وقفه ها (Interrupts)

وقفه چیست ؟

وقفه یک قابلیت می باشد که با فعال شدن آن میکرو کار عادی خود را رها کرده و یک کار ویژه را که برای میکرو تعریف شده است را انجام می دهد و پس از اجرا ، دوباره روال عادی کارهای خود را از سر می گیرد . وقفه ها دو نوع می باشند : داخلی و خارجی . وقفه های داخلی مانند وقفه ارتباط سریال ، وقفه مقایسه کننده آنالوگ و... می باشد . وقفه های خارجی همانگونه که از نامشان پیداست هنگامی اجرا می شوند که میکرو از بیرون تحریک شود . این تحریک می تواند به صورت لبه بالا رونده ، پایین رونده و یا هر تغییری باشد که باید برای میکرو این نحوه تحریک به صورتی که در ادامه خواهیم گفت تعریف شود .

وقفه های خارجی

خانواده MEGA16 دارای سه وقفه خارجی می باشد که شامل پایه های 3 ، 4 و 16 می باشد که که برای فعال کردن و یا غیر فعال کردن و انتخاب نوع تحریک آنها می توان از CodeWizard استفاده کرد . در این تنظیمات رجیسترهای زیر درگیر می شوند .

1- GICR (Global Interrupt Control Register) :

این رجیستر جهت فعال کردن وقفه می باشد برای این کار از سه بیت با ارزش آن استفاده می کنیم . برای فعال کردن وقفه صفر از دستور GICR|=0X40 و برای وقفه یک از GICR|=0X80 و برای وقفه دو از دستور GICR|=0X20 استفاده می کنیم .

7	6	5	4	3	2	1	0	
INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
R/W	R/W	R/W	R	R	R	R/W	R/W	

INT1 (External Interrupt Request Enable) :

فعال ساز وقفه خارجی یک می باشد .

INT0 (External Interrupt Request Enable):

فعال ساز وقفه خارجی صفر می باشد .

INT2 (External Interrupt Request Enable):

فعال ساز وقفه خارجی دو می باشد .

2-رجیستر MCUCR (MCU Control Register) :

توسط چهار بیت کم ارزش این رجیستر می توان نحوه تحریک وقفه های خارجی یک و صفر را تعیین کرد . بیت های این رجیستر به صورت زیر می باشد .

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	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	

توسط دوبیت ISC00 و ISC01 می توان نحوه تحریک وقفه خارجی صفر را تعین نمود .

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.

جدول 6-1

و توسط بیت های ISC10 وISC11 می توان نحوه تحریک وقفه خارجی یک را مشخص نمود .

Table 34. Interrupt 1 Sense 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.

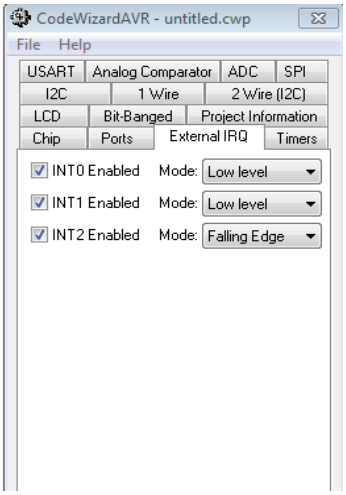
جدول 6-2

3-رجیستر MCUCSR (Control and Status Register)

توسط بیت ISC2 این رجیستر می توان نحوه تحریک وقفه خارجی دو را تعیین کرد به طوریکه اگر این بیت صفر باشد در لبه پایین رونده و اگر یک باشد در لبه بالارونده وقفه رخ خواهد داد . بیت های این رجیستر به صورت زیر می باشد :

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					

تنظیمات CodeWizard



در منوی CodeWizard می توان در برگه External RQ وقفه های خارجی یک , دو و سه را فعال کرد . همچنین می توان در نحوه تحریک وقفه را نیز انتخاب کرد . بدین گونه که در حالت Rising Edge : تحریک با لبه بالا رونده خواهد بود .

Faling Edge : تحریک وقفه را با لبه بالا رونده کرده

Any change : باعث تحریک وقفه با هر تغییری می شود . شکل1-6

پس از تنظیمات مورد نظر در صفحه برنامه شما پس از معرفی کتابخانه ها جمله زیر نمایان خواهد شد که شما باید برنامه ای را که در هنگام وقفه می خواهید انجام شود را در آن بنویسید :

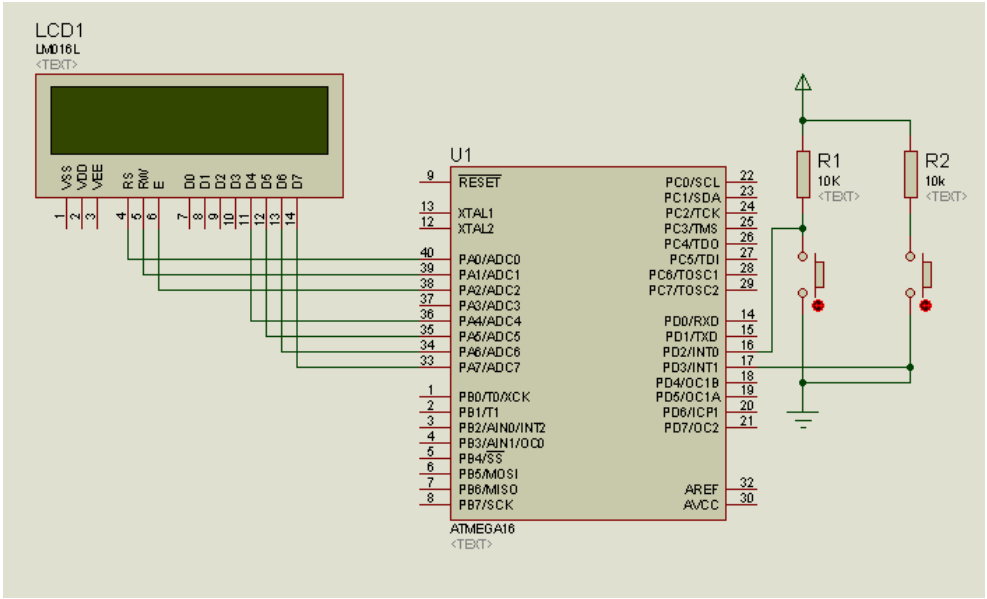
```
interrupt [EXT_INTn] void ext_intn_isr(void)
```

که در آن n شماره وقفه خارجی انتخاب شده خواهد بود .

مثال : می خواهیم برنامه ای بنویسیم که وقفه صفر در لبه بالا رونده و وقفه یک در لبه پایین رونده اسم ما را بر روی نمایشگر نشان دهد .

```
#include <mega16.h>                    //معرفی میکرو مورد استفاده
#include <delay.h>                     //فراخوانی کتابخانه تاخیر زمانی
#asm                                   //شروع برنامه اسمبلی
.equ __lcd_port=0x1B ;PORTA            // معرفی پورتA به عنوان پورت نمایشگر
#endasm                                //پایان برنامه اسمبلی
```

```
#include <lcd.h>                                        //فراخوانی کتابخانهlcd
interrupt [EXT_INT0] void ext_int0_isr(void) {          // تابع وقفه خارجی صفر
lcd_clear();                                            // پاک کردن نمایشگر
lcd_gotoxy(5,0);                                        //رفتن به سطر و ستون مورد نظر
lcd_putsf("zareia");                                    //نوشتن کاراکتر مورد نظر
delay_ms(1500); }                                       //تاخیر زمانی
interrupt [EXT_INT1] void ext_int1_isr(void) {          //تابع وقفه خارجی یک
lcd_clear();                                            //پاک کردن نمایشگر
lcd_gotoxy(6,0);                                        //رفتن به سطر و ستون مورد نظر
lcd_putsf("jowzi");                                     //نوشتن کاراکتر مورد نظر
delay_ms(1500);  }                                      //تاخیر زمانی
void main(void)  {                                      // برنامه اصلی
while (1)  {                                            //حلقه بینهایت
lcd_clear();                                            //تاخیر زمانی
lcd_gotoxy(7,0);                                        //رفتن به سطر و ستون مورد نظر
lcd_putsf("helo");                                      //نوشتن کاراکتر مورد نظر
lcd_gotoxy(0,1);                                        //رفتن به سطر و ستون مورد نظر
lcd_putsf(" choice 1 or 2");                            //نوشتن کاراکتر مورد نظر
delay_ms(1000) } }                                      //تاخیر زمانی
```


LCD1
LM016L
<TEXT>
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
ATMEGA16
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
R1
10K
R2
10k

شکل2-6