ATmega328P Interrupts

Narendiran S

July 28, 2020

1 Introduction

- Each interrupt vector occupies two instruction Word(2x16bit) in Atmega328p.
- The complete placement of Reset and Interrupt Vectors in ATmega328P

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface

- The location of reset vector is affected by ***BOOTRST*** fuse.
- The Interrupt vector start address is affected by ***IVSEL*** bit in **MCUCR** register.
- The reset and interrupt Vector placement is shown below as

BOOTRST	**Reset Address**
0	Boot reset address
1	0x0000

IVSEL	**Interrupt Vectors Start Address**
0	0x0002
1	Boot reset address + 0x0002

1.1 Register Description

MCUCR – MCU Control Register

7	6	5	4	3	2	1	0
-	BODS	BODSE	PUD	-	-	IVSEL	IVCE

- When ***IVSEL*** bit is cleared, the interrupt vectors are placed at the start of the flash memory - the application section.
  - When the ***BLB12*** is programmed, interrupts are disabled while executing from boot loader section.
- When ***IVSEL*** bit is set, the interrupt vectors are moved to the beginning of the boot loader section of the flash.
  - When the ***BLB02*** is programmed, interrupts are disabled while executing from application section.
- The actual address of the start of the boot flash section is determined by the ***BOOTSZ*** fuses.
- Writing ***IVSEL*** bit is done by
  (a) Write interrupt vector change enable ***IVCE*** bit to one.
  (b) Write desired value to ***IVSEL*** while Writing zeros to ***IVCE***.
- ***IVCE*** is cleared by hardware.

2 External Interrupts

- Triggered by *INT0*, *INT1* and *PCING[23:0]* pins.
- Pin changed interrupt *PCI2* will be triggered if any *PCINT[23:16]* toggles based on the **PCMSK2** register.
- Pin changed interrupt *PCI1* will be triggered if any *PCINT[14:8]* toggles based on the **PCMSK1** register.
- Pin changed interrupt *PCI0* will be triggered if any *PCINT[7:0]* toggles based on the **PCMSK0** register.
- Due to asynchronous nature of Pin change interrupts, *PCINT[23:0]* can be used to wake up.
- The *INT0* and *INT1* can be triggered by falling, rising or low level choosen by **EICRA** - External Interrupt Control Register.
- Due to asynchronous nature of External interrupts in low level interrupt, *INT0* and *INT1* can be used to wake up.

2.1 Register Description

EICRA - External Interrupt Control Register A

7	6	5	4	3	2	1	0
-	-	-	-	ISC11	ISC10	ISC01	ISC00

- ***ICS11:ICS10*** - Interrupt Sense Control 1 Bit 1 and Bit 0
- ***ICS01:ICS00*** - Interrupt Sense Control 0 Bit 1 and Bit 0

ICS11:ICS10	Description
00	Low level of ***INT1*** generates interrupt
01	Any logic change on ***INT1*** generates interrupt
10	The falling edge of ***INT1*** generates an interrupt request.
11	The rising edge of ***INT1*** generates an interrupt request.

ICS01:ICS00	Description
00	Low level of ***INT0*** generates interrupt
01	Any logic change on ***INT0*** generates interrupt
10	The falling edge of ***INT0*** generates an interrupt request.
11	The rising edge of ***INT0*** generates an interrupt request.

EIMSK – External Interrupt Mask Register

7	6	5	4	3	2	1	0
-	-	-	-	-	-	INT1	INT0

- Enable the corresponding External Interrupt Request Enable bits (***INT1*** or ***INT0***) and ***I-biy*** of status Register **SREG** to enable the External interrupt.

EIFR – External Interrupt Flag Register

7	6	5	4	3	2	1	0
-	-	-	-	-	-	INTF1	INTF0

- When interrupt occurs on the External interrupt pins *INT0* and *INT1*, the corresponding External Interrupt Flag bits (***INTF1*** or ***INTF0***) are set.
- The Flag is cleared by by writing 1 to it in interrupt routine.

PCICR – Pin Change Interrupt Control Register

7	6	5	4	3	2	1	0
-	-	-	-	-	PCIE2	PCIE1	PCIE0

- Enable the corresponding Pin Change Interrupt Enable bits (***PCIE2*** or ***PCIE1*** or ***PCIE0***) and ***I-bit*** of status Register **SREG** to enable the pin change interrupt.
- Setting 1 to ***PCIE2*** bit enabled interupt to occur in *PCINT[23:16]* pins based on **PCMSK2** register.
- Setting 1 to ***PCIE1*** bit enabled interupt to occur in *PCINT[14:8]* pins based on **PCMSK1** register.
- Setting 1 to ***PCIE0*** bit enabled interupt to occur in *PCINT[7:0]* pins based on **PCMSK0** register.

PCIFR – Pin Change Interrupt Flag Register

7	6	5	4	3	2	1	0
-	-	-	-	-	PCIF2	PCIF1	PCIF0

- When interrupt occurs on the Pin Change interrupt pins *PCINT[23:16]*, the ***PCIF2*** Pin Change Interrupt Flag 2 bits is set.
- When interrupt occurs on the Pin Change interrupt pins *PCINT[14:8]*, the ***PCIF1*** Pin Change Interrupt Flag 1 bits is set.
- When interrupt occurs on the Pin Change interrupt pins *PCINT[7:0]*, the ***PCIF0*** Pin Change Interrupt Flag 0 bits is set.
- The Flag is cleared by by writing 1 to it in interrupt routine.

PCMSK2 – Pin Change Mask Register 2

7	6	5	4	3	2	1	0
PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16

PCMSK1 – Pin Chage Mask Register 1

7	6	5	4	3	2	1	0
-	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8

PCMSK0 – Pin Change Mask Register 0

7	6	5	4	3	2	1	0
PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0

- **PCMSK2**, **PCMSK1** and **PCMSK0** are used to select which pin to be enabled for Pin Change Interrupt.

3 Configuring External Interupt

(I) First, the *INT0* or *INT1* pins are configured as Input. (Optional)

(II) Next, the pull-up register may be enabled if needed.

(III) Next, the Interrupt Sense Control Bits are configured for level or edge triggered.

(IV) Finally, the Interrupt are enabled.

(V) Also, Global interrupt is enabled.

(VI) We define the ISR and check the Interrupt Flags if the interrupt occured.

(VII) An example configuration can be seen below.

```
// making PD2 as input for INT0, though not
↪  neeced
DDRD &= ~(1<<2);
// enabling the internal pull-up register for
↪  PD2 for INT0
PORTD |= (1<<2);

// making EICRA's ISC01 and ISC00 as 10 for
↪  falling edge detection at INT0
EICRA |= (1<<ISC01);
EICRA &= ~(1<<ISC00);
// making EIMSK's INT0 as 1 to enable External
↪  Interrput Request for INT0
EIMSK |= (1<<INT0);

// Enabling global Interrupts
sei();
```

```
ISR(INT0_vect)
{
    if((EIFR & (1<<INTF0)) != 0)        //
    ↪  INT0 interrupt as occured
    {
        //toggle Led at pinc 0
        PINC |= (1<<0);
    }
}
```

4 Configuring Pin Change Interupt

(I) First, the *PCINT[23:0]* pins are configured as Input. (Optional)

(II) Next, the pull-up register may be enabled if needed.

(III) Next, which PCINT is selected.

(IV) Finally, the Interrupt are enabled.

(V) Also, Global interrupt is enabled.

(VI) We define the ISR and check the Interrupt Flags if the interrupt occured.

(VII) An example configuration can be seen below.

```
// making PD4 as input for PCI20
DDRD &= ~(1<<4);
// enabling the internal pull-up register for
↪  PD4 for PCI20
PORTD |= (1<<4);

// Selecting the PCINT20 for PCI2 intterupt
PCMSK2 |= (1<<PCINT20);
// Enabling the PCI2 interupt
PCICR |= (1<<PCIE2);

// Enabling global Interrupts
sei();
```

```
ISR(PCINT2_vect)
{
    if((PCIFR & (1<<PCIF2)) != 0)        //
    ↪  PCI2 interrupt as occured
    {
        //toggle Led at pinc 0
        PINC |= (1<<0);

    }
}
```