

3. External Interrupt Registers

“ For detailed information about the registers, please refer to the Atmega32p datasheet. [here](#)

From the previous section, we have set up the interrupt handler for external interrupt 0. Now, we will set up the registers to setup/initialize this interrupt. Let's say we want to toggle an LED on pin PB5 with falling edge trigger. We can write the following code in our setup section:

```
#define __SFR_OFFSET 0x00
#include "avr/io.h"

.global main
.global INT0_vect

main:
    ; initialize external interrupt 0
    ldi r16, (1<<ISC01) ; Set ISC01 bit = 1 | trigger on falling edge
    sts EICRA, r16 ; write to EICRA register to set the interrupt trigger condition
    ldi r16, (1<<INT0) ; set INT0 bit = 1 | enable external interrupt 0
    sts EIMSK, r16 ; write to EIMSK register to enable the interrupt

    ; enable global interrupts
    sei

INT0_vect:
    ; main interrupt logic here
    in r16, PORTB
    ldi r17, (1 << 5)
    eor r16, r17 ; Toggle PB5
    out PORTB, r16
    reti
```

Let's break it down register by register:

EICRA (External Interrupt Control Register A)

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	

- This register is used to configure the trigger condition for external interrupts. For INT0, we need to set the ISC01 bit to 1 and ISC00 bit to 0 for falling edge trigger. This is done by loading the value $(1 \ll \text{ISC01})$ into register r16 and then writing it to EICRA.

Table 12-2. 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.

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	

- This register is used to enable or disable external interrupts. To enable INT0, we need to set the INT0 bit to 1. This is done by loading the value $(1 \ll \text{INT0})$ into register r16 and then writing it to EIMSK.

sei (Set Global Interrupt Enable)

This instruction enables global interrupts. It **MUST be** called after setting up the individual interrupt configurations to allow the microcontroller to respond to interrupts.

You should get the rough idea for INT1 as well. You just need to set the ISC11 and ISC10 bits in EICRA for the trigger condition and set the INT1 bit in EIMSK to enable it. Read the datasheet for more details on the trigger conditions for INT1.

Revision #3

Created 2026-03-11 17:23:25 UTC by CH

Updated 2026-03-12 15:43:59 UTC by CH