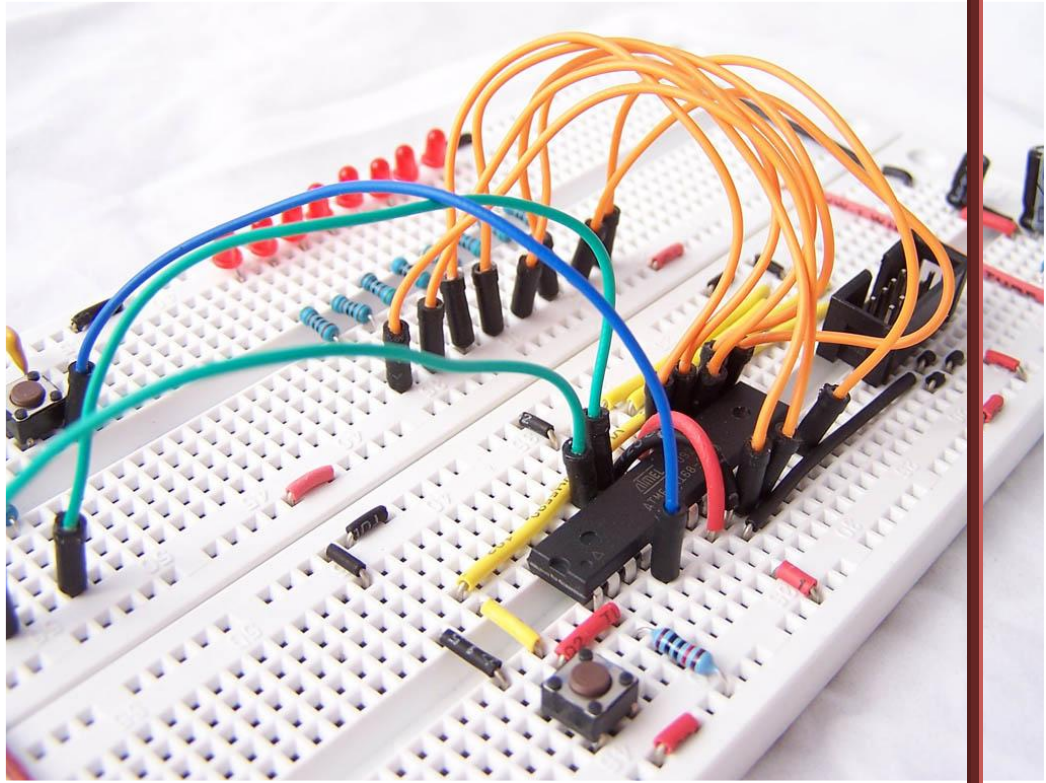


2012

المقاطع الخارجية



36a2 مصط فى

1/12/2012

الحمد لله رب العالمين والصلاة والسلام على سيدنا محمد صاحب الخلق العظيم وعلى آله الطيبين الطاهرين
السلام عليكم ورحمة الله وبركاته ..

موضوع درسنا لهذا اليوم :

المقاطعات الخارجية.. External Interrupts :

هي عدة تعليمات يتم تنفيذها بناء على أمر خارجي ...ولها أولوية أعلى من البرنامج الرئيسي ...
نقوم بتغليف هذه التعليمات ضمن دالة **function** ولكن لا يمكننا استدعاء هذه الدالة ..من البرنامج!!
لأنها تستدعي بأمر خارجي ...

أي أنها دالة تُستدعى ...**(هاردويرياً)** أي أن تغييراً خارجياً هو الذي يسبب استدعاءها وتنفيذها
الفكرة الأساسية في موضوع المقاطعات هي أولوية التنفيذ ...

فقد يكون لدينا عدة مقاطعات ... فيكون لكل مقاطعة ترتيب محدد في سلم أولويات المتحكم
يتوقف تنفيذ البرنامج الأصلي .. ويتحول المتحكم إلى تنفيذ البرنامج الموجود في دالة المقاطعة ...
بعض الأمثلة الواقعية لفهم مبدأ المقاطعات :

لكي نلاحظ المقاطعات في حياتنا لنبحث عن أمور تحدث دون إرادتنا ..فتجعلنا نتوقف عما نقوم به ونقوم بعمل آخر..
مثلاً ... حادث وفاة ... أو مرض يلزم الفراش .. هي مقاطعات خارجة عن إرادتنا ...تضطرنا للتوقف عما نقوم به
والبدء بتنفيذ برنامج حادث الوفاة والذي يتضمن عدة تعليمات مثل تغسيل الميت وتجهيز الكفن والقبر ..إلخ
أو الراحة مثلاً وأخذ الدواء في حال المرض ..
بالمثل ... فهناك في الحاسوب الشخصي زر لإعادة التشغيل ... يجعل الحاسوب يتوقف عن عمله الحالي ويعيد التشغيل.
وفي المتحكم هناك أرجل خاصة ... عند تطبيق جهد معين عليها تقوم بتنفيذ الدالة الخاصة بهذه المقاطعة ...
ومن خلال البرنامج نحن نتحكم بما يلي :

1-هل هناك مقاطعات أم لا ..

2-ما هي المقاطعات التي نريدها أن تعمل ...

3-متى يستجيب المتحكم للمقاطعة ؟ (عند النبضة الصاعدة أو الهابطة أو ثبات الجهد أو أي تغير)

4-ما هو الكود الذي يجب تنفيذه عند الاستجابة للمقاطعة ؟

بالنسبة للسؤال الثالث فسنحدث عنه قليلاً ...

نعلم أن النظام الثنائي يحتوي رقمين . **صفر وواحد** .. ولكن الاحتمالات من أجل رقمين متتاليين هي 4 كما يلي :

صفر ثم صفر ..

صفر ثم واحد ..

واحد ثم صفر ..

واحد ثم واحد ..

فإذا اعتبرنا أن الواحد هو 5 volt وأن الصفر هو 0 فولت ..

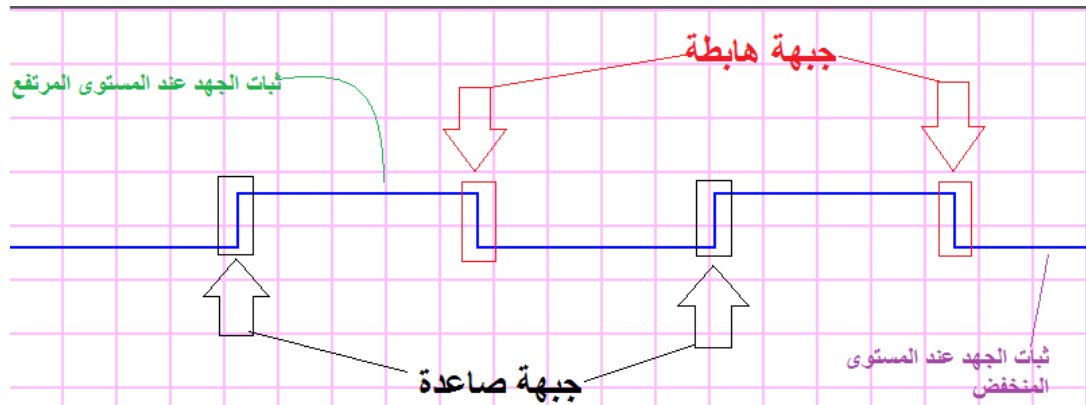
فسنسمي كل احتمال من الاحتمالات الاربعة باسم مختلف كما يلي :

صفر ثم صفر .. (أي ثبات الجهد عن الصفر ... نسميه المستوى المنخفض (Low Level))

صفر ثم واحد .. (أي انتقال الجهد منخفض إلى مرتفع .. نسميه جبهة صاعدة (Rising Edge))

واحد ثم صفر .. (أي انتقال الجهد من المرتفع إلى المنخفض ونسميه جبهة هابطة (Falling Edge))

واحد ثم واحد .. (أي ثبات الجهد عند المستوى المرتفع .. (High Level))

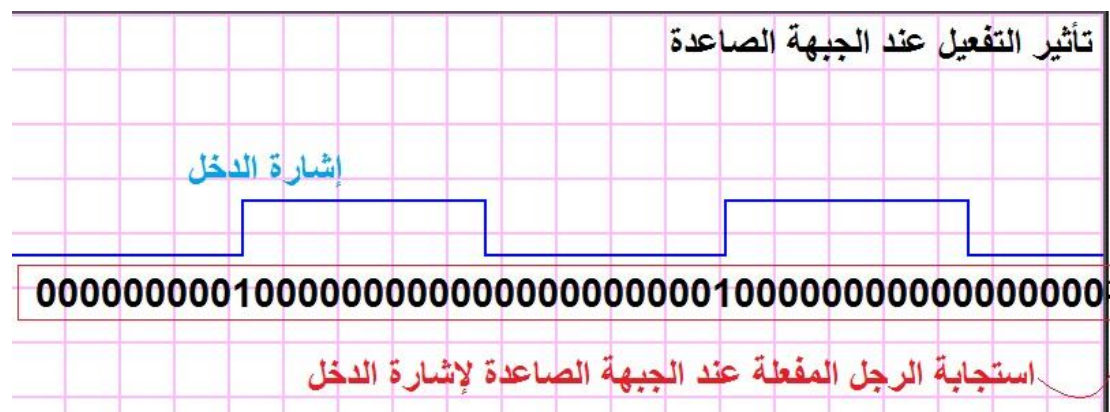


وبذلك سيصبح لدينا 4 أوامر مختلفة يمكن إعطاؤها عن طريق نوعين فقط من الجهود مرتفع ومنخفض ..

ولكن بالنسبة للتحكم ... فسيتلقى واحداً أو صفرًا فقط ..

فلون أننا فعلنا إحدى أرجل التحكم عند المستوى المنخفض ... فعندما نعطي التحكم جهداً مرتفعاً .. سيعتبره صفرًا منطقيًا .. (لأنه مفعّل عند المستوى المنخفض وبالتالي .. غير مفعّل عن المستوى المرتفع)

وبالمثل .. إذا فعلنا الرجل عند الجبهة الصاعدة .. فسيعتبر الانتقال من المستوى المنخفض إلى المستوى المرتفع واحد منطقي .. وكل ما سواه صفر منطقي .. (وبما أن مدة الانتقال هي فترة زمنية صغيرة ... فهذا يعني أنه سيتلقى واحد منطقي لهذه المدة فقط .. وكل ما سواه صفر منطقي)

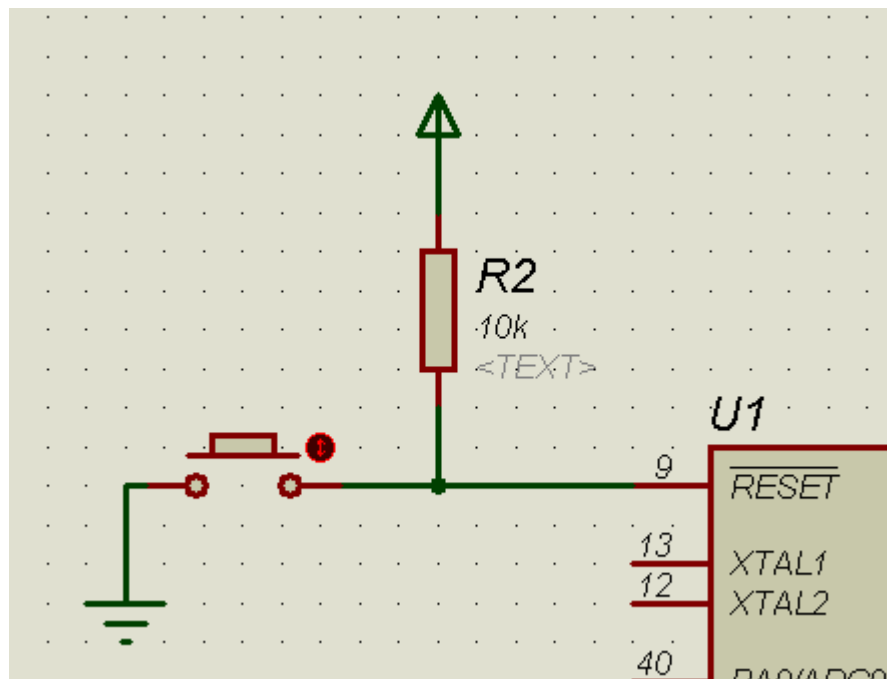


والآن لنتعرف على مقاطعات المتحكم atmega16

مقاطعة إعادة التشغيل :

المقاطعة الرئيسية والأعلى أولوية بين الجميع هي مقاطعة إعادة التشغيل Reset وهي فعالة عند المستوى المنخفض... أي أنه يجب وصلها إلى جهد مرتفع وعندما نريد عمل إعادة تشغيل للمتحكم نقوم بوصلها إلى الأرض .. وطبعاً هذا سيؤدي إلى وصل الجهد المرتفع بالأرض وحدوث تماس ..

ولمنع حدوث ذلك نقوم بوصل مقاومة على الجهد المرتفع قبل وصله برجل اعادة التشغيلبهذا الشكل ..



وهذه ما تسمى مقاومة الرفع .. Pull Up resistor ... لا تنسها أبداً ..

أذكر بأننا غالباً لا نضع قواطع **Switches** على أرجل المقاطعات .. بل أزرار **Buttons** لأننا نريد تأثير **آني فقط وغير مستمر** ... فمثلاً : لو أبقينا ... زر إعادة التشغيل مضغوطاً فلن يعمل المتحكم أبداً ...

ملاحظة : هل ترى هذا الخط فوق كلمة **RESET**



إنه يعني أن هذه الرجل تكون فعالة عند المستوى المنخفض ... فأينما رأيت هذا الخط فاعلم أن هذا هو معناه .. ☺

لكل رجل من أرجل البوابات في المتحكم وظيفتان على الأقل ... الدخول والخروج هو الوظيفة الأولى ...
وبالنسبة للمقاطعات الخارجية ..

فهناك 3 أرجل في المتحكم Atmega16 متخصصة بالمقاطعات الخارجية ...

المقاطعة الخارجية رقم صفر INT0 .. وتأتي أولويتها بعد مقاطعة إعادة التشغيل مباشرة ... في Atmega16 نتحكم بها من الرجل 16 والتي هي الرجل 2 من البوابة D ..

المقاطعة الخارجية رقم واحد INT1 .. بعد سابقتها مباشرة .. نتحكم بها من الرجل 17 وهي الرجل 3 من البوابة D

PD0/RXD	14		
PD1/TXD	15		
PD2/INT0	16		
PD3/INT1	17		
PD4/OC1B	18		
PD5/OC1A	19		
PD6/ICP	20		
PD7/OC2	21		

المقاطعة الخارجية رقم 2 INT2 ... ولها مرتبة متدنية في سلم المقاطعات في المتحكم .. فهناك 16 مقاطعة داخلية أعلى منها في سلم أولويات المقاطعات ... ونتحكم بها من الرجل 3 وهي الرجل 2 في البوابة B

1	PB0/XCK/T0
2	PB1/T1
3	PB2/INT2/AIN0
4	PB3/OC0/AIN1
5	PB4/SS
6	PB5/MOSI
7	PB6/MISO
8	

الفرق الجوهرى بين المقاطعة وبين دالة تكتبها لتحل محل المقاطعة ...

قد يخطر ببال أحدنا ... أن يقوم بعمل اختبار if لأحد أرجل المتحكم ..ثم يقوم باستدعاء تابع بناء على ذلك ...

مثلاً ... إذا كانت الرجل 4 من البوابة C قيمتها 1 .. اذهب goto إلى بداية الدالة main وذلك عوضاً عن Reset

حل معقول ولكن ...

أين سنضع عبارة الشرط if ??

في أول البرنامج ..؟ في آخره ؟ في منتصفه .. أم في كل مكان ..

أم أننا سنستهلك كل البرنامج فقط ليختبر هذه الرجل !!

الفرق الجوهرى بين المقاطعة وأي كود آخر ..

هي أن المتحكم لا يستهلك أي عمليات معالجة داخلية ليختبر المقاطعة ...

فهو لا يقوم ..(كما نظن) باختبار المقاطعة هل تفعّلت أم لا مع كل حركة ونَفَس ... وبذلك لن يستهلك أي عمليات ولن يبطئ من البرنامج الأصلي ...

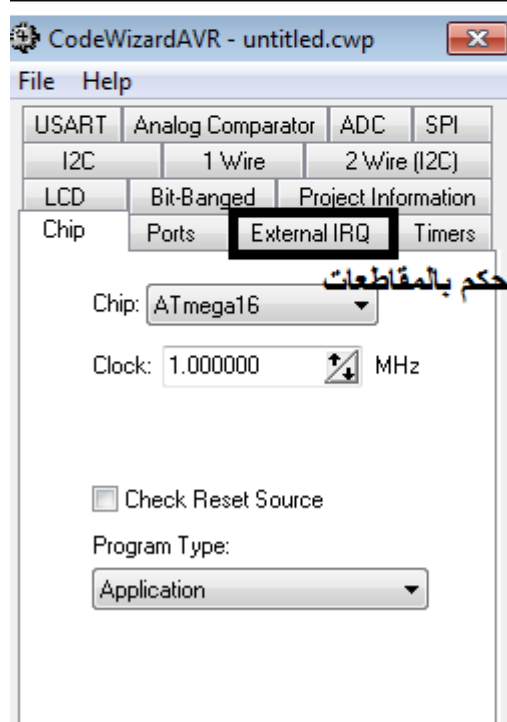
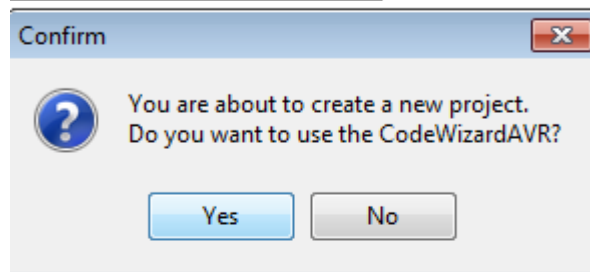
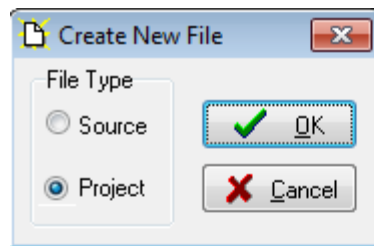
يكفي أن أذكر لك أن عدد المقاطعات الداخلية والخارجية في atmega16 هو 21 ولو كان المعالج يختبرها جميعاً مع كل تعليمة لكان البرنامج الأصلي في عداد الموتي ...

أذكر أيضاً أنه عند برمجة المقاطعة لتكون فعالة يتوجب علينا برمجة رجلها لتكون دخل .. ويجب أن تكون حالة استجابتها للجهد معرفة في مكانين : عند تعريفها كمقاطعة وعند تعريفها كرجل دخل ..

فلن تعمل إذا عرفنا الدخل pullup ثم وضعنا المقاطعة لتعمل عند الجبهة الصاعدة ..لأنها لن تتحسس لها ابداً

والآن إلى القسم العملي ... والمقاطع الرائعة ..

بعد فتح Code Vision وكما تعودنا ..



هذه هي نافذة
التحكم
بالمقاطعات
الخارجية ..

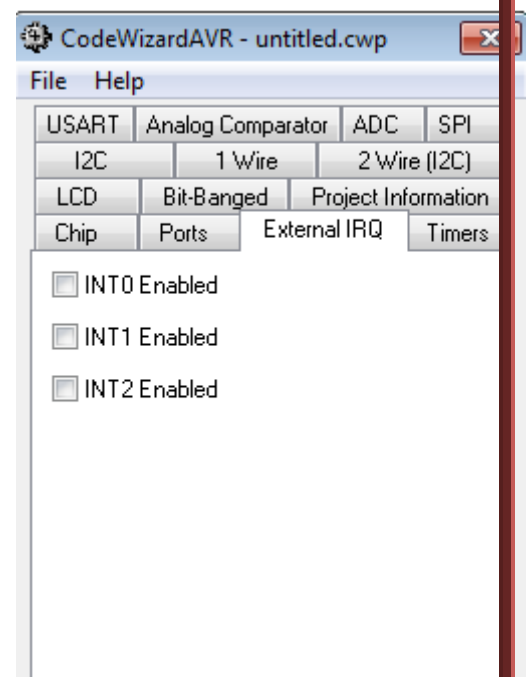
قسم التحكم بالمقاطعات

لاحظ أنه يمكننا التحكم فقط

بالمقاطعات الثلاث الاختيارية ..

أما مقاطعة إعادة التشغيل فهي

مفعلة تلقائياً ..



إذا قمنا بتفعيل إحدى هذه المقاطعات فسيظهر ما يلي :

عدة خيارات لتحديد طريقة الاستجابة للمقاطعة عند المستوى المنخفض أو أي تغيير ..

☒ INT0 Enabled Mode: Low level

☐ INT1 Enabled Mode: Low level

☐ INT2 Enabled Mode: Any change

Falling Edge

Rising Edge

أو عند الجبهة الصاعدة أو الجبهة الهابطة ... (لنختار الجبهة الصاعدة) وننتقل للمقاطعة التالية :

ولها نفس الخيارات السابقة .. سنهتار الجبهة المستوى المنخفض ..

☒ INT0 Enabled Mode: Rising Edge

☒ INT1 Enabled Mode: Low level

☐ INT2 Enabled Mode: Low level

Any change

Falling Edge

Rising Edge

وننتقل لآخر مقاطعة

☒ INT0 Enabled Mode: Rising Edge

☒ INT1 Enabled Mode: Low level

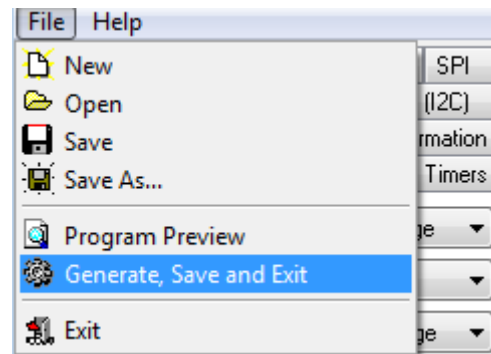
☒ INT2 Enabled Mode: Falling Edge

Falling Edge

Rising Edge

لها خياران فقط .. جبهة هابطة وجبهة صاعدة سنختار

الخيار الاول ..



ثم

لنرى ما الذي أنتجه في الكود ..

ثلاث توابع تنتظر كتابة التعليمات داخلها ... تابع لكل مقاطعة من المقاطعات الثلاث ...

```

23
24 #include <mega16.h>
25
26 // External Interrupt 0 service routine
27 interrupt [EXT_INT0] void ext_int0_isr(void)
28 {
29 // Place your code here
30
31 }
32
33 // External Interrupt 1 service routine
34 interrupt [EXT_INT1] void ext_int1_isr(void)
35 {
36 // Place your code here
37
38 }
39
40 // External Interrupt 2 service routine
41 interrupt [EXT_INT2] void ext_int2_isr(void)
42 {
43 // Place your code here
44
45 }
46

```

والآن..سنرى أين تم تفعيل كل من المقاطعات الثلاث ... وكيف تم تحديد نوع استجابتها من 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 GIFR=0xE0;

```

من الملاحظات..يتوضح لدينا نوع الاستجابة .. ولكن ما هي هذا الكلمات والأرقام ؟
 إنها عبارة عن مسجلات Registers كالتى تحدثنا عنها في درس الدخل والخرج

أي أنها خانات من الذاكرة مخصصة لحفظ بيانات محددة جداً ..فكل بت داخل هذه المسجلات له وظيفة محددة ..

سنتكلم عنها في نهاية الدرس بإذن الله (حتى نركز على التنفيذ)

```

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

```

هذه هي التعليمة الخاصة بتفعيل المقاطعات بشكل عام

وهي مكتوبة بلغة التجميع الخاصة بالمتحكم حيث يمكن إضافة تعليمات

#asm("هنا")

هذه اللغة داخل الكود عن طريق كتابتها

والآن لنتذكر ما قلناه في البداية

ومن خلال البرنامج نحن نتحكم بما يلي :

```
127 GICR|=0xE0;
128 MCUCR=0x03;
129 MCUCSR=0x00;
130 GIFR=0xE0;

asm("sei") .. هل هناك مقاطعات أم لا ..
٢-ما هي المقاطعات التي نريدها أن تعمل ...
٣-مى يستجيب المتحكم للمقاطعة ؟ (عذ النبضة الصاعدة أو الهابطة أو ثبات الجهد أو أي تغير)
٤-ما هو الكود الذي يجب تنفيذه عند الاستجابة للمقاطعة ؟

27 interrupt [EXT_INT0] void ext_int0_isr(void)
28 {
29 // Place your code here
30 }
31
```

حسناً ... لنخطط لما سنفعله ..

- 1- سنقوم بكتابة برنامج بسيط يقوم بزيادة قيمة **PORTA** كل ثانية ..
- 2- سنقوم بكتابة كود تجريبي داخل كل من توابع المقاطعات الثلاث ..
- 3- سنقوم بتركيب الدارة بحيث نضع القواطع المناسبة بالشكل المناسب على كل أرجل المقاطعات
- 4- نضع قاطعا مع مقاومة رفع على رجل إعادة التشغيل ..
- 5- نشغل البرنامج ونلاحظ طريقة عمل إعادة التشغيل .. وكل من المقاطعات الثلاث ..

لنبدأ على بركة الله ..

الخطوة الأولى :

```
143
144 DDRA=0xFF;      اعتبر كل الأرجل خرج
145 PORTA=0x00;     أعطاها قيمة ابتدائية صفرية
146 while (1) :    كرر ما يلي :
147 {
148 // Place your code here
149 PORTA++;
150 delay_ms(1000); انتظر لمدة ثانية
151 };
```

ولا ننسى `#include<delay.h>` في بداية الكود ..

الخطوة الثانية :

```
26 // External Interrupt 0 service routine
27 interrupt [EXT_INT0] void ext_int0_isr(void)
28 {
29 // Place your code here
30     DDRC=0xFF;
31     PORTC=0xFF;
32 }
33
34 // External Interrupt 1 service routine
35 interrupt [EXT_INT1] void ext_int1_isr(void)
36 {
37 // Place your code here
38     DDRC=0xFF;
39     PORTC=0b01010101;
40 }
41
42 // External Interrupt 2 service routine
43 interrupt [EXT_INT2] void ext_int2_isr(void)
44 {
45 // Place your code here
46     while(PORTA!=0)
47     {
48         PORTA--;
49         delay_ms(1000);
50     }
51 }
```

المقاطعة الأولى تقوم بتفعيل كل أرجل البوابة C

المقاطعة الثانية تقوم بتفعيل أرجل البوابة C بشكل متناوب

تقوم بالعد تنازلياً حتى يعود PORTA إلى الصفر

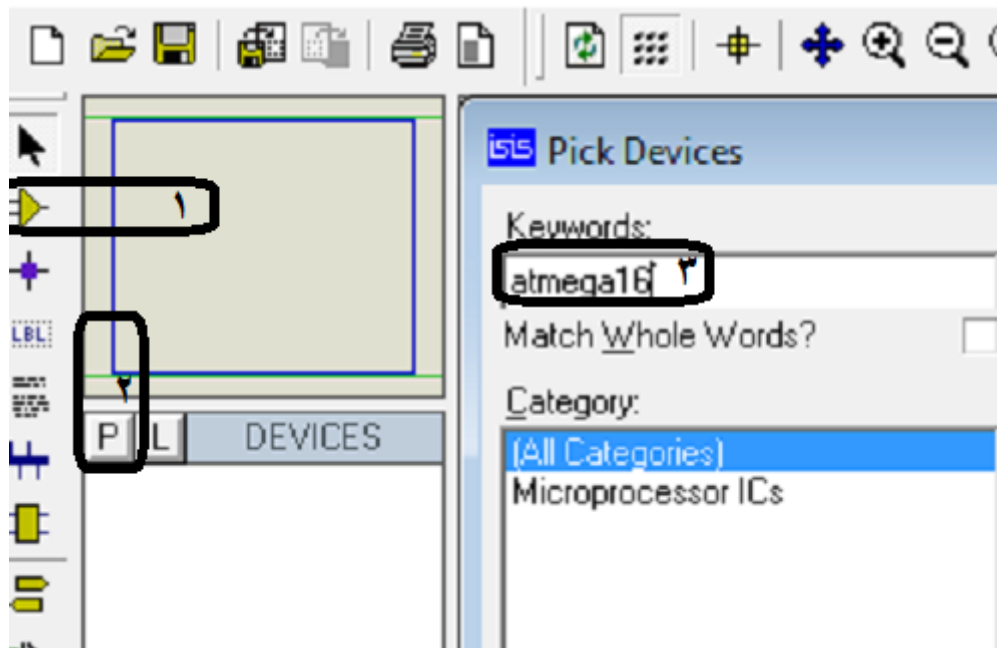
نقوم بعمل Build للكود.. (اضغط Ctrl + F9) ...

ونتأكد من خلوه من الأخطاء ..

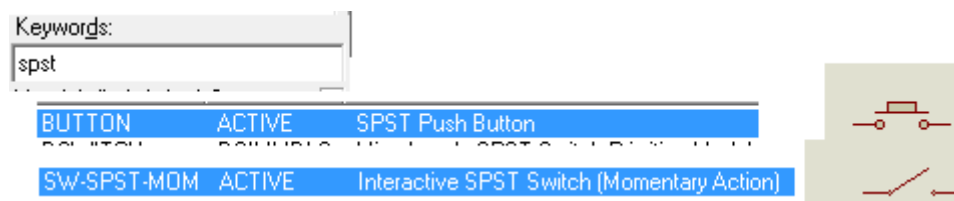
```
201 line(s) compiled
No errors
No warnings
```

والآن إلى تركيب الدارة والخطوة الثالثة ..

نفتح proteus كالمعتاد ..



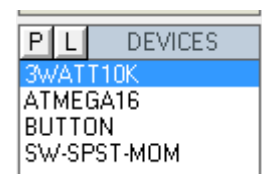
ونحتاج إلى أزرار و قواطع ..



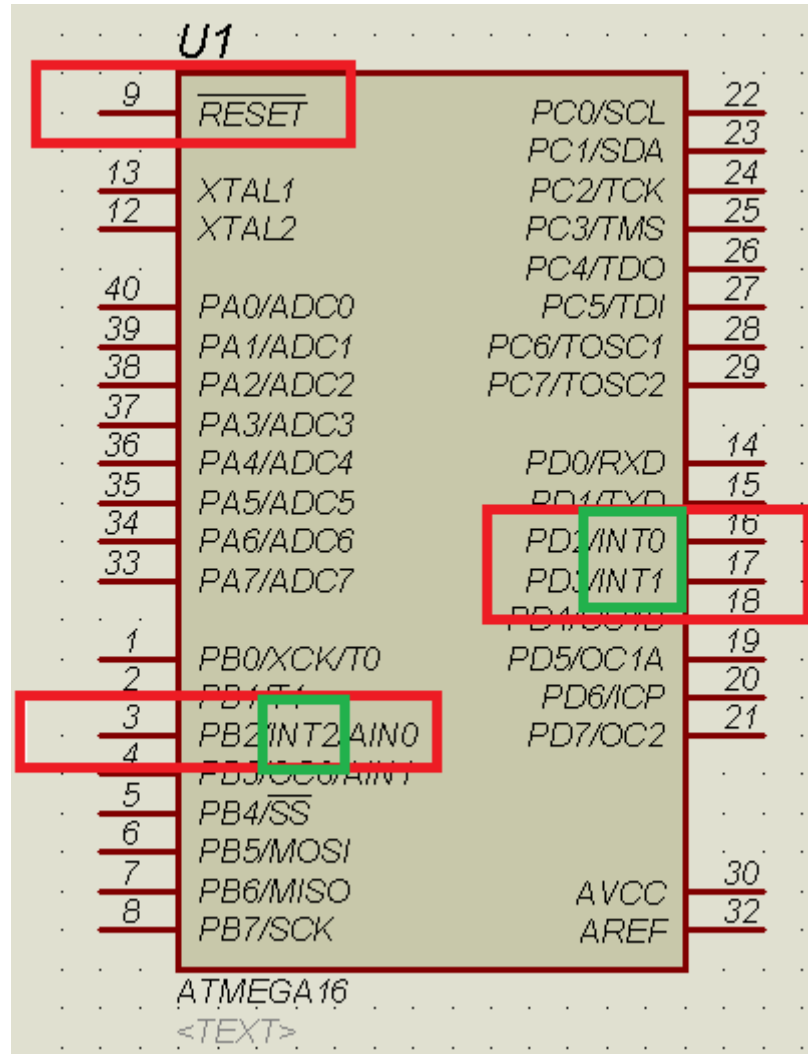
ومقاومات للرفع في حال الحاجة



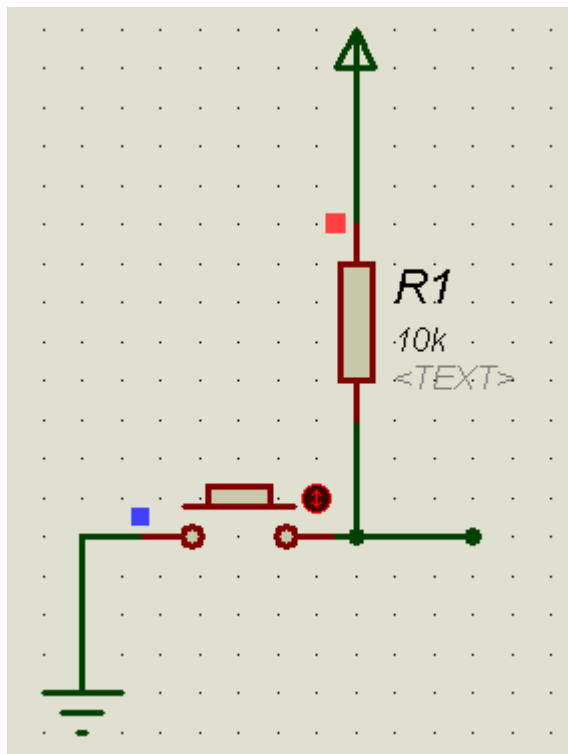
وهذه هي القائمة الحالية للعناصر المطلوبة



..لاحظ اماكن أرجل المقاطعات في المتحكم



سيكون الشكل الرئيسي لأي قاطع هو كما يلي :

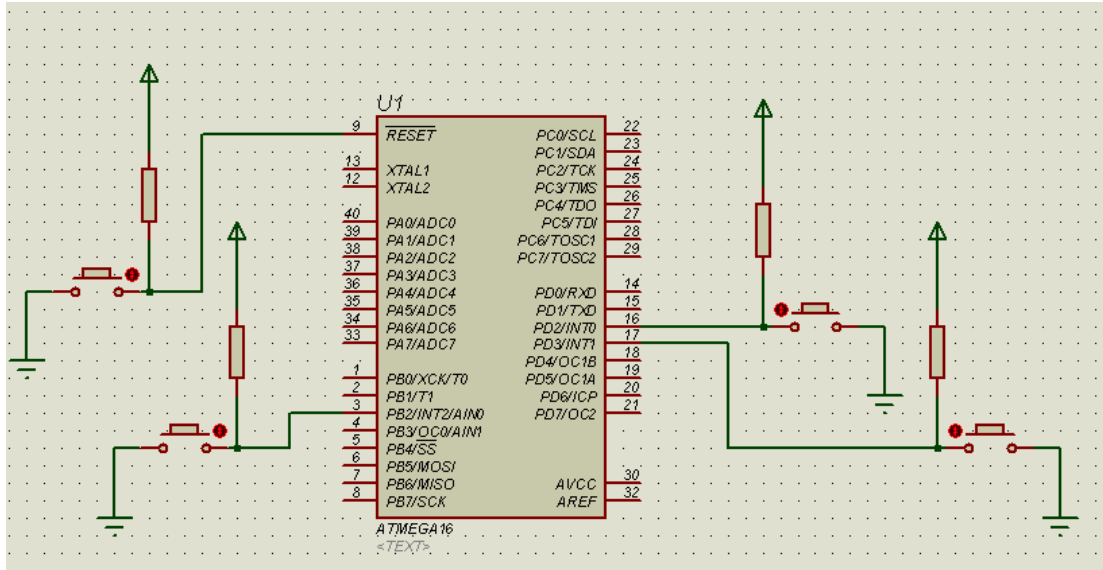


النقطة الحرة هي مخرج القاطع والتي سنصلها إلى رجل المتحكم ..

لاحظ أن المخرج يعطي مستوى مرتفع

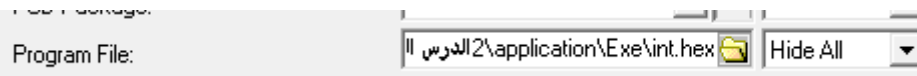
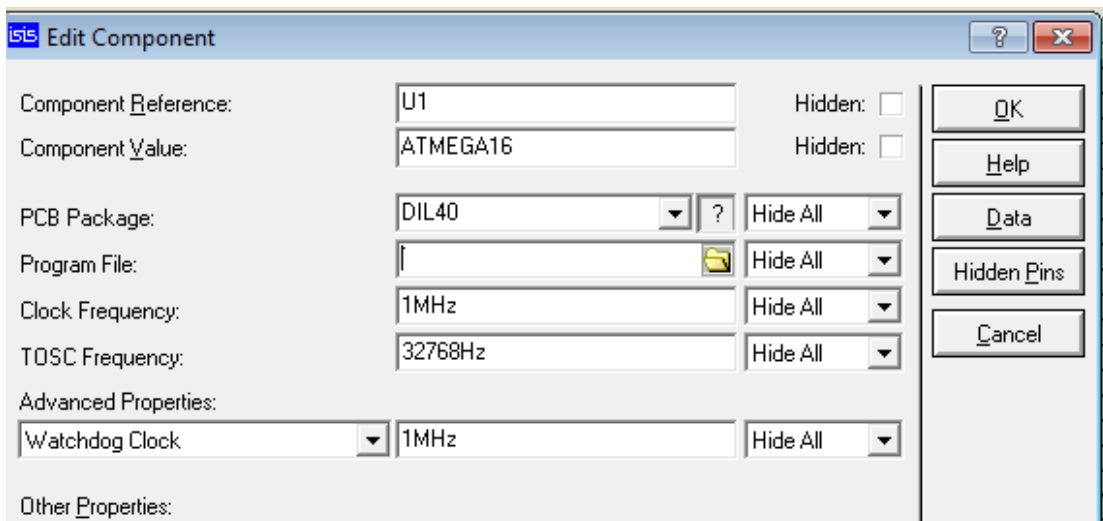
عندما يكون القاطع مفتوح ومستوى منخفض عندما يكون القاطع مغلق ..

وسيكون شكل الدارة بعد ربط القواطع على أرجل المتحكم كما يلي :



بالضغط مرتين على المتحكم ... تظهر النافذة التالية ..

نقوم بتحديد البرنامج الذي نريد للمتحكم ان ينفذه ..



ونبدأ المحاكاة ...



الشيء الوحيد الذي سيتغير هو الجهود على أرجل البوابة A بما أن قيمة PORTA تزيد كل لحظة ..

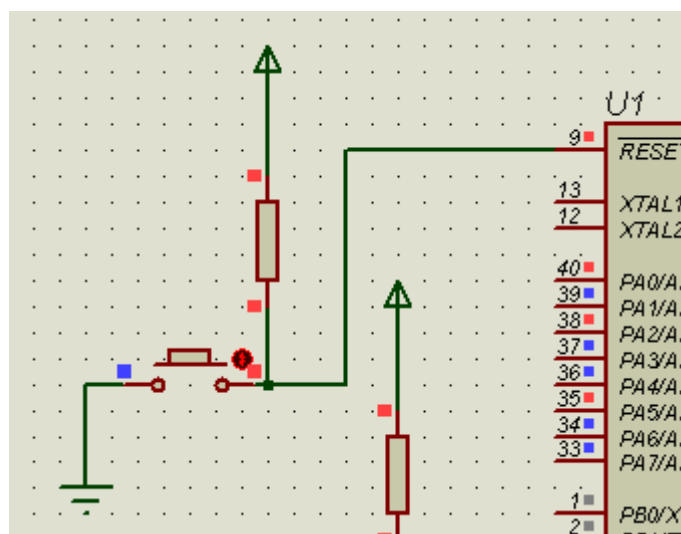
فإن الجهود ستصبح كما يلي :

0000 ثم 0001 ثم 0010 ثم 0011 ثم 0100 ثم 0101 ثم 0110 وهكذا 0111 ...

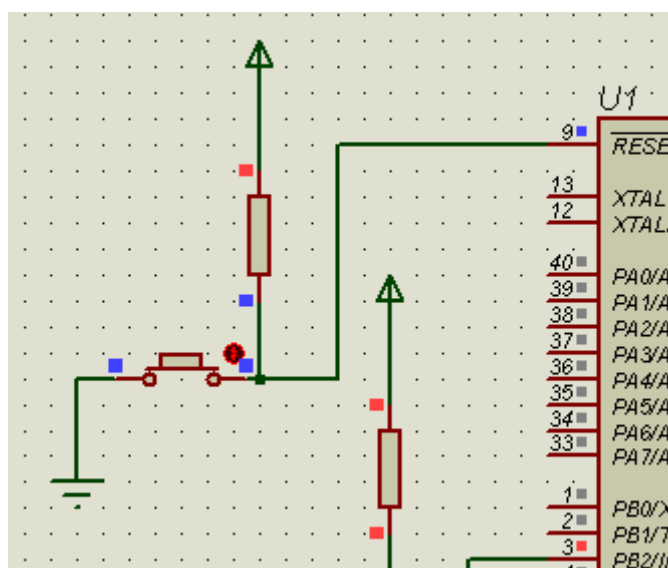
أي أنها تابعة للقيمة الثنائية binary للPORTA

والآن لنجرب عمل Reset كما قلنا .. هذه المقاطعة فعالة عند الانتقال من المستوى المرتفع إلى المستوى المنخفض ... وهذا بالضبط ما دفعنا لتصميم القاطع بشكله السابق (مع مقاومة الرفع)

قبل ضغط الزر :

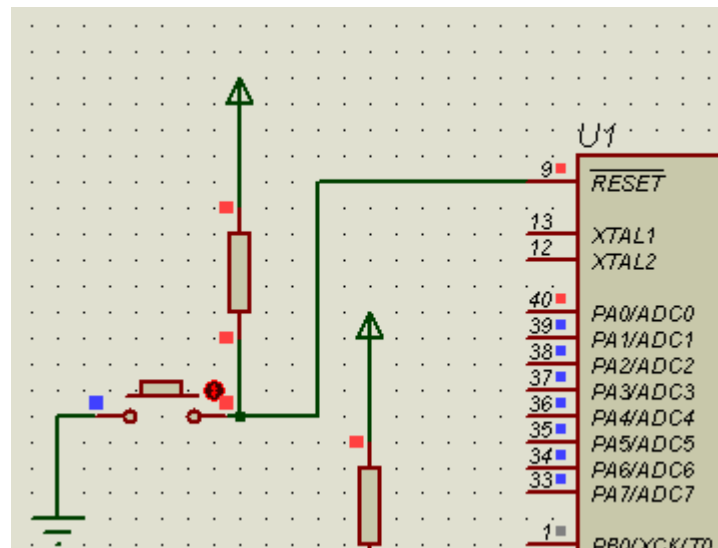


عند ضغط الزر :



لاحظ أن لون الأرجل أصبح رمادياً وكأن المتحكم لم يعمل بعد .. عاد إلى نقطة الصفر ..

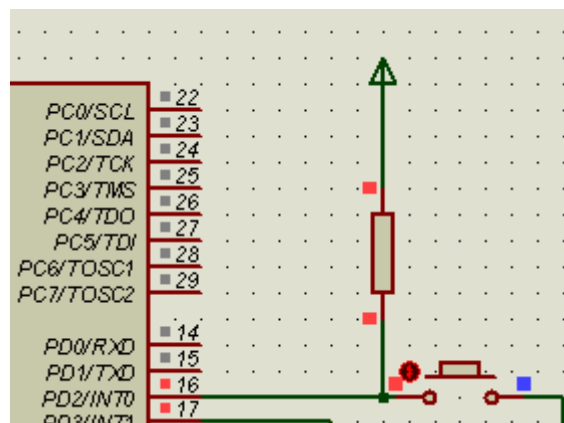
وعند ترك الزر ليعود كما كان ..يعود المتحكم ليبدأ البرنامج من جديد ..



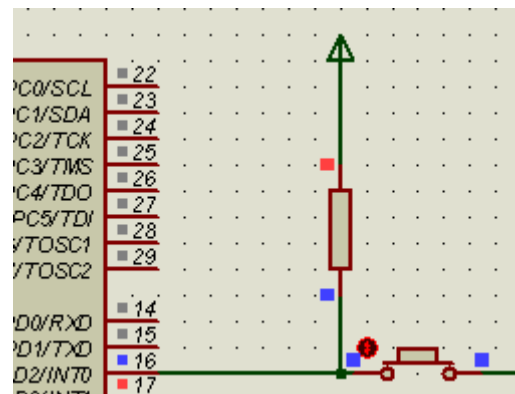
لنجرب باقي المقاطعات ..

المقاطعة رقم 0 .. قمنا بتفعيلها عند الجبهة الصاعدة .. (أي أن إغلاق القاطع لن يكون له تأثير)

قبل ضغط زر المقاطعة 0 :

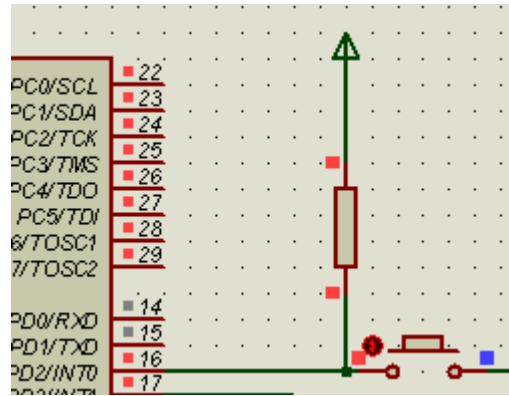


بعد ضغط الزر :



لم يحدث أي تغيير ...

أما عند ترك الزر ليعود كما كان ..(يحدث انتقال من المستوى المنخفض إلى المرتفع) :

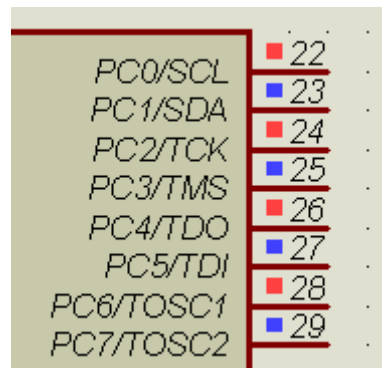
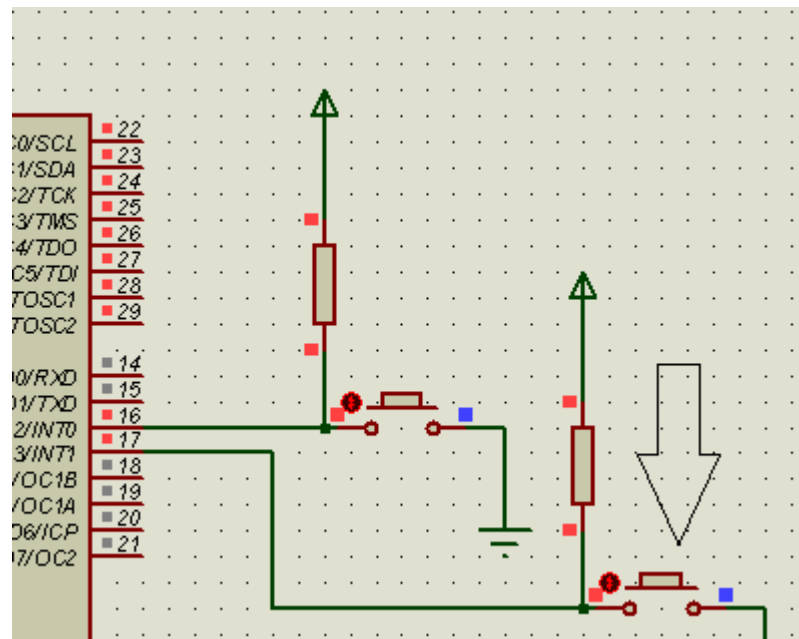


تم تنفيذ المقاطعة ... (والتي كانت وظيفتها : PORTC=0xFF)

هل توضحت آلية عمل الجبهة الصاعدة الآن .. ؟

لننتقل للمقاطعة رقم 1 والتي قمنا بتفعيلها عن المستوى المنخفض :

قبل ضغط الزر ..



عن الضغط ..سيتم تنفيذ لأنها فعالة

عند المستوى المنخفض ..

أي أن الصفر تعتبره واحدا منطقياً

وبما أنها فعالة عند المستوى المنخفض

فطالما أن الزر مضغوط .. سيبقى المتحكم

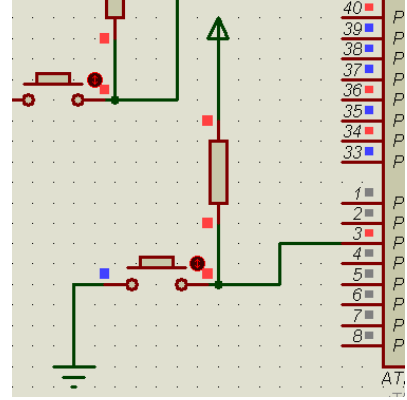
ينفذ دالة المقاطعة .. ولن يخرج منها

حتى يتغير المستوى المنخفض إلى مستوى مرتفع (وهذا ما يميز التفعيل عند المستوى المنخفض عن الجبهة الهابطة)

والآن إلى المقاطعة الأخيرة ...

نتذكر أنها كانت فعالة عند الجبهة الهابطة

قبل ضغط الزر :

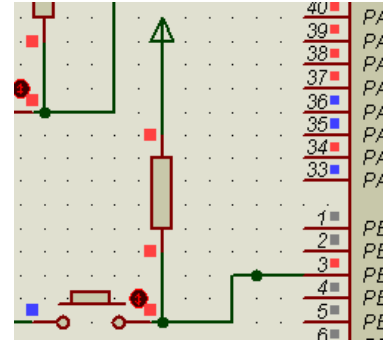


عند الضغط مباشرة (حيث ننتقل من مستوى مرتفع إلى مستوى منخفض Falling Edge)

يجب أن تتفعل المقاطعة ويتم تنفيذ التابع الخاص بها ..

(ولكن لسبب أجهله المقاطعة تتفعل عند إزالة الضغط عن الزر .. أي عند الجبهة الصاعدة .(ربما خطأ من المحاكاة !!))

ما يهمنا الآن .. هو أنه عندما يبدأ تنفيذ تابع المقاطعة ... سيأخذ وقتاً خلال العد العكسي



خلال هذا الوقت سيتوقف تنفيذ الكود الأصلي .. ويبدأ بتنفيذ كود المقاطعة ...

ويمكننا عمل إعادة التشغيل أثناء عمل هذه المقاطعة ..

ولكن : ما الذي سيحدث لو استدعينا المقاطعة رقم 0 أو المقاطعة رقم 1 أثناء عمل المقاطعة رقم 2 ???

يقوم المعالج بتخزين أمر المقاطعة الجديدة إلى حين انتهاء تنفيذ المقاطعة الحالية ..

ولو كان هناك عدة مقاطعات أثناء تنفيذ المقاطعة الحالية يتم مراكمة الأوامر إلى حين انتهاء المقاطعة الحالية

وهنا يأتي دور الأولويات : فعملية مراكمة المقاطعات تتم بحيث يكون للمقاطعة ذات الأولوية الأعلى السبق في التنفيذ بعد الانتهاء من المقاطعة الحالية ..

يمكنك تجربة ذلك بضغط زر المقاطعة 0 أثناء عمل المقاطعة 2 .. وسيتم تنفيذها بعد انتهاء الحالية ..

حاولت تفعيل المقاطعة 1 أثناء عمل المقاطعة 2 ولكنها لم تخزن ... (لست متأكداً من السبب ولكن أظن أن هذا لأننا جعلناها تستجيب للمستوى المنخفض .. وليست للجبهة الصاعدة أو الهابطة ..(وأجهل السبب)).

بهذا نكون قد وصلنا إلى نهاية درس المقاطعات الخارجية .. أسأل الله لي ولكم التوفيق

الله علمنا ما ينفعنا وانفعنا بما علمتنا وزدنا علما يا رب العالمين ...

وصلى الله على سيدنا محمد وعلى آله والحمد لله رب العالمين ..