

Interrupts



interrupts : sources

external (INTn), pin-change (PCINTn)

timer/counter capture, compare, overflow

ADC

SPI, USART, I2C

interrupts : process

include interrupt support

write a good interrupt handler

configure the interrupt source

enable global interrupts

interrupts : support

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

or version 0.2 or higher of:

```
#include "m_general.h"
```

interrupts : handler

```
ISR(SOURCE_vect) {  
    ...  
}
```

where **SOURCE** is one of:

```
ANALOG_COMP, ADC,  
INT0, INT1, INT2, INT3, INT6, PCINT0,  
TIMER0_COMPA, TIMER0_COMPB, TIMER0_OVF,  
TIMER1_CAPT, TIMER1_COMPA, TIMER1_COMPB, TIMER1_COMPC, TIMER1_OVF,  
TIMER3_CAPT, TIMER3_COMPA, TIMER3_COMPB, TIMER3_COMPC, TIMER3_OVF,  
TIMER4_COMPA, TIMER4_COMPB, TIMER4_COMPD, TIMER4_OVF, TIMER4_FPF,  
TWI, SPM_READY, SPI_STC, USART1_RX, USART1_UDRE, USART1_TX, EE_READY
```

make it quick!

(entering handler automatically clears the interrupt flag)

interrupts : source configuration

configure interrupt trigger

demask the interrupt

interrupts : global enable

enable using

```
sei(); //enable
```

disable using

```
cli(); //disable
```

interrupts : example (ADC)

```
#include "m_general.h"
```

```
...
```

```
set(ADCSRA,ADIE); // demask ADC interrupt
```

```
sei(); // enable global interrupts
```

```
...
```

```
ISR(ADC_vect) { // conversion complete ISR
```

```
    ... // do something quickly
```

```
}
```


interrupts : example (Timer OC)

```
#include "m_general.h"

...

set(TIMSK1,OCIE1B); // demask OCR1B interrupt

sei(); // enable global interrupts

...

ISR(TIMER1_COMPB_vect) { // OCR1B match ISR

    ...

}
```