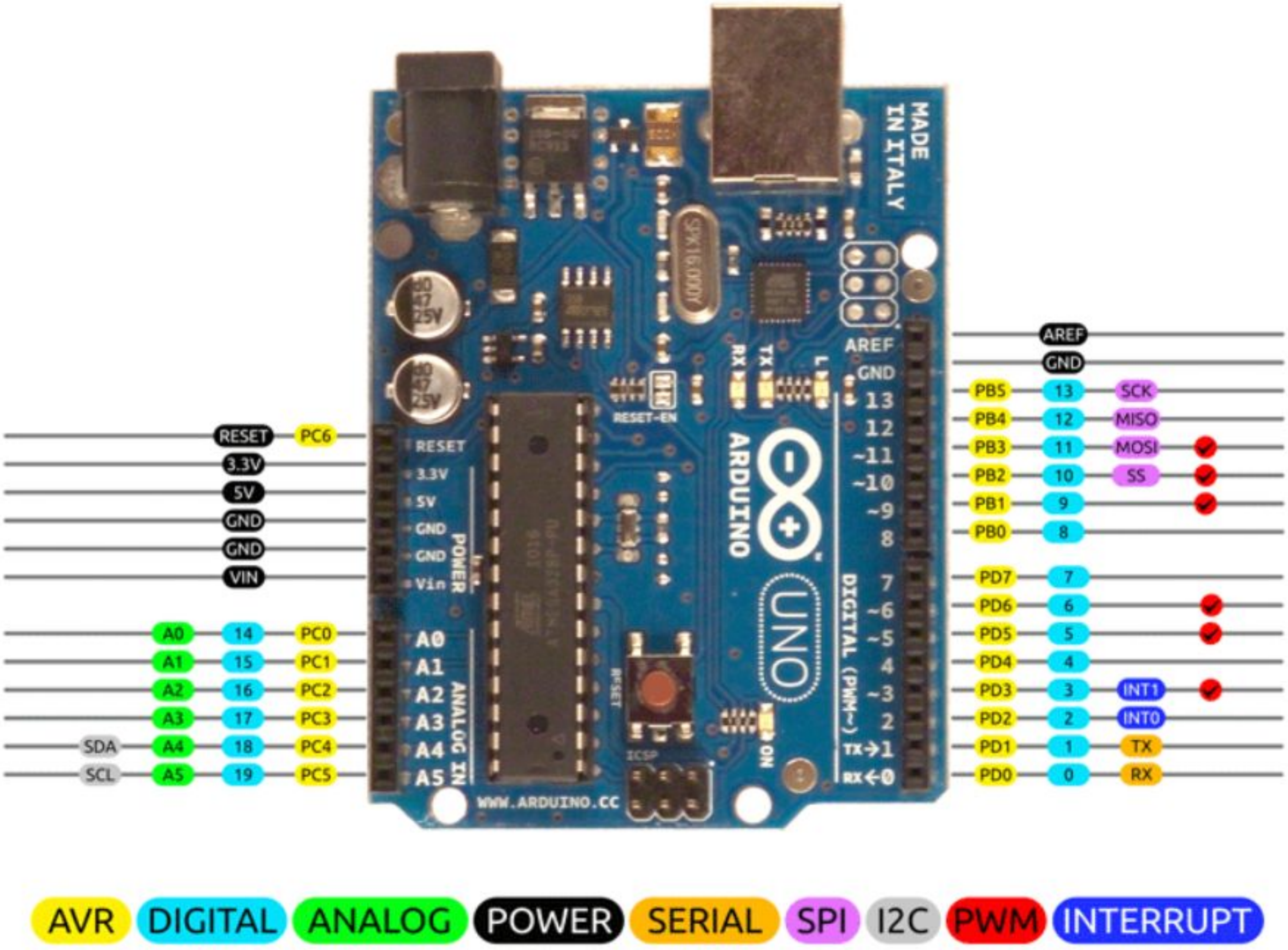
AVR Interfacing

Interrupt

Agenda

- Interrupts Definition
- AVR Interrupts
- AVR External Interrupts
- AVR External Interrupts Programming

I/O Ports


RESET PC6
3.3V
5V
GND
GND
VIN
A0 14 PC0
A1 15 PC1
A2 16 PC2
A3 17 PC3
SDA A4 18 PC4
SCL A5 19 PC5
AREF
GND
PB5 13 SCK
PB4 12 MISO
PB3 11 MOSI
PB2 10 SS
PB1 9
PB0 8
PD7 7
PD6 6
PD5 5
PD4 4
PD3 3 INT1
PD2 2 INT0
PD1 1 TX
PD0 0 RX
ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN
MADE IN ITALY
WWW.ARDUINO.CC
AVR
DIGITAL
ANALOG
POWER
SERIAL
SPI
I2C
PWM
INTERRUPT

Interrupts Definition

- An interrupt is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention.

- When an interrupt event occurs, the microcontroller pause its current task and attend to the interrupt by executing an **Interrupt Service Routine (ISR)** at the end of the ISR the microcontroller returns to the task it had pause and continue its normal operations.

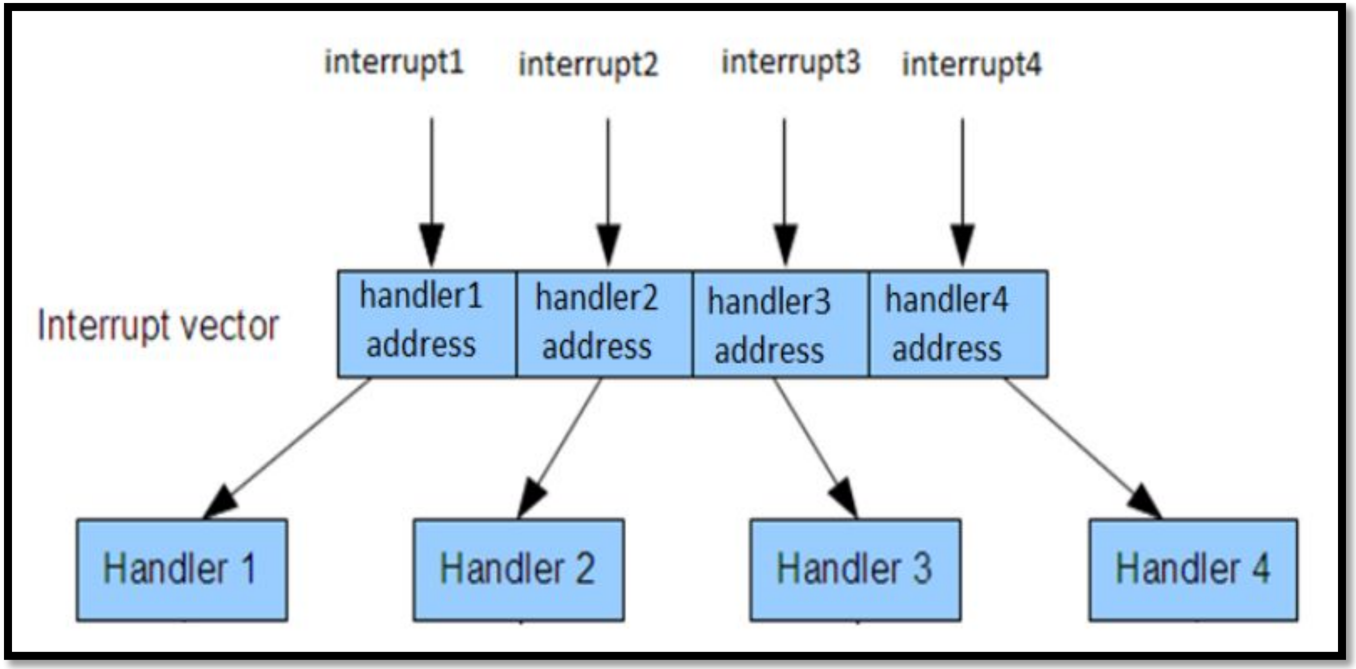
- **Interrupt Service Routine (ISR) or Interrupt Handler**
  Piece of code that should be execute when an interrupt is triggered.

- Each interrupt has its own ISR. Its address in ROM is save in interrupt vector table.

- It should be deterministic and short as possible so not to pause the CPU much of time.

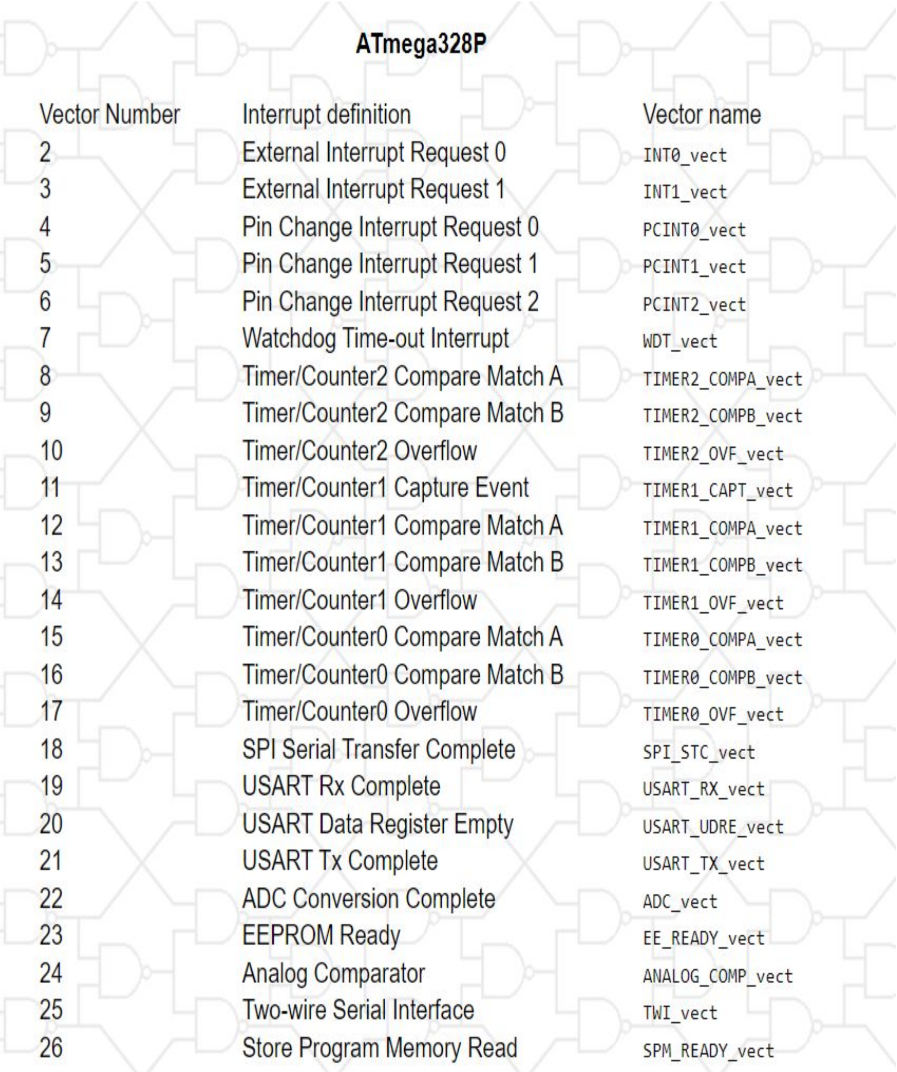
Interrupts Definition

- **Interrupt Vector Table(IVT)**
  - Constant table in Flash EEPROM Program memory.
  - Each interrupt has specific address in the interrupt vector table for its ISR.
  - **Handler = ISR = Code**



AVR Interrupts

- **Interrupt Vector Table(IVT) of Atmega328P**
  - The lower the address the higher is the priority level.
  - All interrupts are assigned individual enable bits to which must be written a logic one together with the Global Interrupt Enable bit



ATmega328P

Vector Number	Interrupt definition	Vector name
2	External Interrupt Request 0	INT0_vect
3	External Interrupt Request 1	INT1_vect
4	Pin Change Interrupt Request 0	PCINT0_vect
5	Pin Change Interrupt Request 1	PCINT1_vect
6	Pin Change Interrupt Request 2	PCINT2_vect
7	Watchdog Time-out Interrupt	WDT_vect
8	Timer/Counter2 Compare Match A	TIMER2_COMPA_vect
9	Timer/Counter2 Compare Match B	TIMER2_COMPB_vect
10	Timer/Counter2 Overflow	TIMER2_OVF_vect
11	Timer/Counter1 Capture Event	TIMER1_CAPT_vect
12	Timer/Counter1 Compare Match A	TIMER1_COMPA_vect
13	Timer/Counter1 Compare Match B	TIMER1_COMPB_vect
14	Timer/Counter1 Overflow	TIMER1_OVF_vect
15	Timer/Counter0 Compare Match A	TIMER0_COMPA_vect
16	Timer/Counter0 Compare Match B	TIMER0_COMPB_vect
17	Timer/Counter0 Overflow	TIMER0_OVF_vect
18	SPI Serial Transfer Complete	SPI_STC_vect
19	USART Rx Complete	USART_RX_vect
20	USART Data Register Empty	USART_UDRE_vect
21	USART Tx Complete	USART_TX_vect
22	ADC Conversion Complete	ADC_vect
23	EEPROM Ready	EE_READY_vect
24	Analog Comparator	ANALOG_COMP_vect
25	Two-wire Serial Interface	TWI_vect
26	Store Program Memory Read	SPM_READY_vect

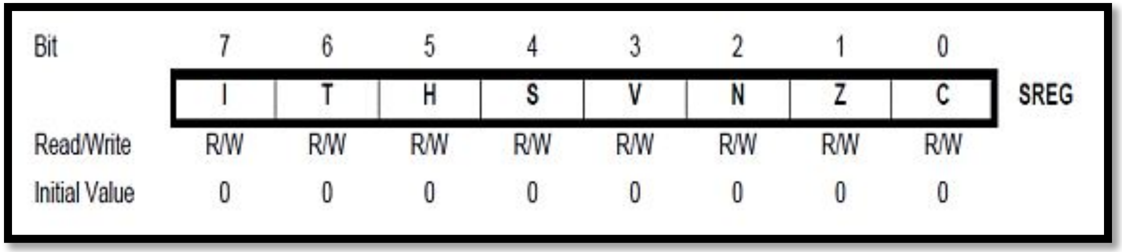
AVR Interrupts

- **What happens when an interrupt occurs in AVR Microcontrollers?**

  - The microcontroller completes the execution of the current instruction.
  - Clears the Global interrupt enable bit.
  - Stores the address of the next instruction that should have been executed (the content of the **PC**) and all the CPU registers are pushed onto the stack.
  - The interrupt vector of the triggered interrupt (ISR start address of this interrupt) is then loaded in the **PC(program counter)** from the **interrupt vector table.**
  - Microcontroller starts execution from that point up until reaches the end of the **ISR.**
  - The address that was stored on the stack in **step 1** is reloaded in the **PC** register.
  - The Global interrupt enable is re-enabled.
  - The micro-controller then continue executing the program.

AVR Interrupts

- **Status Register**
  Contains the status of the flags such as Overflow flag, Negative flag, Zero flag, Carry flag, Half-carry flag, **Global Interrupt mask (I) bit.**



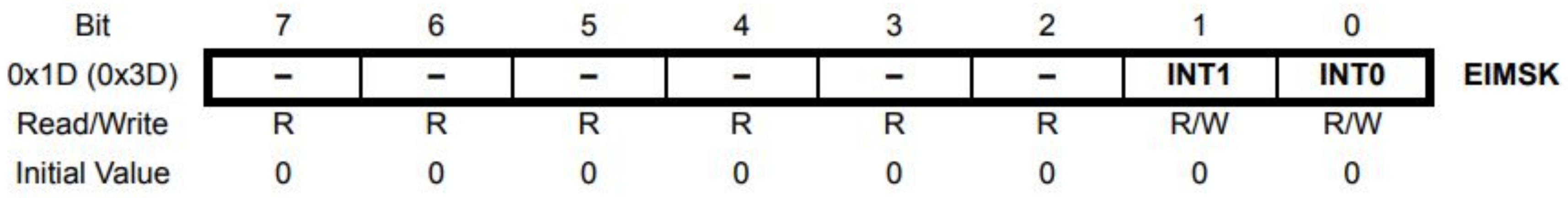
Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **How to set the I-bit?**
  SREG = SREG | (1<<7);
- **How to clear the I-bit?**
  SREG = SREG & (~(1<<7));
- On Atmega328P microcontroller there are two (2) external interrupts:
  - **External Interrupt 0 (INT0)** : Triggered from pin 2 (PD2).
  - **External Interrupt 1 (INT1)** : Triggered from pin 3 (PD3).

AVR External Interrupts Programming

- **External Interrupt Mask Register**

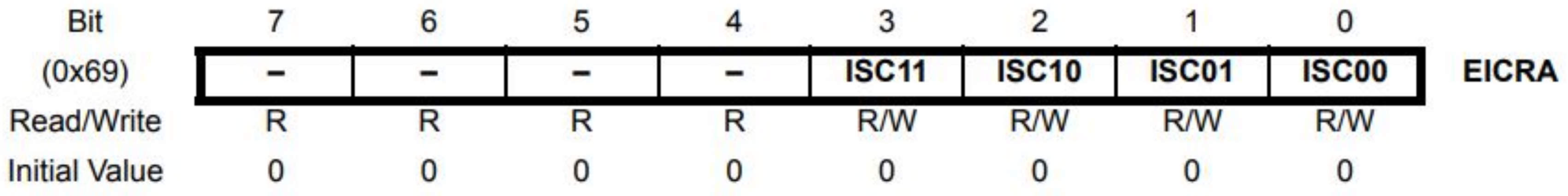
The ATmega 328P supports two external interrupts which are individually enabled by setting bits INT1 and INT0 in the External Interrupt Mask Register



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	

- **EICRA – External Interrupt Control Register A**

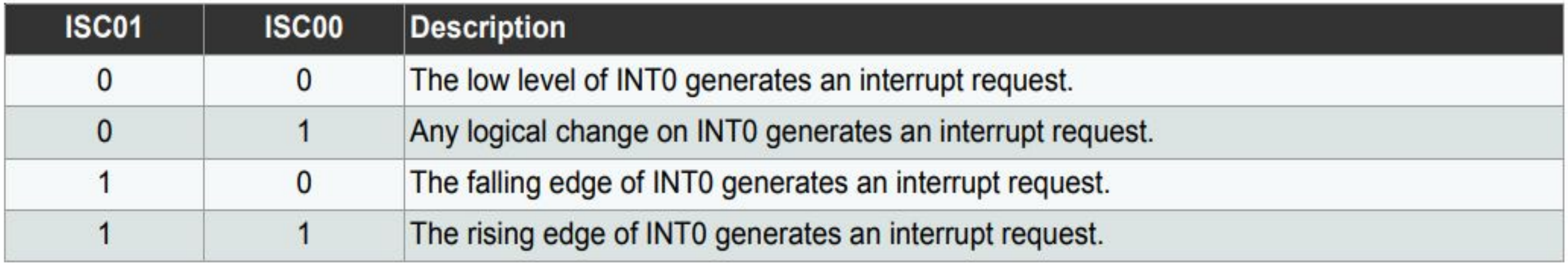
  The external interrupt control register A contains control bits for interrupt sense control



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	

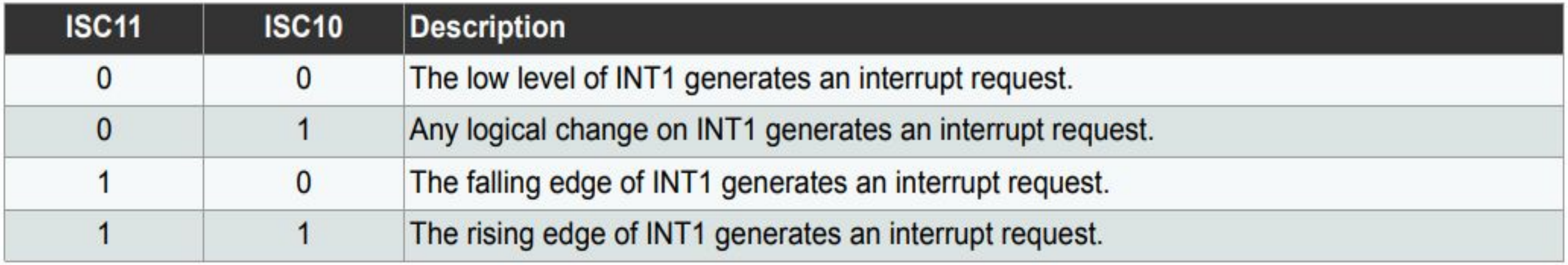
AVR External Interrupts Programming

- **EICRA – External Interrupt Control Register**
  - INT0 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 control

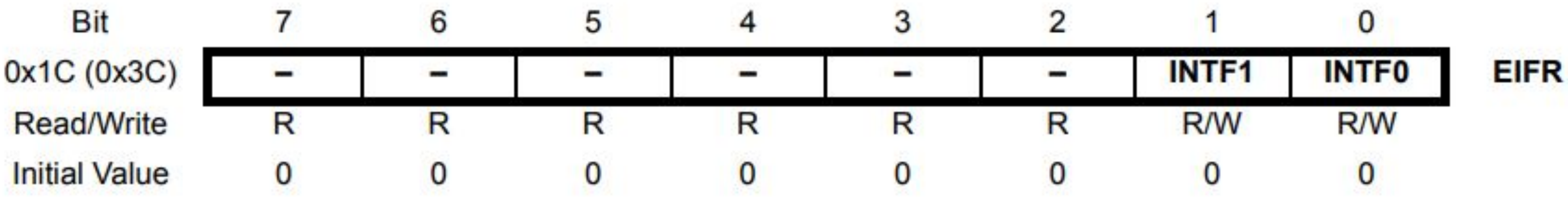


ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

AVR External Interrupts Programming

- **EIFR – External Interrupt Flag Register**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in EIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed.



Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	–	–	–	–	–	–	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	