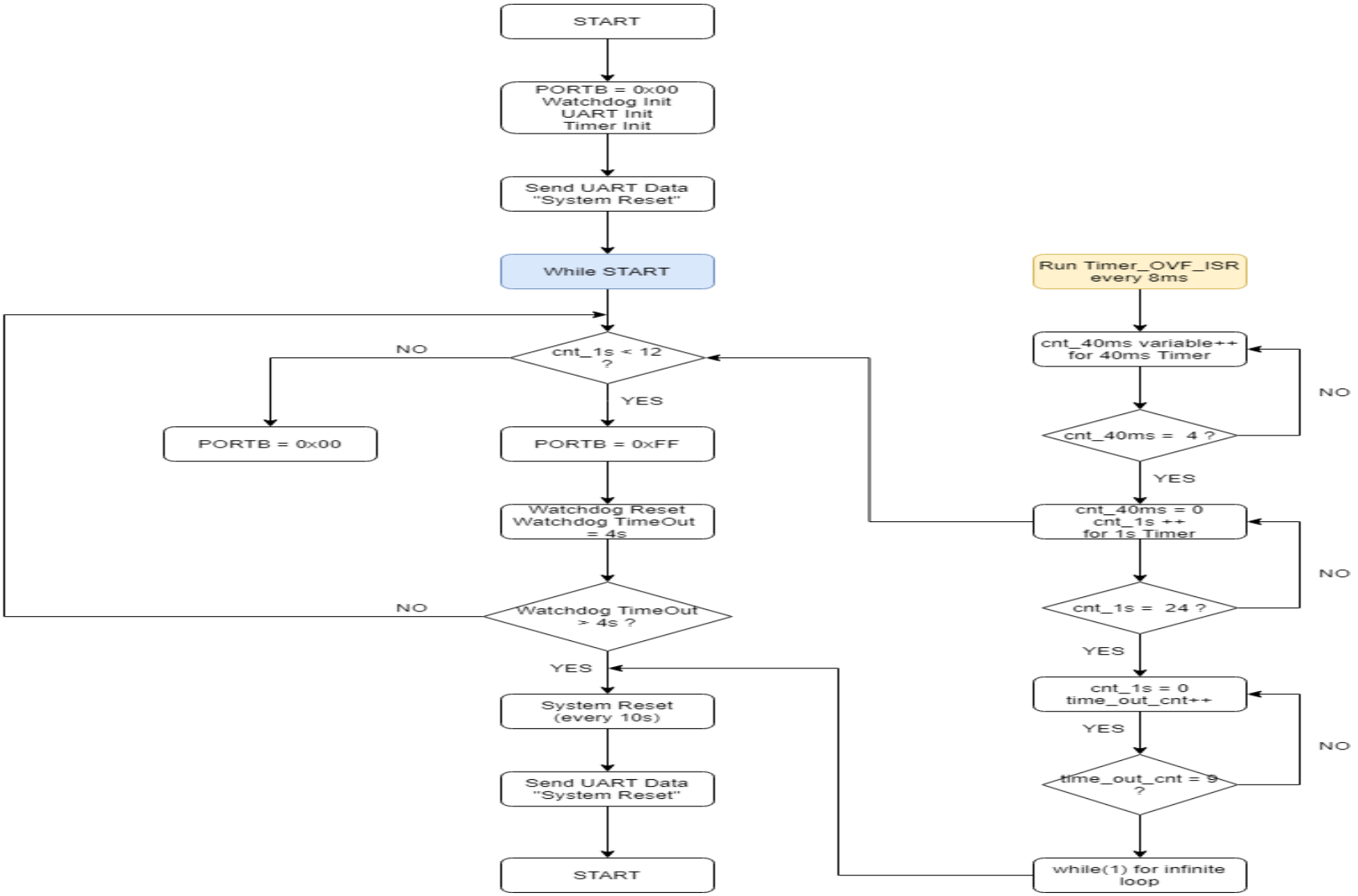
- Clocked from separate on-chip oscillator
- 3 operating modes
  - Interrupt
  - System reset
  - Interrupt and system reset
- Selectable time-out period from 16ms to 8s
- Possible hardware fuse watchdog always on (WDTON) for fail-safe mode

Time Out주기를 16ms ~ 8s까지 설정 할 수 있음



1. WATCHDOG(WATCHDOG TIMER)

(3) ATmega328P의 Watchdog Test SW 순서도



1. WATCHDOG(WATCHDOG TIMER)

(3) ATmega328P의 Watchdog Test SW

```
volatile unsigned char cnt_40ms = 0;
volatile unsigned char cnt_1s = 0;
volatile unsigned char time_out_cnt = 0;

SIGNAL(TIMER0_OVF_vect)
{
    cnt_40ms++;
    if(cnt_40ms == 4)
    {
        cnt_1s++;
        cnt_40ms = 0;
    }

    if(cnt_1s == 24)
    {
        time_out_cnt++;
        cnt_1s = 0;
    }

    if(time_out_cnt == 9)
    {
        while(1);
    }
}
```

TimeOut 테스트용
10초 뒤 무한루프 실행

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>
#include <stdlib.h>
#include <avr/wdt.h>

int main(void)
{
    /* Replace with your application code */
    cbi(SREG, 7);
    PORTB = 0x00;
    DDRB = 0xFF;
    sbi(SREG, 7);

    Timer0_init();
    Wdt_init();
    UART_INIT();
    UART_string_transmit("System Reset\n");

    time_out_cnt = 0;

    while (1)
    {
        PORTB = (cnt_1s < 12) ? 0xFF : 0x00;
        wdt_reset();
    }
}
```

Watchdog 관련 함수 헤더파일

초기화 함수

1초마다 LED ON/OFF
와치독 TimeOut시 4초로ON/OFF

와치독 리셋 함수
와치독 TimeOut주기는 4초

1. WATCHDOG(WATCHDOG TIMER)

(3) ATmega328P의 Watchdog Test SW

```
void Wdt_init(void)
{
    cbi(MCUSR, WDRF);

    sbi(WDTCSR, WDE);
    sbi(WDTCSR, WDCE);

    sbi(WDTCSR, WDP3);
}
```

9.1 MCUSR – MCU Status Register

The MCU status register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 3 - WDE: Watchdog System Reset Enable**

WDE is overridden by WDRF in MCUSR. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared first. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

와치독 초기화를 위해서 MCUSR의 WDRF의 Bit를 0로 Set해야한다.

1. WATCHDOG(WATCHDOG TIMER)

(3) ATmega328P의 Watchdog Test SW

```
void Wdt_init(void)
{
    cbi(MCUSR, WDRF);

    sbi(WDTCSR, WDE);
    sbi(WDTCSR, WDCE);

    sbi(WDTCSR, WDP3);
}
```

WDTCSR – Watchdog Timer Control Register

Bit	7	6	5	4	3	2	1	0	
(0x60)	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	X	0	0	0	

- **Bit 4 - WDCE: Watchdog Change Enable**

This bit is used in timed sequences for changing WDE and prescaler bits. To clear the WDE bit, and/or change the prescaler bits, WDCE must be set.

Once written to one, hardware will clear WDCE after four clock cycles.

- **Bit 3 - WDE: Watchdog System Reset Enable**

WDE is overridden by WDRF in MCUSR. This means that WDE is always set when WDRF is set. To clear WDE, WDR must be cleared first. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

와치독 TimeOut주기를 변경하기 위해
WDCE를 Set해줘야 한다.

와치독 모듈 리셋

1. WATCHDOG(WATCHDOG TIMER)

(3) ATmega328P의 Watchdog Test SW

```
void Wdt_init(void)
{
    cbi(MCUSR, WDRF);

    sbi(WDTCSR, WDE);
    sbi(WDTCSR, WDCE);

    sbi(WDTCSR, WDP3);
}
```

- **Bit 5, 2..0 - WDP3..0: Watchdog Timer Prescaler 3, 2, 1 and 0**

The WDP3..0 bits determine the watchdog timer prescaling when the watchdog timer is running. The different prescaling values and their corresponding time-out periods are shown in Table 10-3.

Table 10-3. Watchdog Timer Prescale Select

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 5.0V
0	0	0	0	2K (2048) cycles	16ms
0	0	0	1	4K (4096) cycles	32ms
0	0	1	0	8K (8192) cycles	64ms
0	0	1	1	16K (16384) cycles	0.125s
0	1	0	0	32K (32768) cycles	0.25s
0	1	0	1	64K (65536) cycles	0.5s
0	1	1	0	128K (131072) cycles	1.0s
0	1	1	1	256K (262144) cycles	2.0s
1	0	0	0	512K (524288) cycles	4.0s
1	0	0	1	1024K (1048576) cycles	8.0s
1	0	1	0	Reserved	
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

와치독 TimeOut주기를 4초로 설정한다.
와치독 별도의 오실레이터는 128kHz
128KHz/524288 = 0.244140625Hz = 4.096s

1. WATCHDOG(WATCHDOG TIMER)

(3) ATmega328P의 Watchdog의 3가지 모드

Table 10-2. Watchdog Timer Configuration

WDTON(1)	WDE	WDIE	Mode	Action on Time-out
1	0	0	Stopped	None
1	0	1	Interrupt mode	Interrupt
1	1	0	System reset mode	Reset
1	1	1	Interrupt and system reset mode	Interrupt, then go to system reset mode
0	x	x	System reset mode	Reset

Note: 1. WDTON fuse set to "0" means programmed and "1" means unprogrammed.

*WDTON은 FUSE BIT로 Device Programming시 해당 FUSE Bit를 설정해주면 된다.