



روش های handle و قفه ها :  
① pulling : میگرد بود سرکش نمود (در آن زمان توکان کار دیر می شود ) بسیار بد است

## وقفه ها

خارجی خارجی INT0, INT1, INT2  
دستگاه های خارجی متراسده رینها وصل بگویند  
وقفه ها  
داخلی

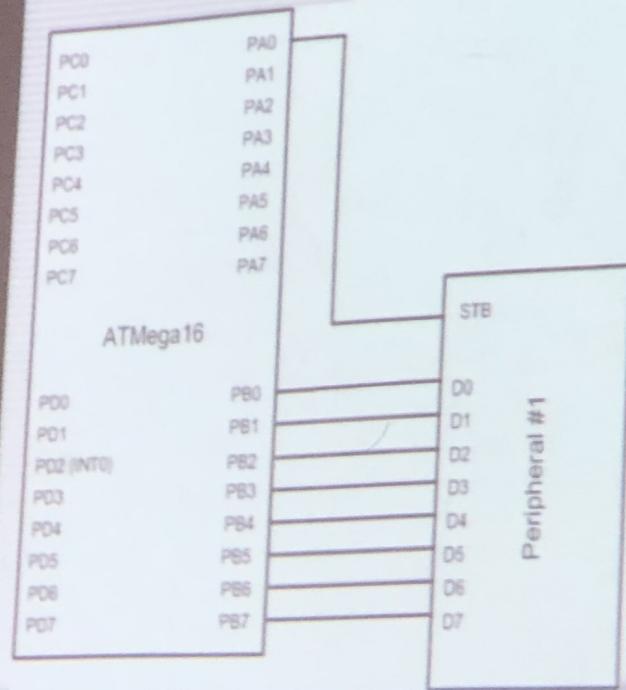
Interrupt driven ②

# در میکروکنترلرهای خانواده AVR

## کلیات و قفه‌ها

روش‌های سرویس‌دهی به تجهیزات جانبی توسط میکروکنترلر:

- روش سرکشی (Polling)
- وسیله جانبی با فعال کردن سر STB تقاضای سرویس می‌نماید.



MOV R0, BytesToBePrinted

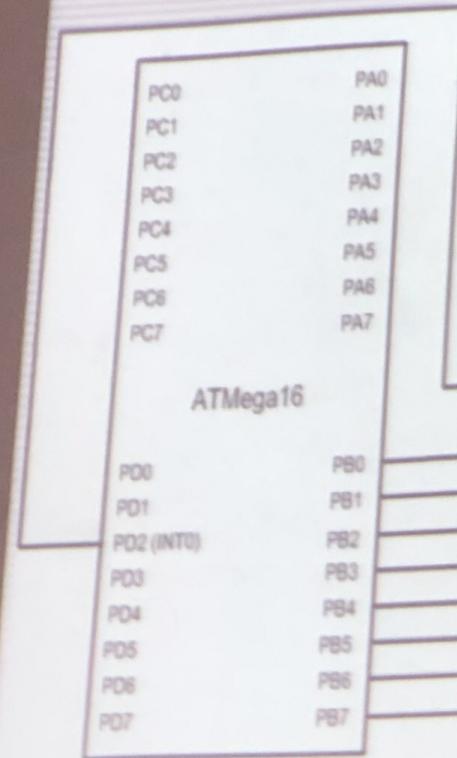
Loop:

SBIS PINA, PA0;  
RJMP LOOP;  
LD R1, Z+;  
OUT PORTB, R1;  
DEC R0;  
BREQ LOOP; (Z Flag = 1) ?

## کلیات وقفه‌ها (ادامه)

روش‌های سرویس دهی به تجهیزات جانبی توسط میکروکنترلر (ادامه):

- روش مبتنی بر وقفه (Interrupt driven)



### Main Program

```
- sei  
-  
-  
-  
-  
-
```

```
;=====;  
;Interrupt Service Routine  
INT0:  
LD R0, X; (X) contains the number of bytes to be printed  
LD R1,Z+;  
OUT PORTB, R1;  
DEC R0;  
ST X, R0;  
RETI
```



# فهرست مطالب

- آنچه در این بخش خواهیم دید:
  - وقفه‌ها
  - وقفه‌های خارجی

Professional Edition

## بازنشانی و کنترل وقفه‌ها

- بازاء هر منبع ایجاد وقفه یک بیت فعال‌ساز وقفه (enable bit) وجود دارد که تا این بیت و نیز بیت فعال‌ساز وقفه سراسری (بیت A)، یک نشوند، آن وقفه در صورت رخداد پذیرفته و اجراء نمی‌شود.
- بعضی از منابع وقفه علاوه بر بیت فعال‌ساز، یک پرچم هم دارند که چنانچه آن وقفه رخداد، آن پرچم یک می‌شود.

## انواع وقفه‌ها در ATMega16

وقفه نوع اول: در این نوع وقفه سه بیت به شرح زیر استفاده می‌شود:

Interrupt Flag



Interrupt Enable bit



Global Interrupt Flag  
(I Flag)



- پرچمی که تقاضای وقفه در آن ثبت می‌شود.

که تعداد رفعه نزدیک دارد

وقفه نوع یک بسته می‌شود

- یک پرچم ویژه برای فعال یا غیرفعال‌سازی وقفه

- پرچم فعال‌سازی وقفه سراسری (پرچم)

## بازنشانی و کنترل وقفه‌ها

انواع وقفه را می‌توان به دو دسته تقسیم کرد:

اولین نوع وقفه: برای این نوع وقفه یک بیت خاص برای سُبَّتِ تُدْن رجید دارد متاک: وقفه سربرز تایم

- این نوع وقفه توسط یک واقعه که پرچم وقفه را فعال می‌کند تحریک می‌شود.
- برای این نوع وقفه‌ها آدرس شروع روال وقفه در شمارنده برنامه قرار می‌گیرد تا بدین ترتیب روال این وقفه اجراء شود.
- پس از اجرای وقفه متعاقباً سخت‌افزار میکروکنترلر، پرچم وقفه رخ داده را غیرفعال می‌سازد. با غیرفعال کردن پرچم‌های وقفه نیز می‌توان وقفه‌ها را غیرفعال کرد.

## بازنشانی و کنترل وقفه‌ها

اولین نوع وقفه (ادامه):

- اگر شرایط رخداد یک وقفه هنگامی که بیت فعال‌ساز وقفه مربوطه غیرفعال است فراهم شود، پرچم وقفه یک شده و در همین وضعیت باقی می‌ماند تا زمانی که وقفه فعال شده (بیت فعال‌ساز وقفه یک شود)، یا پرچم وقفه توسط نرم‌افزار صفر شود.
- بطور مشابه اگر شرایط رخداد وقفه برای یک یا تعداد بیشتری وقفه هنگامی که بیت سراسری وقفه صفر (غیرفعال) است فراهم شود، پرچم(ها)ی وقفه مربوطه یک شده و تا زمان فعال شدن بیت وقفه سراسری در همان وضعیت باقی می‌مانند و بعداً متناسب با ترتیب اولویت، روال‌های وقفه رخ داده اجرا می‌شوند.

## أنواع وقفه‌ها در ATMega16

مثال وقفه نوع اول: وقفه سرریز تایمر TOV0

Interrupt Flag

TOV0

Bit	7	6	5	4	3	2	1	0	TIFR
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	
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	

Interrupt Enable bit

TOIE0

Bit	7	6	5	4	3	2	1	0	TIMSK
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	
Read/Write	RW	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Global Interrupt Flag  
(I Flag)

I

## أنواع وقفه‌ها

دومین نوع وقفه:

در این نوع وقفه دو بیت به شرح زیر استفاده می‌شود:

Interrupt Enable bit



Global Interrupt Flag  
(I Flag)



- این نوع از وقفه‌ها ضرورتاً دارای پرچم وقفه نیستند و در صورت یک بودن بیت فعال‌ساز آنها، در زمانی که شرایط وقفه فراهم است تحریک و اجراء می‌شوند.
- به دلیل نداشتن پرچم وقفه، برای این نوع از وقفه‌ها اگر شرایط وقفه قبل از فعال شدن وقفه برطرف شود، وقفه عملاً آغاز نمی‌شود.

## بازنشانی و کنترل وقفه‌ها

دومین نوع وقفه:

- این نوع از وقفه‌ها ضرورتاً دارای پرچم وقفه نیستند و در صورت یک بودن بیت فعال‌ساز آنها، در زمانیکه شرایط وقفه فراهم است تحریک و اجراء می‌شوند.
- به دلیل نداشتن پرچم وقفه، برای این نوع از وقفه‌ها اگر شرایط وقفه قبل از فعال شدن وقفه برطرف شود، وقفه عملاً آغاز نمی‌شود.

## بازنشانی و کنترل وقفه‌ها

• هنگامی که یک وقفه در حال اجرا باشد و یک وقفه معوق هم داشته باشیم، اجرای وقفه جاری تمام می‌شود، میکروکنترلر به برنامه اصلی بازگشته و یک یا تعداد بیشتری از دستورالعمل‌ها را قبل از اجرای وقفه معوق اجرا می‌نماید.

• توجه داشته باشید که ثبات وضعیت با ورود به روال وقفه بطور اتوماتیک ذخیره نمی‌شود و نیز هنگام بازگشت از روال وقفه بازیابی نمی‌شود. اینکار می‌تواند توسط نرم‌فzar انجام شود.



## بازنشانی و کنترل وقفه ها

- هنگامی که دستورالعمل CLI که پرچم وقفه سراسری را غیرفعال میکند را بکار میبریم، وقفه ها بلا فاصله غیرفعال می شوند.
- بعد از اجرای دستورالعمل CLI، هیچ وقفه ای اجرا نخواهد شد حتی اگر آن وقفه همزمان با اجرای دستورالعمل CLI رخ دهد.
- برنامه های اسلایدهای بعد نشان می دهد که چطور می توان از این نکته استفاده کرد تا از بروز وقفه ها در طی عملیات نوشتمن در حافظه EEPROM جلوگیری کرد. با دستور CLI

# بازنشانی و کنترل وقفه ها

## Assembly Code Example

```
in r16, SREG ; store SREG value  
cli ; disable interrupts during timed sequence  
sbi EECR, EEMWE ; start EEPROM write  
sbi EECR, EEWE  
out SREG, r16 ; restore SREG value (I-bit)
```

## C Code Example

```
char cSREG;  
cSREG = SREG; /* store SREG value */  
/* disable interrupts during timed sequence */  
_CLI();  
EECR |= (1<<EEMWE); /* start EEPROM write */  
EECR |= (1<<EEWE);  
SREG = cSREG; /* restore SREG value (I-bit) */
```

## بازنشانی و کنترل وقفه‌ها

در صورت استفاده از دستورالعمل **SEI** برای فعال کردن وقفه‌ها، همانطور که مثال زیر نشان می‌دهد، دستورالعمل بعد از دستور **SEI** (در مثال زیر دستور **Sleep**، قبل از **SEI**) **هر وقفه معوق اجرا می‌شود**:

### Assembly Code Example

```
sei ; set global interrupt enable
sleep ; enter sleep, waiting for interrupt
; note: will enter sleep before any pending
; interrupt(s)
```

### C Code Example

```
_SEI(); /* set global interrupt enable */
_SLEEP(); /* enter sleep, waiting for interrupt */
/* note: will enter sleep before any pending interrupt(s) */
```

## بازنشانی و کنترل وقفه ها

- زمان پاسخ دادن به وقفه ها برای تمامی وقفه های فعال شده ATmega16 حداقل چهار پالس ساعت به طول می انجامد.
- بعد از چهار پالس ساعت، وقفه رخ داده از آدرس مربوطه اجرا می شود.
- در طی این چهار پالس ساعت، محتوی شمارنده برنامه به درون پشته ارسال می شود.

## بازنشانی و کنترل وقفه ها

- اگر یک وقفه در طی اجرای یک دستورالعمل که در چند چرخه ساعت اجرا می شود،  
فعال شود، این دستورالعمل قبل از شروع روال وقفه کامل می شود. اجرای دستورالعمل ها اعیان است
- اگر یک وقفه هنگامی که میکروکنترلر در **حالت خواب** قرار دارد رخ دهد، زمان پاسخ  
**اجرای روآل وقفه** به اندازه **۴ چرخه ساعت افزایش** می یابد.
- **یک روآل بازگشت از وقفه چهار پالس ساعت بطول می انجامد.**
- در طی این چهار پالس ساعت، **شمارنده برنامه** (دو بایت) از بالای پشته بازیابی و  
اشاره گر پشته به اندازه **دو واحد افزایش** می یابد و بیت **I** در ثبات وضعیت SREG یک  
می شود.

برداشت از یک سری لز وقفه ها استفاده نکنیم  
از در آرس خاص به یک آدرس دیر صفحه زرگنج

# وقفه ها

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

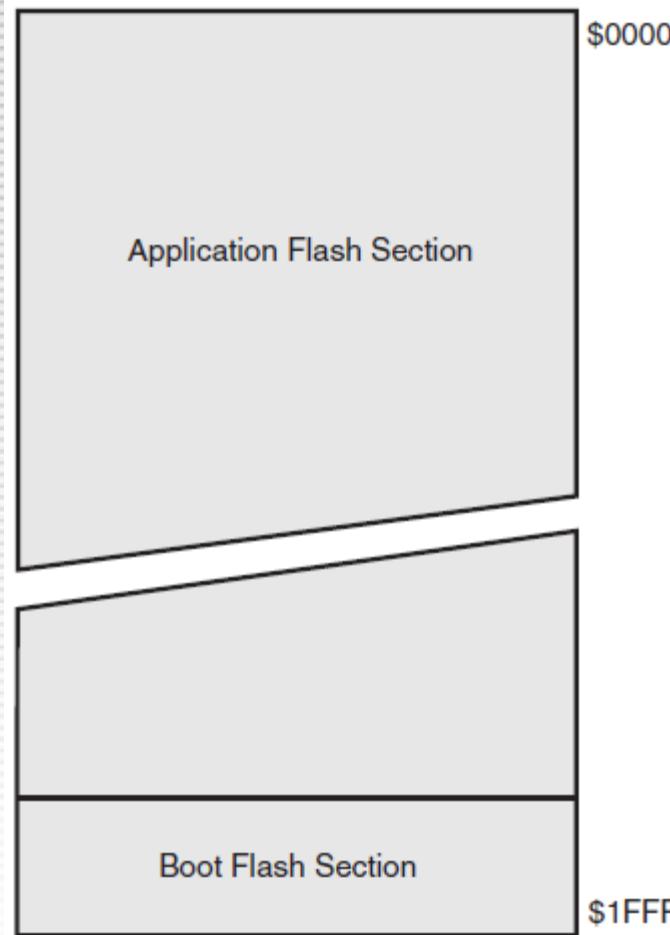
## بردارهای وقفه و باز نشانی

- هنگامی که فیوز **BOOTRST** برنامه ریزی شود، پس از بازنشانی میکروکنترلر به **Loader** می‌رود.

- هنگامی که **GICR** در **IVSEL** یک شود، بردارهای وقفه به شروع قسمت **boot** فلش منتقل می‌شوند. آدرس هر بردار وقفه در اینصورت برابر با آدرس موجود در جدول فوق بعلاوه آدرس شروع قسمت **boot** فلش خواهد بود.

- BOOTRST : Boot reset**

# وقفه ها



- بخش برنامه **Boot** در انتهای فضای حافظه برنامه و بعد از بخش برنامه کاربردی قرار می گیرد.

بلای تعیین اینکه آدرس قرارگیری رتفه‌ها در سمت کاربرد یا برت باشد بسته به نحو مناسب تنظیم شوند:  
 BootRST = 0 (یک نیزبیت است) ← یعنی اینکه وتفه در حافظه Boot تراکیز دارد  
 بسته IVSEL در ثبات GICR (که یکی از رجیستر های 64 گانه است)  
 IVSEL = 1 : یعنی اینکه در خواهیم سایر رتفه‌ها (رتفه‌هایی که وتفه Reset را دارند)

## وقفه‌ها

در صفت Boot تراکیز دارد

- اگر برنامه هیچ‌گاه یک منبع وقفه را فعال نکند، بردارهای وقفه عملاً استفاده نمی‌شوند و کد برنامه اصلی می‌تواند در این مکان‌ها قرار بگیرد.

- اگر هنگامی که بردارهای وقفه در بخش boot قرار دارند، بردار بازنشانی در بخش کاربرد قرار داشته باشد و بالعکس نیز به همین صورت می‌باشد.

Interrupt Vector Selector

BOOTRST	IVSEL	Reset address	Interrupt Vectors Start Address
1	0	\$0000	\$0002
1	1	\$0000	Boot Reset Address + \$0002
0	0	Boot Reset Address	\$0002
0	1	Boot Reset Address	<u>Boot Reset Address + \$0002</u>

آدرس شروع Boot به اضافه \$ 0002

فیوز BOOTRST برابر "1" به معنی برنامه‌ریزی نشده و "0" به معنی برنامه‌ریزی شده است. ← نیزبیت

\* نیزبیت‌ها در حالت پیش‌فرض برابر "1" است

# وقفه ها

- پیکربندی تخصیص فضای حافظه به بخش boot و بخش برنامه های کاربردی

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application section	Boot Reset Address (start Boot Loader Section)
1	1	128 words	2	\$0000 - \$1F7F	\$1F80 - \$1FFF	\$1F7F	\$1F80
1	0	256 words	4	\$0000 - \$1EFF	\$1F00 - \$1FFF	\$1EFF	\$1F00
0	1	512 words	8	\$0000 - \$1DFF	\$1E00 - \$1FFF	\$1DFF	\$1E00
0	0	1024 words	16	\$0000 - \$1BFF	\$1C00 - \$1FFF	\$1BFF	\$1C00

# وقفه ها

Address	Labels	Code	Comments
\$000		jmp RESET	; Reset Handler
\$002		jmp EXT_INT0	; IRQ0 Handler
\$004		jmp EXT_INT1	; IRQ1 Handler
\$006		jmp TIM2_COMP	; Timer2 Compare Handler
\$008		jmp TIM2_OVF	; Timer2 Overflow Handler
\$00A		jmp TIM1_CAPT	; Timer1 Capture Handler
\$00C		jmp TIM1_COMPA	; Timer1 CompareA Handler
\$00E		jmp TIM1_COMPB	; Timer1 CompareB Handler
\$010		jmp TIM1_OVF	; Timer1 Overflow Handler
\$012		jmp TIM0_OVF	; Timer0 Overflow Handler
\$014		jmp SPI_STC	; SPI Transfer Complete Handler
\$016		jmp USART_RXC	; USART RX Complete Handler
\$018		jmp USART_UDRE	; UDR Empty Handler
\$01A		jmp USART_TXC	; USART TX Complete Handler
\$01C		jmp ADC	; ADC Conversion Complete Handler
\$01E		jmp EE_RDY	; EEPROM Ready Handler
\$020		jmp ANA_COMP	; Analog Comparator Handler
\$022		jmp TWSI	; Two-wire Serial Interface Handler
\$024		jmp EXT_INT2	; IRQ2 Handler
\$026		jmp TIM0_COMP	; Timer0 Compare Handler
\$028		jmp SPM_RDY	; Store Program Memory Ready Handler
;			
\$02A	RESET:	ldi r16, high(RAMEND)	; Main program start
\$02B		out SPH, r16	; Set Stack Pointer to top of RAM
\$02C		ldi r16, low(RAMEND)	
\$02D		out SPL, r16	
\$02E		sei	; Enable interrupts
\$02F		<instr> xxx	
.....	.....	.....	.....

$BOOTRST = 1 \rightarrow App.$   
 $IVSEL = 0$

معمول ترین و عمومی ترین برنامه  
برای تنظیم آدرس بردارهای  
بازنشانی و وقفه در میکروکنترلر

ATmega16

# وقفه ها

- معمول ترین و عمومی ترین برنامه برای تنظیم آدرس بردارهای بازن Shanی و وقفه ، هنگامی که:

BOOTRST = 1

- فیوز BOOTRST برنامه ریزی نشده است

اندازه قسمت Boot روی ۲ کیلوبایت تنظیم شده باشد

بیت IVSEL در ثبات GICR قبل از این که هر وقفه ای فعال شود، یک شده باشد.

IVSEL = 1

روال بازن Shanی از بخش کاربرد (آدرس \$000) اجرا می شود.

• بردارهای سایر وقفه ها در بخش boot قرار دارند.

Address Labels Code Comments

\$000 RESET: ldi r16, high(RAMEND) ; Main program start

\$001 out SPH,r16 ; Set Stack Pointer to top of RAM

\$002 ldi r16, low (RAMEND)

\$003 out SPL, r16

\$004 sei ; Enable interrupts

\$005 <instr> xxx

;

.org \$1C02 → آدرس سمت Boot

\$1C02 jmp EXT\_INT0 ; IRQ0 Handler

\$1C04 jmp EXT\_INT1 ; IRQ1 Handler

... .. . . . ;

\$1C28 jmp SPM\_RDY ; Store Program Memory Ready Handler

سایر وقفه ها در نصت Boot

# وقفه ها

- معمول ترین و عمومی ترین برنامه برای تنظیم آدرس بردارهای بازن Shanی و وقفه، هنگامی که:  
 $\text{BOOTRST} = 0$       فیوز **BOOTRST** برنامه ریزی شده باشد  
 $\text{IVSEL} = 0$       اندازه قسمت Boot روی ۲ کیلوبایت تنظیم شده باشد  
بردارهای وقفه در بخش کاربرد قرار داشته باشند.
- روال بازن Shanی از شروع بخش حافظه boot (آدرس \$1C00) اجرا می شود.

Address	Labels	Code	Comments
.org \$002		\$002 jmp EXT_INT0 ; IRQ0 Handler	
		\$004 jmp EXT_INT1 ; IRQ1 Handler	
... .. . . . ;			
\$028		jmp SPM_RDY ; Store Program Memory Ready Handler	
;			
.org \$1C00			
\$1C00		RESET: ldi r16,high(RAMEND) ; Main program start	
		\$1C01 out SPH,r16 ; Set Stack Pointer to top of RAM	
		\$1C02 ldi r16,low(RAMEND)	
		\$1C03 out SPL,r16	
		\$1C04 sei ; Enable interrupts	
		\$1C05 <instr> xxx	

# وقفه ها

- معمول ترین و عمومی ترین برنامه برای تنظیم آدرس بردارهای بازن Shanی و وقفه، هنگامی که:  
 $\text{BOOTRST} = 0$ 
  - فیوز BOOTRST برنامه ریزی شده باشد
  - اندازه بخش Boot روی ۲ کیلوبایت تنظیم شده باشد
  - بیت IVSEL در ثبات GICR پیش از این که هر وقفه ای فعال باشد، یک شده باشد.  
 $\text{IVSEL} = 1$
  - بردار وقفه بازن Shanی و بردارهای سایر وقفه ها در بخش boot قرار دارند.
  - برنامه وقفه بازن Shanی در بخش boot قرار دارد.

Address	Labels	Code	Comments
		.org \$1C00	
\$1C00		jmp RESET ; Reset handler	
\$1C02		jmp EXT_INT0 ; IRQ0 Handler	
\$1C04		jmp EXT_INT1 ; IRQ1 Handler	
...		.... ;	
\$1C28		jmp SPM_RDY ; Store Program Memory Ready Handler	
		;	
\$1C2A	RESET:	ldi r16,high(RAMEND) ; Main program start	
\$1C2B		out SPH,r16 ; Set Stack Pointer to top of RAM	
\$1C2C		ldi r16,low(RAMEND)	
\$1C2D		out SPL,r16	
\$1C2E		sei ; Enable interrupts	
\$1C2F		<instr> xxx	

# ثبتات کنترل وقفه سراسری (GICR)

\* برای یک‌گردن IVSEL با یک‌تبلیغ IVCE را یک‌نیم (4 کلک و نیم داریم)

- ثبات کنترل وقفه سراسری:
- مکان جدول بردار وقفه را تعیین می‌کند.
- امکان فعال یا غیر فعال سازی وقفه‌های خارجی INT0، INT1 و INT2 را فراهم می‌سازد.
- در زمانی که IVCE یک رزودتا 4 کلک پالس همچ وقفه‌ای پیزیز نمود

Bit	7	6	5	4	3	2	1	0	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	

- GICR: General Interrupt Control Register

# ثبت کنترل وقفه سراسری (GICR)

## بیت ۱ - IVSEL: بیت انتخاب بردار وقفه

- وقتی که بیت IVSEL صفر شود، بردارهای وقفه در آغاز حافظه فلش قرار داده می‌شوند. هنگامی که این بیت یک شود، بردارهای وقفه به آغاز بخش Boot Loader فلش برده می‌شوند. آدرس واقعی شروع بخش Boot توسط فیوزهای BOOTSZ تعیین می‌شود.

## بیت ۰ - IVCE: بیت فعالسازی تغییر بردار وقفه

- بیت IVCE باید برای فراهم ساختن امکان تغییر در بیت IVSEL به حالت یک منطقی تنظیم شود

برای جلوگیری از ایجاد تغییرات ناخواسته در جدول بردار وقفه، یک روند ویژه نوشتن باید برای تغییر بیت IVSEL طی شود:

- ابتدا باید بیت فعالسازی تغییر بردار وقفه (IVCE) باید یک شود.
- مقدار مورد نظر را در IVSEL قرار دهیم. بیت IVCE بعد از ۴ چرخه ساعت از زمان نوشتن در آن به صورت سختافزاری صفر می‌شود.

# ثبتات کنترل وقفه سراسری (GICR)

- در حین اجری مراحل فوق، وقفه‌ها به طور خودکار غیرفعال می‌شوند.
- وقفه‌ها در چرخه‌ای که IVCE یک می‌شود، غیر فعال شده و تا دستورالعمل بعد از نوشته شدن IVSEL نیز غیرفعال باقی می‌مانند.
- اگر IVSEL نوشته نشود، وقفه‌ها به مدت ۴ چرخه ساعت غیرفعال باقی می‌مانند.
- بیت I موجود در ثبات وضعیت در طی این غیر فعال‌سازی خودکار، دست نخورده باقی می‌ماند.

# ثبتات کنترل وقفه سراسری (GICR)

Set Bit  
بهره برد از دستور  
استفاده نکرد

## Assembly Code Example

```
Move_interrupts:  
; Enable change of interrupt vectors  
    ldi r16, (1<<IVCE)  
    out GICR, r16  
; Move interrupts to boot Flash section  
    ldi r16, (1<<IVSEL)  
    out GICR, r16  
    ret
```

## C Code Example

```
void Move_interrupts(void)  
{  
/* Enable change of interrupt vectors */  
    GICR = (1<<IVCE);  
/* Move interrupts to boot Flash section */  
    GICR = (1<<IVSEL);  
}
```

- در این برنامه ابتدا IVCE و سپس IVSEL به کمک ثبات r16 و دستور out یک می‌شوند. با اینکار بردارهای وقفه به آغاز بخش Boot Loader فلش برده می‌شوند.

# وقفه ها

پیکر بندی تخصیص فضای حافظه به بخش boot و بخش کاربرد

	<u>میرزه است اند</u> BOOTSZ1	<u>میرزه است اند</u> BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application section	Boot Reset Address (start Boot Loader Section)
1	1	1	128 words	2	\$0000 - \$1F7F	\$1F80 - \$1FFF	\$1F7F	\$1F80
1	0	!! ۸۵ های word های ۸ بیت !! منظریه خانه های حافظه است	{ 256 words	4	\$0000 - \$1EFF	\$1F00 - \$1FFF	\$1EFF	\$1F00
0	1	1	512 words	8	\$0000 - \$1DFF	\$1E00 - \$1FFF	\$1DFF	\$1E00
0	0	0	1024 words	16	\$0000 - \$1BFF	\$1C00 - \$1FFF	\$1BFF	\$1C00

# ثبت کنترل وقفه سراسری (GICR)

توجه:

- بسته به مقدار موجود در شمارنده برنامه که ممکن است به آدرسی در حافظه برنامه بخش کاربرد یا در حافظه **boot** اشاره کند، وقفه‌ها با توجه به برنامه‌ریزی شده بودن یا نبودن بیت‌های **BLB12** یا **BLB02** ممکن است فعال یا غیر فعال شوند.

BLB02 = 0

Boot Loader Bit

- اگر بردارهای وقفه در بخش Boot Loader قرار داشته باشند و بیت BLB02 برنامه‌ریزی شده باشد، هنگامی که برنامه در قسمت کاربرد (application section) در حال اجرا باشد وقفه‌ها غیرفعال می‌شوند.

BLB12 = 0

- اگر بردارهای وقفه در بخش کاربرد قرار داشته و بیت BLB12 برنامه‌ریزی شده باشد، هنگامیکه برنامه در قسمت Boot Loader در حال اجرا باشد وقفه‌ها غیرفعال می‌شوند.



# وقفه‌های خارجی

- ۷ وقفه‌های خارجی توسط پایه‌های INT0، INT1 و INT2 تحریک می‌شوند.
- در صورتی که وقفه‌ها فعال باشند، حتی اگر پایه‌های INT0..2 در وضعیت خروجی پیکربندی شده باشند، وقفه‌ها تحریک خواهند شد. این ویژگی، راهی را برای تولید یک وقفه نرمافزاری فراهم می‌آورد.
- وقفه‌های خارجی می‌توانند بوسیله لبه پایین یا بالا رونده و یا با یک سطح ولتاژ پایین تحریک شوند. یک وقفه فعال شونده با لبه می‌باشد INT2.
- این تنظیم آنطور که در تعاریف MCUCR (ثبتات کنترل میکروکنترلر) و MCUCSR (ثبتات کنترل و وضعیت میکروکنترلر) آمده است، انجام می‌شود.

حالات دستگاه و خراfeld رخداد وقفه ایش را اعلام کند؛ مثلاً پایه INTO رو چهار کند؟

## (یادآوری از فصل I/O)

• بیت ۲ از درگاه D: این پایه ورودی وقفه خارجی شماره ۰ است.

• بیت ۳ از درگاه D: این پایه ورودی وقفه خارجی شماره ۱ است.

• بیت ۲ از درگاه B: این پایه ورودی وقفه خارجی شماره ۲ است.

برای نمایاندن دستگاه (تغییر) در جیفر (GIFR) و INTF0، INTF1، INTF2 را درست کنید و دفعه eg:  نوع وقفه یا:

برای نمایاندن دستگاه (تغییر) در جیکر (GICR) و INT0، INT1، INT2 را درست کنید و سیت فعال ساز و توقف eg:  سیت فعال ساز و توقف  ✓

I (بیت فعال ساز سراسر)   
که ترکهارسترن sei یک منزد

رجیسترهاي GICR و GIFR و جز 64 رجیستر I/O کند

# وقفه‌های خارجی

- هنگامی که یک وقفه خارجی تحریک شود و بصورت حساس به سطح (فقط در مورد INT0 و یا INT1) تنظیم شود، وقفه، وقتی که پایه در سطح پایین نگه داشته می‌شود، تحریک خواهد شد.
- تشخیص لبه بالا یا پایین رونده وقفه‌ها بر روی پایه‌های INT0 و INT1 مستلزم وجود  $\text{Clk}_{\text{i/o}}$  است که در بخش سیستم‌های ساعت و توزیع آن‌ها توضیح داده شد.
- وقفه‌های حساس به سطح پایین بر روی INT0 و یا INT1 و وقفه حساس به لبه بر روی INT2 بصورت ناهمگام دریافت می‌شوند.
- این بدین معنی است که این وقفه‌ها می‌توانند برای بیدار کردن قطعه از حالات خواب (sleep) به جز حالت بیکار، بکار روند.
- ساعت  $\text{Clk}_{\text{i/o}}$  در تمامی حالت‌های خواب به جز حالت بیکار، متوقف (halt) می‌شود.

# وقفه‌های خارجی

- اگر یک وقفه حساس به سطح برای بیدار شدن از حالت خاموش (power down) بکار رود، سطح ولتاژ تغییر یافته باید به مدت معینی برای بیدار کردن میکروکنترلر حفظ شود. این باعث می‌شود که میکروکنترلر کمتر نسبت به نویز حساس باشد.
- در صورتی که ورودی وقفه در طول این نمونهبرداری، سطح مورد نظر را داشته باشد و یا اینکه تا پایان زمان start-up سطح ولتاژ نگه داشته شود، میکروکنترلر بیدار خواهد شد.
- زمان start-up توسط فیوزهای SUT که در بخش سیستم ساعت سیستم و تنظیمات آن آمده است، تعریف شد.

# نوسان‌ساز کریستالی (Clock system (یادآوری از فصل

در حالت انتخاب ساعت power down و power save بعد از حالات صرفه‌جویی در توان (مثل حالت start-up زمان‌های نوسان‌ساز کریستالی

CKSEL0	SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ( $V_{CC} = 5.0V$ )	Recommended Usage
0	00	258 CK <sup>(1)</sup>	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK <sup>(1)</sup>	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK <sup>(2)</sup>	—	Ceramic resonator, BOD enabled
0	11	1K CK <sup>(2)</sup>	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK <sup>(2)</sup>	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	—	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly rising power

# ✓ ثبات کنترلی MCUCR : MCU

ثبتات کنترلی میکروکنترلر شامل بیت‌هایی برای کنترل نحوه حس وقفه و همچنین عملکردهای عمومی میکروکنترلر می‌باشد.

Bit	7	6	5	4	3	2	1	0	MCUCR
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

**MCUCR: MCU Control Register**

## وقفه خارجی ۱

- لبه‌ها و سطحی که در صورت مشاهده آن‌ها برروی پایه خارجی INT1، وقفه فعال می‌شود، در جدول اسلاید بعد تعریف شده است.
- مقدار روی پایه INT1 نمونه‌برداری می‌شود. در صورتی که وقفه حساس به لبه و یا وقفه تغییر حالت انتخاب شود، پالس‌هایی که بیش از یک پریود پالس ساعت بطول می‌انجامند، یک وقفه تولید خواهند کرد. پالس‌های کوتاه‌تر تضمینی نیست که یک وقفه ایجاد نمایند.
- در صورتی که وقفه حساس به سطح پایین انتخاب شود، برای اینکه وقفه‌ای تولید شود، سطح پایین باید تا زمان اتمام دستورالعمل در حال اجرا، حفظ شود.

✓ بیت‌های ۲ و ۳ (ICS11 و ICS10) بیت‌های کنترل نحوه حس وقفه در پایه وقفه خارجی ۱

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.

## وقفه خارجی \*

- در صورتی که بیت وقفه سراسری | در SREG و بیت پوشش وقفه شماره . در ثبات GICR یک شوند، وقفه خارجی . می‌تواند توسط پایه خارجی INTO فعال شود.
- لبه‌ها و سطحی که در صورت مشاهده آن‌ها برروی پایه خارجی INTO، وقفه فعال می‌شود، در جدول اسلاید بعد تعریف شده است.
- مقدار روی پایه INTO نمونه‌برداری می‌شود. در صورتی که وقفه حساس به لبه و یا وقفه تغییر حالت انتخاب شود، پالس‌هایی که بیش از یک پریود پالس ساعت بطول می‌انجامند، یک وقفه تولید خواهند کرد.
- پالس‌های کوتاه‌تر تضمینی نیست که یک وقفه ایجاد نمایند. در صورتی که وقفه حساس به سطح پایین انتخاب شود، برای اینکه وقفه‌ای تولید شود، سطح پایین باید تا زمان اتمام دستورالعمل در حال اجرا، حفظ شود.



## بیت‌های ۰ و ۱ (ISC01 و ISC00)

بیت‌های کنترل نحوه حس وقفه در پایه وقفه خارجی

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.

## بیت ششم (ISC2): بیت کنترل نحوه حس وقفه در پایه وقفه خارجی ۲

- در صورتی که بیت وقفه سراسری I در SREG و بیت پوشش وقفه شماره ۲ در GICR یک شوند، وقفه خارجی و ناهمگام ۲ می‌تواند توسط پایه خارجی INT2 فعال شود.
- در صورتی که مقدار صفر در ISC2 نوشته شود، یک لبه پایین رونده بر پایه INT2 وقفه را فعال خواهد کرد و در صورتی که مقدار یک در ISC2 نوشته شود، یک لبه بالا رونده بر پایه INT2 وقفه را فعال خواهد کرد.
- لبه‌ها بر روی پایه INT2 بصورت ناهمگام ثبت می‌شوند.
- در حین تغییر بیت ISC2، وقفه می‌تواند به وقوع بپیوندد. بنابراین، توصیه می‌شود که INT2 ابتدا به وسیله صفر کردن بیت فعال‌ساز مربوط به آن در GICR، غیر فعال شود و سپس بیت ISC2 تغییر داده شود.

# ثبتات کنترلی و وضعیت میکروکنترلر: MCUCSR

Bit	7	6	5	4	3	2	1	0	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

## MCUCSR : MCU Control and Status Register

# حداقل عرض پالس برای وقفه خارجی ناهمگام



Symbol	Parameter	Condition	Min	Typ	Max	Units
$t_{INT}$	Minimum pulse width for asynchronous external interrupt			50		ns

INT2 هم نیاز به  $Clk_{IO}$  ندارد (ب علت پیاره سازی) سفت از ندای

آخر INT ها حساس به لبه باشند

تمها حالت خواب که در ترکیز بسیار زدن از آن استفاده کرد حالت Idle است جو  $Clk_{IO}$  فعال است  
و لی اگر  $INT_0$ ,  $INT_1$ ,  $INT_2$  حساس ب طبع بایسین باشد در ترکیز از حالت های خواب دیگر هم استفاده کرد جو در این حالت آنکه از است  
له حالا وقتی باید در طبع بایسین تکمیل کردم ب اندیشه time-out startup time from Power down to Power up

# ثبتات کنترل وقفه سراسری یا GICR

Bit	7	6	5	4	3	2	1	0	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	

**GICR: General Interrupt Control Register**

## بیت هفتم (INT1): بیت فعال‌ساز درخواست وقفه خارجی ۱

- هنگامی که بیت INT1 و همچنین بیت وقفه سراسری I از ثبات وضعیت (ثبتات SREG) یک شوند، وقفه پایه خارجی INT1 فعال می‌شود.
- بیت‌های کنترلی ISC11 و MCUCR در ثبات ISC10 تعیین می‌کنند که وقفه خارجی در لبه‌های پایه INT1 (لبه بالارونده و یا پایین‌رونده) تشخیص داده شود یا در سطح.
- فعال کردن این پایه، گرچه INT1 در وضعیت خروجی تنظیم شده باشد، موجب درخواست یک وقفه می‌شود. وقفه متناظر با درخواست وقفه خارجی ۱، از بردار وقفه INT1 اجرا می‌شود.

## بیت ششم (INT0): بیت فعال‌ساز درخواست وقفه خارجی ۰

- هنگامی که بیت INT0 و همچنین بیت وقفه سراسری I از ثبات وضعیت (ثبتات یک باشند، وقفه پایه خارجی INT0 فعال می‌شود.
- بیت‌های کنترلی ISC01 و ISC00 در ثبات MCUCR، اینکه وقفه خارجی در لبه‌های پایه INT0 (لبه بالارونده و یا پایین‌رونده) و یا در سطح تشخیص داده شود را تعیین می‌کنند.
- فعالیت بر روی پایه، گرچه INT0 در وضعیت خروجی تنظیم شده باشد، موجب درخواست یک وقفه می‌شود. وقفه متناظر با درخواست وقفه خارجی ۰، از بردار وقفه INT0 اجرا می‌شود.

## بیت پنجم (INT2): بیت فعال‌ساز درخواست وقفه خارجی ۲

- هنگامی که بیت INT2 و همچنین بیت وقفه سراسری I از ثبات وضعیت (ثبتات INT2) یک باشند، وقفه پایه خارجی INT2 فعال می‌شود.
- بیت کنترلی ISC2 در ثبات MCUCR، اینکه وقفه خارجی در لبه‌های پایه INT2 (لبه بالارونده و یا پایین‌رونده) و یا در سطح تشخیص داده شود را تعیین می‌کنند.
- فعالیت بر روی پایه، گرچه INT2 در وضعیت خروجی تنظیم شده باشد، موجب درخواست یک وقفه می‌شود. وقفه متناظر با درخواست وقفه خارجی ۲، از بردار وقفه INT2 اجرا می‌شود.

# ثبتات عمومی پرچم‌های وقفه : GIFR

Bit	7	6	5	4	3	2	1	0	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

**GIFR: General Interrupt Flag Register**

## بیت هفتم (INTF1): پرچم وقفه خارجی ۱

- هنگامی که یک لبه و یا تغییر سطح بر روی پایه INT1، یک درخواست وقفه را تحریک می‌کند، پرچم INTF1 یک می‌شود.
- در صورتی که بیت I از ثبات SREG و بیت INT1 از GICR یک باشند، میکروکنترلر به بردار وقفه متناظر پرسخواهد کرد.
- این پرچم، وقتی که روال وقفه اجرا شود، صفر می‌گردد و یا اینکه می‌توان با نوشتن یک منطقی در آن، آن را صفر نمود.

## بیت ششم (INTF0): پرچم وقفه خارجی صفر

- هنگامی که یک لبه و یا تغییر سطح بر روی پایه INT0، یک درخواست وقفه را تحریک می‌کند، پرچم INT0 یک می‌شود.
- در صورتی که بیت ۱ از ثبات SREG و بیت ۰ از INT0 از GICR یک باشند، میکروکنترلر به بردار وقفه متناظر پرش خواهد کرد.
- این پرچم، وقتی که روال وقفه اجرا شود، صفر می‌گردد و یا اینکه می‌توان با نوشتن یک منطقی در آن، آن را صفر نمود.

## بیت پنجم (INTF2): پرچم وقفه خارجی ۲

- هنگامی که یک رویداد بر روی پایه INT2، یک درخواست وقفه را تحریک می‌کند، پرچم INT2 یک می‌شود.
- در صورتی که بیت وقفه سراسری I از ثبات SREG و بیت INT2 از GICR یک باشند، میکروکنترلر به بردار وقفه متناظر پرش خواهد کرد. این پرچم، وقتی که روال وقفه اجرا شود، صفر می‌گردد، و یا اینکه می‌توان با نوشتمن یک منطقی در آن، آن را صفر کرد.