

Chapter 10: 1, 3, 25, 31, 32

1) Interrupt

2) \$00, \$28

25) False - The CPU will auto clear the TOV0 flag

31) \$02 \$04 \$06  
PD2 PD3 PB2

32) GICR

LPI R16, 1&lt;&lt;INT0

OUT GICR, R16

#1 #&lt;avr/io.h&gt;

#&lt;avr/interrupt.h&gt;

ISR(TIMER0\_OVF\_vect);

ISR(TIMER2\_OVF\_vect);

int main()

{

DDRB = 0x06;

sei();

TCNT0 = -63;

TCCR0 = 0x03;

TCNT2 = -250;

TCCR2 = 0x03;

TIMSK = (1&lt;&lt;TOIE0) | (1&lt;&lt;TOIE2);

while(1);

return 0;

}

ISR(TIMER2\_OVF\_vect)

{

TCNT2 = -250;

PORTB ^= 0x04;

}



```
2  #include <avr/io.h>
    #include <avr/interrupt.h>
    ISR(TIMERO_OVF_vect);
```

```
int main(void)
{
```

```
    DDRB = 0xFF;
    DIDRD = 0x00;
    PORTD = 0xFF;
    DDRA = 0xFF;
```

```
    TCRO = 0x04;
```

```
    TCNT0 = -16;
```

```
    sei();
```

```
    TIMSK = (1 << TOIE0);
```

```
    while(1)
```

```
    {
        PORT = PIN0;
    }
```

```
    return 0;
```

```
}
```

```
ISR(TIMERO_OVF_vect)
```

```
{
```

```
    TCNT0 = -16;
```

```
    PORTB ^= 0x01;
```

```
}
```



3 Steps:

- overflow is triggered
- finish instruction
- change program counter
- loads vector address
- turn off global interrupt
- reset the triggered interrupt
- turn on global interrupt
- pop the program counter to the correct location

The last instruction is RETI and it resets the interrupt and return it from the ISR

4 ~~TOV2~~ vector Address = 0x000A ~~TOV0~~ vector Address = 0x001A  
External interrupt 1 vector address = 0x0004

if they are triggered simultaneously, it checks which of the interrupts has the lowest vector address and it will service it first. TOV2 ✓

if all 3 are triggered @ the same time, then external interrupt would be serviced first.

5 #include <avr/io.h>  
#include <avr/interrupt.h>  
ISR (TIMER0\_OVF\_vect);

int main(void)  
{

DDRB |= 0x01;  
DDRC = 0xFF;  
DDRD = 0x00;  
PORTD = 0x00;  
PORTD = 0xFF;



```

1 TCNT0 = -20;
  TCCR0 = 0x02;
  TIMSK = 0x01;
  sei();
  while(1)
  {
    PORTC = PIN0;
  }
}

ISR(TIMER0_OVR_vect)
{
  PORTB ^= 0x01;
  TCNT0 = -20;
}

```