

Interrupts

(in Bezug auf den Atmega32)

Interrupts sind Routinen, welche den aktuellen Programmablauf unterbrechen und eine im Programm definierte Interrupt-Service-Routine (ISR) ausführen. Sie können durch Veränderungen von Pin-Eingangspegeln, Timern, etc. ausgelöst werden.

Damit Interrupts überhaupt aufgerufen werden können, müssen sie global aktiviert werden (Befehl: sei();) und das spezifische Maskenbit des Interrupts muss gesetzt werden (z.B. für ext. Interrupts im GICR). Oft kann der Interrupt noch genauer angegeben werden, wie z.B., dass er bei einer steigenden oder fallenden Flanke ausgelöst wird (für ext. Interrupts in MCUCSR und MCUCR).

ISR's sollten möglichst kurz gestaltet werden, da es sonst passieren kann, dass während der Routine ein weiterer Interrupt auftritt und z.B. „verschluckt“ (nicht bemerkt) wird. Schleifen, lange Berechnungen etc. sollten also vermieden werden. Sinnvoll ist es oft volatile-Steuervariablen („Flags“) zu verwenden.

Beispielcode:

```
#include <avr/io.h>
#include <avr/interrupt.h> //Stellt Interrupt-Funktionen bereit
#include <avr/iom32.h>      //Erleichtert Benennung der Interrupts
//...
volatile int Interrupt_erkannt;
//...
ISR(INT2_vect) //Interrupt für INT2 ("_vect" ranhengen)
{
    cli(); //deaktiviert Interrupts
    Interrupt_erkannt = 1;
}
//...
int main(void)
{
    Interrupt_erkannt = 0;
    GICR |= (1<<INT2); //Aktiviert INT2 als Interrupt PIN
    MCUCSR |= (1<<ISC2); //Interrupt INT2 bei steigender Flanke
    sei(); //Interrupts global aktivieren
    while(1) //Endloschleife
    {
        //TODO...
        if(Interrupt_erkannt = 1)
        {
            //ein Interrupt wurde erfolgreich ausgelöst
            //DOOTHERSTUFF
        }
        //...
    }
}
```

Aus dem
Atmega32
Datenblatt:

Table 18. Reset and Interrupt Vectors

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	\$000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$002	INT0	External Interrupt Request 0
3	\$004	INT1	External Interrupt Request 1
4	\$006	INT2	External Interrupt Request 2
5	\$008	TIMER2 COMP	Timer/Counter2 Compare Match
6	\$00A	TIMER2 OVF	Timer/Counter2 Overflow
7	\$00C	TIMER1 CAPT	Timer/Counter1 Capture Event
8	\$00E	TIMER1 COMPA	Timer/Counter1 Compare Match A
9	\$010	TIMER1 COMPB	Timer/Counter1 Compare Match B
10	\$012	TIMER1 OVF	Timer/Counter1 Overflow
11	\$014	TIMER0 COMP	Timer/Counter0 Compare Match
12	\$016	TIMER0 OVF	Timer/Counter0 Overflow
13	\$018	SPI, STC	Serial Transfer Complete
14	\$01A	USART, RXC	USART, Rx Complete
15	\$01C	USART, UDRE	USART Data Register Empty
16	\$01E	USART, TXC	USART, Tx Complete
17	\$020	ADC	ADC Conversion Complete
18	\$022	EE_RDY	EEPROM Ready
19	\$024	ANA_COMP	Analog Comparator
20	\$026	TWI	Two-wire Serial Interface
21	\$028	SPM_RDY	Store Program Memory Ready

General Interrupt Control Register – GICR

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – INT1: External Interrupt Request 1 Enable**
- **Bit 6 – INT0: External Interrupt Request 0 Enable**
- **Bit 5 – INT2: External Interrupt Request 2 Enable**

MCU Control Register – MCUCR The MCU Control Register contains control bits for interrupt sense control and general MCU functions.

Bit	7	6	5	4	3	2	1	0	
	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
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	

Table 35. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

INT1 analog zu INT0

MCU Control and Status Register – MCUCSR

Bit	7	6	5	4	3	2	1	0	
	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0						See Bit Description

- **Bit 6 – ISC2: Interrupt Sense Control 2**

The Asynchronous External Interrupt 2 is activated by the external pin INT2 if the SREG I-bit and the corresponding interrupt mask in GICR are set. If ISC2 is written to zero, a falling edge on INT2 activates the interrupt. If ISC2 is written to one, a rising edge on INT2 activates the interrupt. Edges on INT2 are registered asynchronously.