

السلام عليكم ورحمة الله وبركاته

تتمة لدرس المقاطعات .. لا بد من الحديث ولو قليلاً .. عن مسجلات المتحكم الخاصة بالمقاطعات ..

والتي عن طريق تغير قيمها تتغير استجابة المتحكم للمقاطعة .. أو يتم الغاؤها نهائياً ..

إذا تذكرنا هذه الأسطر التي رأيناها في الكود .. في الدرس السابق :

```
119
120 // External Interrupt(s) initialization
121 // INT0: On
122 // INT0 Mode: Rising Edge
123 // INT1: On
124 // INT1 Mode: Low level
125 // INT2: On
126 // INT2 Mode: Falling Edge
127 GICR|=0xE0;
128 MCUCR=0x03;
129 MCUCSR=0x00;
130 GIFR=0xE0;
```

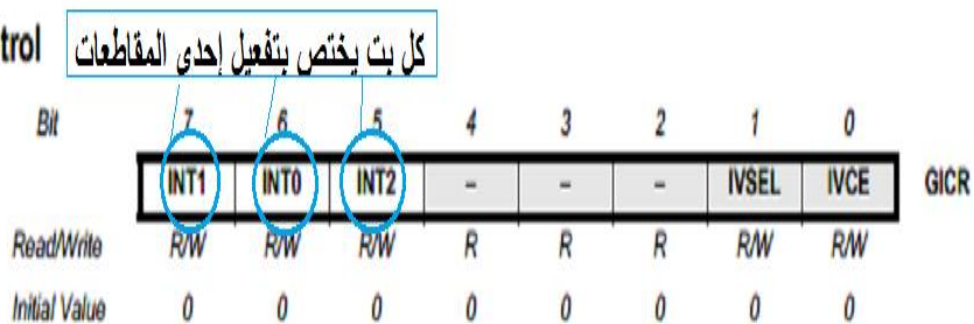
المتحولات GICR و MCUCR و MCUCSR و GIFR هي المسجلات الخاصة بالمقاطعات ...

وكل مسجل منها طوله ٨ بت .. لكل بت منها تقريباً معنى محدد .. تبينها ال datasheet الخاصة بالمتحكم :

نبدأ بالمسجل الأول GICR : General Interrupt Control Register

(صفحة ٦٩ External Interrupts)

General Interrupt Control Register – GICR



ملاحظة هامة : في ال datasheet وعند شرح محتويات أحد المسجلات .. تكون البتات التي تهتمنا في هذه الفقرة لونها أبيض ... وتكون البتات الغير هامة .. (التي تتبع لفقرات أخرى) لونها رمادي .. كي ينحصر التركيز على الهام .

لاحظ كيف أن البت ٥ .. إذا كانت قيمته ٠ .. فهذا يعني أن المقاطعة INT2 غير مفعلة .. وإن كان ١ .. فهي مفعلة

وكذلك البت ٦ يتحكم بالمقاطعة INT0 .. والبت ٧ بالمقاطعة INT1 ...

ولأن البتات الخمسة الأولى لا تهتمنا .. فيجب عند التحكم بالمقاطعات ألا نستخدم عملية الإسناد = بل نستخدم عملية المنطقية | وذلك حتى لا نسبب تغيير بتات أخرى في المسجل ..

مثلاً كان المسجل به القيمة : ٠٠١٠٠٠١١ وأردنا تفعيل المقاطعة INTO .. فلو كتبنا GICR=01000000 فسنغير حالة المقاطعة INT2 دون قصد ... كما سنغير قيم البت رقم ٠ والبت رقم ١ .. وهذا خاطئ ...

لذلك نستخدم العملية | كما يلي : GICR|=01000000 وهذا يكافئ : GICR=GICR|01000000 وبذلك لن نغير البتات السابقة ..

والآن لنتابع في الكود ... الذي كتبناه .. حيث قمنا بتفعيل المقاطعات الثلاث ..

أي نريد عمل | للقيمة ١١١٠٠٠٠٠ والتي تكافئ بالعد الست عشري . E0 .. انظر الكود :

```

119
120 // External Interrupt(s) initialization
121 // INT0: On
122 // INT0 Mode: Rising Edge
123 // INT1: On
124 // INT1 Mode: Low level
125 // INT2: On
126 // INT2 Mode: Falling Edge
127 GICR|=0xE0; GICR=GICR|1110000;
128 MCUCR=0x03;
129 MCUCSR=0x00;
130 GIFR=0xE0;

```

هذه وظيفة مسجل التحكم بالمقاطعات العام .. GICR ..

والآن إلى المسجل الثاني MCUCR: MCU control Register

الاختصار MCU يعني Micro Controller Unit (والله أعلم)

هذا هو بيان المسجل (صفحة ٦٨)

MCU Control Register – MCUCR The MCU Control Register contains control bits for interrupt sense control and general MCU functions.

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	

يهتمنا منه البتات الأربع الأولى .. كما نرى من الصورة .. وبهذا البتات نتحكم بكيفية استجابة المقاطعتين INTO,INT1

فقد رأينا أن لهما أربع أنماط للاستجابة .. (جبهة صاعدة أو هابطة ..أو المستوى المنخفض .. أو أي تغيير)

أربع احتمالات .. لكل مقاطعة .. إذا نحتاج ٢ بت .. لتمثيل حالة الاستجابة للمقاطعة الواحدة ..

وهنا لدينا ISC01 و ISC00 نستخدمها لمثل أي نمط استجابة المقاطعة INT0 كما يلي :

Table 35. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request. المستوى المنخفض
0	1	Any logical change on INT0 generates an interrupt request. أي تغيير
1	0	The falling edge of INT0 generates an interrupt request. جبهة هابطة
1	1	The rising edge of INT0 generates an interrupt request. جبهة صاعدة

فمثلاً لو أردنا جعل المقاطعة INT0 فعالة عند الجبهة الصاعدة نكتب: MCUCR=0b00000011; وبلدينا ISC10 و ISC11 لتمثيل كيفية استجابة المقاطعة INT1 مثل سابقتها :

Table 34. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT0 generates an interrupt request. المستوى المنخفض
0	1	Any logical change on INT0 generates an interrupt request. أي تغيير
1	0	The falling edge of INT0 generates an interrupt request. جبهة هابطة
1	1	The rising edge of INT0 generates an interrupt request. جبهة صاعدة

بقي لنا التحكم بكيفية استجابة المقاطعة INT2 ولكن جميع البتات في المسجل MCUCR محجوزة كما نرى هنا :

MCU Control Register – MCUCR The MCU Control Register contains control bits for interrupt sense control and general MCU functions.

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	

ولذلك ترك التحكم بالمقاطعة INT2 للمسجل التالي MCUCSR: MCU Control and Status Register

لنتذكر أن INT2 كان لديها احتمالات للاستجابة فقط .. جبهة صاعدة و جبهة هابطة .. احتمالات .. يحتاجان بت واحد

MCU Control and Status Register – MCUCSR

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

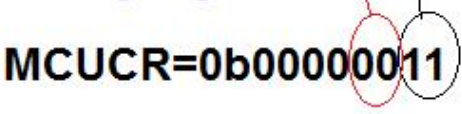
البت رقم ٦ من المسجل .. إن كان يحوي ٠ فالمقاطعة تستجيب للجبهة الهابطة ، وإن كان يحوي ١ فللجبهة الصاعدة

طبعا عند الحديث عن كيفية استجابة المقاطعة للحدث .. فهذا يعني أننا قمنا بتفعيلها سابقاً ..

والآن لنتابع ما كتبناه في الكود :

بالنسبة للمقاطعتين INT0,INT1

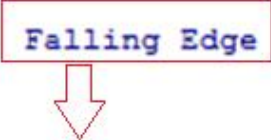
```
119
120 // External Interrupt(s) initialization
121 // INT0: On
122 // INT0 Mode: Rising Edge
123 // INT1: On
124 // INT1 Mode: Low level
125 // INT2: On
126 // INT2 Mode: Falling Edge
127 GICR|=0xE0;
128 MCUCR=0x03;
129 MCUCSR=0x00;
130 GIFR=0xE0;
```



MCUCR=0b00000011

أما المقاطعة INT2 :

```
119
120 // External Interrupt(s) initialization
121 // INT0: On
122 // INT0 Mode: Rising Edge
123 // INT1: On
124 // INT1 Mode: Low level
125 // INT2: On
126 // INT2 Mode: Falling Edge
127 GICR|=0xE0;
128 MCUCR=0x03;
129 MCUCSR=0x00;
130 GIFR=0xE0;
```



0b00000000

بقي لنا المسجل الأخير GICR: General Interrupts Flag Register

General Interrupt Flag Register – GICR

Bit	7	6	5	4	3	2	1	0	
	INTF1	INTF0	INTF2	–	–	–	–	–	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

توزيع البتات فيه مشابه للمسجل GICR ... ولكن GICR كان يتحكم بتفعيل أو عدم تفعيل المقاطعة ...

أما هذا المسجل فهو مسجل أعلام .. (وليس تحكم) أي أن قيمته تابعة تعبر عن حدوث حدث ما يخص مقاطعة ما ..

فبعد حدوث تغير في مستوى الجهد على المقاطعة INT2 مثلاً .. فإن البت رقم ٥ يحتفظ بالقيمة ١ إلى حين تنفيذ المقاطعة .. (يقوم بذلك سواء كانت المقاطعة مفعلة أم لا) .. وعند تغير هذه القيمة يتم التحقق مما يلي :

١ - هل المقاطعات مفعلة أم لا .. (بشكل عام)

٢ - هل المقاطعة التي تغيرت القيمة الخاصة بها مفعلة أم لا ..

٣ - هل نمط استجابة هذه المقاطعة يتوافق مع التغير الحاصل ..

إذا كانت الشروط السابقة متوفرة .. يتم تنفيذ المقاطعة .. وإلا يتم تجاهل هذا التغير ويعود البت لوضع ٠ .

إذا يمكننا اعتبار هذا المسجل المسؤول الرئيسي عن استدعاء المقاطعات ..

ذكرنا في بداية الدرس السابق أن المقاطعات تُستدعى خارجياً ... ولكن رغم ذلك .. يمكننا استدعاؤها داخلياً ..

وذلك بوضع القيمة ١ في بت المقاطعة التي نريد استدعاءها ...

قد يتبادر للذهن سؤال .. لماذا قام البرنامج بتفعيل المقاطعات هنا :

```
119
120 // External Interrupt(s) initialization
121 // INT0: On
122 // INT0 Mode: Rising Edge
123 // INT1: On
124 // INT1 Mode: Low level
125 // INT2: On
126 // INT2 Mode: Falling Edge
127 GICR|=0xE0;
128 MCUCR=0x03;
129 MCUCSR=0x00;
130 GICR=0xE0; ??
```

بدون سبب !! وعلى أي حال .. الأسطر الأربعة المكتوبة لن تعمل .. بدون السطر الذي جاء بعدها :

والمسؤول عن تفعيل المقاطعات بشكل رئيسي

وستحدث عنه الآن ..

```
141 // Global enable interrupts
142 #asm("sei")
```

إضافة إلى السطور السابقة كان هناك السطر :

```
141 // Global enable interrupts
142 #asm("sei")
```

والمسؤول عن تفعيل المقاطعات بشكل عام ...

يمكننا الاستعاضة عن تعليمة الأسمبلي السابقة .. بعد فهمنا لعملها :

هناك مسجل هام جداً في المتحكم ... مسجل أعلام الحالة : SREG : Status Register

The AVR Status Register – **SREG** – is defined as:

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	

آخر بت فيه (البت رقم ٧) .. مسؤول عن تفعيل أو عدم تفعيل المقاطعات ..

فإن كانت قيمته ١ .. فالمتحكم يستجيب للمقاطعات المفعلة ..

وإن كانت قيمته ٠ .. فحتى لو قمنا بتفعيل المقاطعات الثلاث من المسجل GICR فلن يستجيب لها المتحكم

إذا: يمكننا الاستعاضة عن السطر السابق بكتابة : SREG|=0b10000000;

وبالنسبة للسؤال الذي طرحناه قبل قليل :

```
119
120 // External Interrupt(s) initialization
121 // INT0: On
122 // INT0 Mode: Rising Edge
123 // INT1: On
124 // INT1 Mode: Low level
125 // INT2: On
126 // INT2 Mode: Falling Edge
127 GICR|=0xE0;
128 MCUCR=0x03;
129 MCUCSR=0x00;
130 GIFR=0xE0; ??
```

فكما رأينا أن هذه القيمة سرعان ما تعود إلى ٠ لأن المقاطعات غير مفعلة بعد .. ولن يكون لها أي تأثير ...

يمكنكم تجربة حذف السطر السابق ..

وتغيير هذا السطر

```
141 // Global enable interrupts
142 #asm("sei")
```

إلى SREG|=0b10000000; للتأكد من أنه سيعمل بدون أخطاء ..

إلى هنا يصل ملحق درس المقاطعات إلى نهايته .. أرجو أن يكون فيه الفائدة .. والله ولي التوفيق