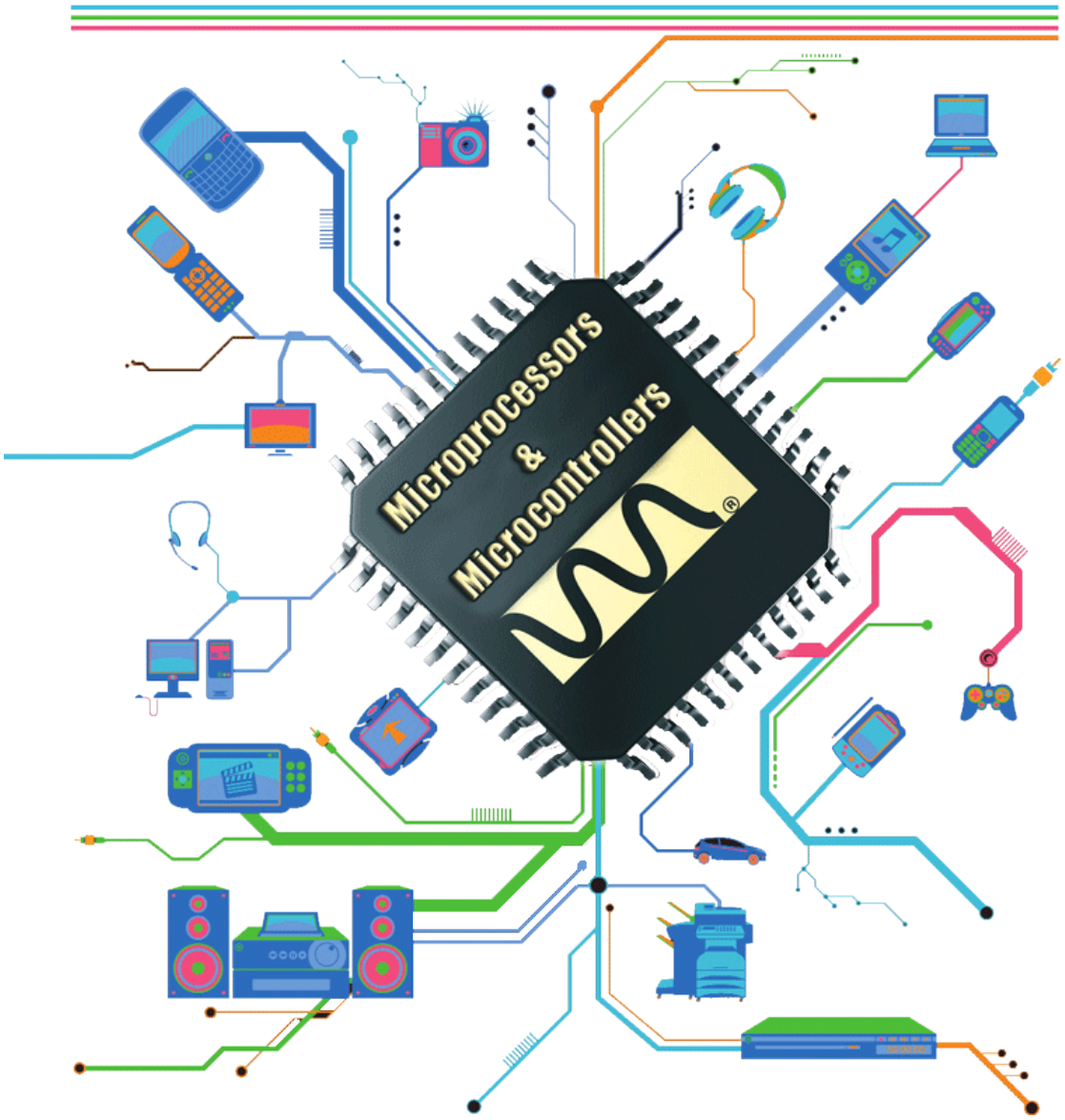
جامعـــــــــــة حلـــــــــــب
كلية الهندسة الكهربائية والإلكترونية
قسـم هندسـة التحكـم والأتمتـة

﴿الجلسات العملية لمادة المعالجات والمتحكمات المصغرة﴾

Microprocessors & Microcontrollers Lab Sessions

السنة الثالثة | قسم اتصالات

﴿الجلسة العملية الخامسة﴾



م. وليد بليد

Wednesday, April 04, 2012

Copyright © 2012 Walid Balid - All rights reserved.

الجلسة العملية الخامسة

نظرة عامة (Overview):

هذه المحاضرة تشرح تعليمات الإزاحة والدوران وتقدم مثالاً تطبيقياً عليها. ثم تقدم المقاطعات في متحكمات AVR وأنواعها وتشرع في المقاطعات الخارجية ومبدأ عملها ولمحة عن المسجلات الداخلية للمقاطعات الخارجية. ثم تقدم تطبيقاً عملياً لاستثمار المقاطعات الخارجية في متحكمات AVR وبرمجتها في البيئة BASCOM-AVR ومحاكاتها في البيئة Proteus. وأخيراً طريقة توصيل لوحة مفاتيح مصفوفية ومنهجية المسح.

1-5 عمليات الإزاحة والتدوير (Shifting and Rotating):

تستخدم تعليمات الإزاحة والدوران بهدف إزاحة بت أو أكثر – من بايت أو أكثر – إلى اليمين أو إلى اليسار؛ وهناك فرق بين عملية الإزاحة وعملية الدوران لقيمة ما حيث:

- في الإزاحة كل بت يخرج (من اليمين أو اليسار) يدخل مكانه صفر. مثال ذلك: إذا تم إزاحة القيمة &B11111111 ثمان مرات إلى اليمين أو اليسار فستصبح القيمة عندها &B00000000.

مثال:

```
A = &B11011011
Shift A , Right , 1
```

0	1	1	0	1	1	0	1

```
Shift A , Right , 4
```

0	0	0	0	1	1	0	1

- في الدوران كل بت يخرج (من اليمين أو اليسار) يدخل من الطرف الآخر – أي يتم تدوير القيمة. مثال ذلك: إذا تم تدوير القيمة &B11111111 ثمان مرات إلى اليمين أو اليسار فستبقى القيمة على حالها. وإذا تم تدوير القيمة &B00001111 أربع مرات إلى اليمين أو اليسار فستصبح &B11110000.

مثال:

```
A = &B11011011
Rotate A , Right , 1
```

1	1	1	0	1	1	0	1

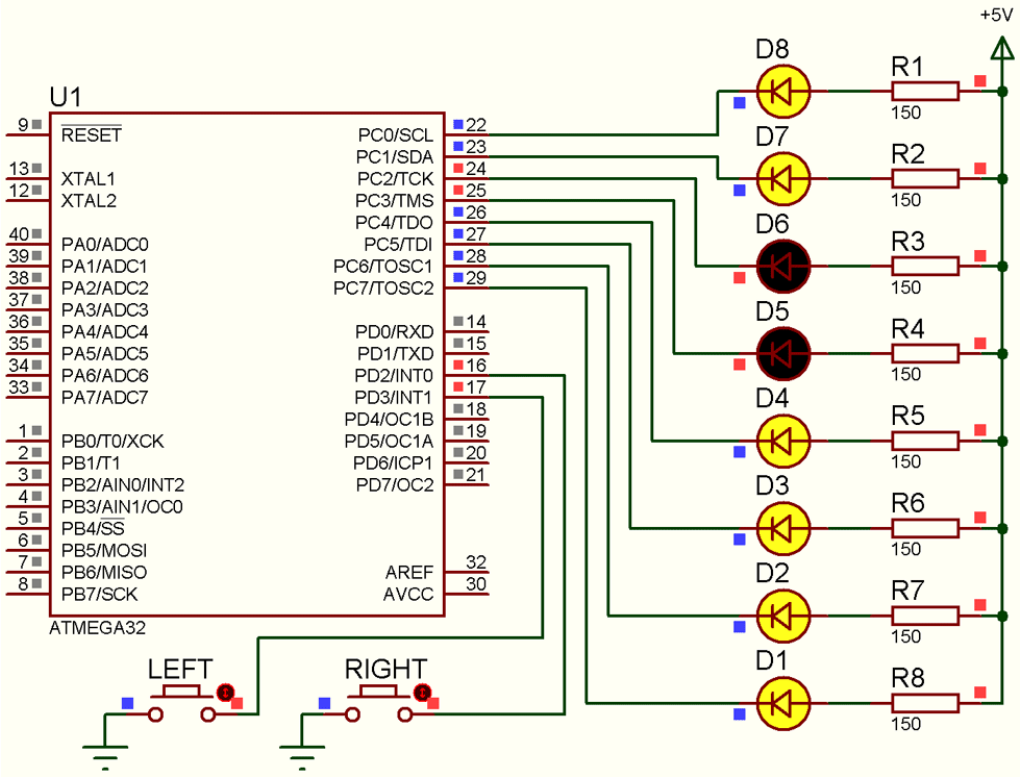
```
Rotate A , Right , 4
```

1	1	0	1	1	1	0	1

2-5 تعليمات الإزاحة والتدوير في Bascom (Shifting and Rotating Instructions in BASCOM-AVR):

شرح التعليمة	التعليمة البرمجية
إزاحة بت من متحول (var) إلى اليمين أو اليسار وعدد خانات الإزاحة محددة بـ [, shift]	`Shift var , Right/Left [, shift]`
تدوير بت من متحول (var) إلى اليمين أو اليسار وعدد خانات الدوران محددة بـ [, shift]	`Rotate var , Right/Left [, rotate]`

التجربة الحادية عشرة: استخدم المفاتيح اللحظية S1-S2 (PIND.3, PIND.2) على اللوحة التعليمية لإزاحة وتدوير قيمة تظهر على الثنائيات الضوئية الثمانية (LEDs) الموصولة إلى البوابة PORTC.



الشكل5-1 توصيل الثنائيات والمفاتيح مع المتحكم للتجربة 11

البرنامج Exp.11.bas في بيئة BASCOM-AVR:

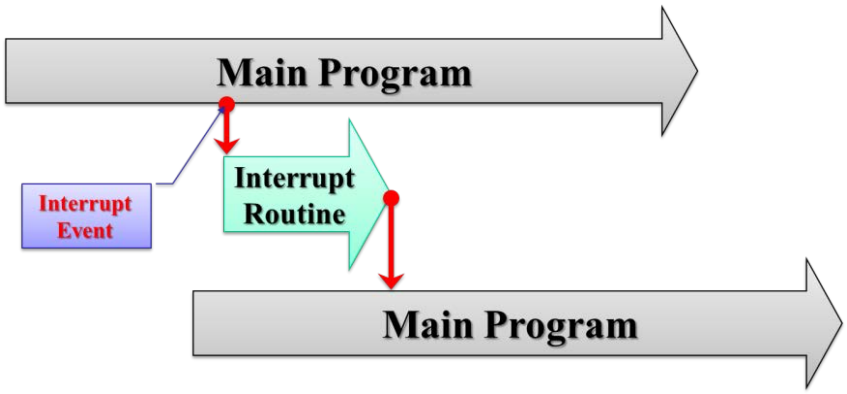
```
' ********************************************************************************
' * Title          : Exp.11.bas                                                  *
' * Target MCU     : ATMega128A                                                  *
' * Author         : Walid Balid                                                 *
' * IDE            : BASCOM AVR 2.0.7.3                                          *
' * Peripherals    : DIP-Switch                                                  *
' * Description    : Shift/Rotate                                                *
' ********************************************************************************
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'-----------------------[Definitions]
$regfile = "m32def.dat"
$crystal = 8000000
'------------------------
'-----------------------[GPIO Configurations]
Config Portc = Output : Leds Alias Portc

Config Pind.2 = Input : Sw_1 Alias Pind.2 : Portd.2 = 1    'PU Internal Resistor
Config Pind.3 = Input : Sw_2 Alias Pind.3 : Portd.3 = 1
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[Main Program]
Leds = &B11011011
Do
   Debounce Sw_1 , 0 , Shift_r , Sub
   Debounce Sw_2 , 0 , Shift_l , Sub
Loop
End
'---<[End Main]
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[Shift LEDs to Right]
Shift_r:
  Shift Leds , Right , 1
   'Rotate Leds , Right , 1
Return
'---<
'--->[Shift LEDs to Left]
Shift_l:
   Shift Leds , Left , 1
   'Rotate Leds , Left , 1
Return
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

5-3 المقاطعات في متحكمات AVR (Interrupts in AVR MCUs):

تعرف المقاطعة (Interrupt) بأنها آلية إعلام داخلية (دون تدخل المتحكم في آلية عمل المقاطعة داخلياً) تعلم وحدة المعالجة المركزية بوجود حدث يجب معالجته مما يسبب تغير في سير البرنامج الرئيسي لإنجاز برنامج فرعي يسمى برنامج خدمة المقاطعة. لتوضيح الفكرة نأخذ على سبيل المثال برنامج قراءة حالة مفتاح موصول إلى قطب المتحكم، وبالتالي يوجد لحالتين لبرمجة عمل المفتاح:

1) الطريقة التقليدية (Polling): يمكن أن نطلق على هذه الطريقة المقاطعة البرمجية (Software Interrupts) وتتم بالفحص الدوري لحالة القطب (... **If** Sw = 0 **Then**) من أجل اكتشاف تغير حالة المفتاح ويتم هذا بشكل برمجي، وبالتالي سوف يشغل المتحكم في عملية الفحص الدوري المتكرر للتحقق من حالة المفتاح بشكل دائم، الأمر الذي سيؤدي إلى ضياع في قدرة المعالجة للمتحكم واستهلاك في الطاقة.

2) طريقة المقاطعة (Interrupt): يمكن أن نطلق على هذه الطريقة مقاطعة الكيان الصلب (Hardware Interrupts) وتتم من خلال آلية مستقلة مبنية ضمن المتحكم تقوم على مقاطعة المتحكم عندما تتحقق الحالة المطلوبة فقط، وبالتالي لن ينشغل المتحكم بتفحص المفاتيح من أجل معرفة فيما إذا تغيرت حالة المفتاح أم لا، وإنما عندما تتغير الحالة المنطقية للمفتاح على قطب المقاطعة سوف يتم مقاطعة المتحكم ويقفز إلى برنامج خدمة المقاطعة الخارجية المتعينة من أجل تنفيذها.



الشكل5-2 تمثيل عملية المقاطعة والقفز من البرنامج الرئيسي إلى برنامج المقاطعة والعودة إلى البرنامج الرئيسي

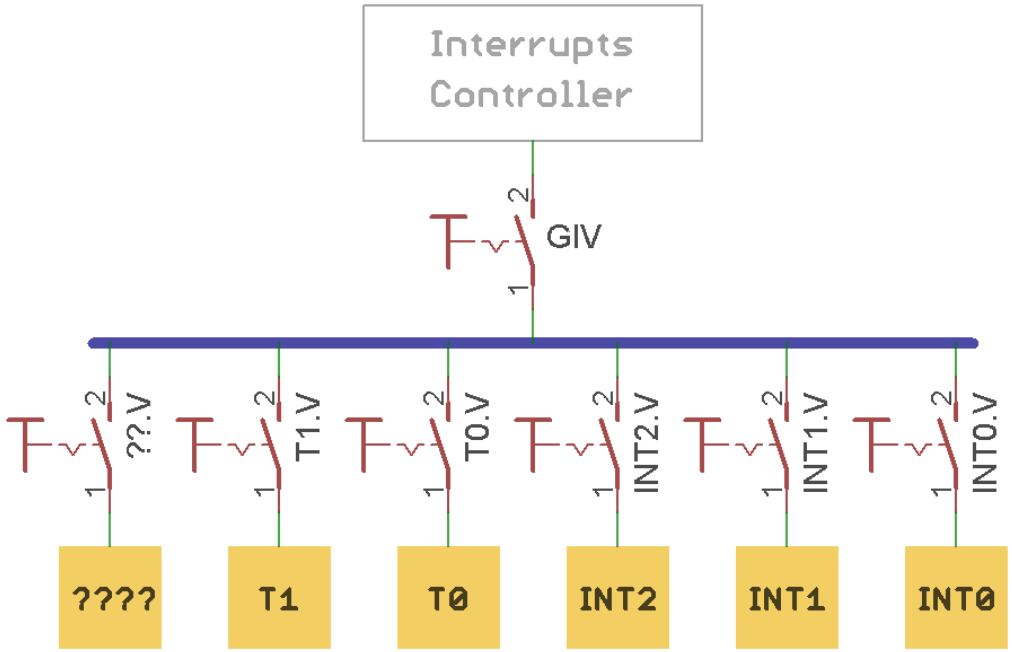
5-4 مصادر المقاطعات في متحكمات AVR (AVR MCU Interrupts Sources):

تمتلك العائلة AVR مجموعة كبيرة من مصادر المقاطعة المختلفة، وتمتلك كل من هذه المقاطعات عنوان مستقل في حيز ذاكرة البرنامج، ولكل مقاطعة خانة تمكين مستقلة، فعندما نرغب بتفعيل إحدى المقاطعات فإنه يتوجب علينا تفعيل الخانة المخصصة لها في مسجل التحكم بالمقاطعة المعنية إلى جانب تفعيل خانة تمكين المقاطعة العامة I في مسجل الحالة SREG.

Bit	7	6	5	4	3	2	1	0
SREG	**I**	**T**	**H**	**S**	**V**	**N**	**Z**	**C**
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

الشكل5-3 الخانة 7 شعاع المقاطعات العام (Global Interrupt Vector) في مسجل الحالة SREG

يمكن تمثيل شعاع المقاطعات العام (Global Interrupt Vector) بقاطع رئيسي، وباقي المقاطعات كقواطع فرعية، وبالتالي لا يكفي تفعيل المقاطعة الفرعية وإنما يجب أيضاً تفعيل شعاع المقاطعات العام معها كما هو مبين على الشكل التالي.



الشكل5-4 تمثيل لحالة شعاع المقاطعات العام (Global Interrupt Vector) والمقاطعات الفرعية الأخرى

5-5 تصنيف المقاطعات في متحكمات AVR (AVR MCU Interrupts Classification):

يمكن تقسيم المقاطعات في العائلة AVR إلى مجموعتين رئيسيتين، تضم كل مجموعة من المجموعتين مجموعات فرعية أخرى:

- مقاطعات خارجية (External Interrupts): لها ارتباط مباشر مع الأقطاب الفيزيائية للمتحكم وتستجيب لأحداث خارجية مطبقة على أقطاب المتحكم وهي:
  - مقاطعة التصفير (Reset).
  - مقاطعات الطلب الخارجي (INT0 ~ INT7).
- مقاطعات داخلية (Internal Interrupts): لها ارتباط مع الوحدات المحيطية الداخلية فقط للمتحكم وهي:
  - مقاطعات المؤقتات (OV, COMP).
  - مقاطعات حادثة المسك للمؤقتات/عدادات (ICP).
  - مقاطعات العدادات.
  - مقاطعة اكتمال التحويل للADC.
  - مقاطعة اكتمال الإرسال للنافذة التسلسلية SPI (STC).
  - مقاطعات النافذة التسلسلية USART (RX,TX,UDR).

- مقاطعة المقارن التشابهي (ANALOG COMP).
- مقاطعة اكتمال كتابة المعطيات إلى الذاكرة EEPROM.
- مقاطعة النافذة التسلسلية TWI.
- وغيرها...

6-5 مبدأ عمل المقاطعات في متحكمات AVR (AVR MCU Interrupts Basic):

تتمتع وحدة المقاطعات بمسجلات تحكم خاصة في المساحة المخصصة للدخل/الخرج بالإضافة إلى خانة تفعيل خاصة (I) ضمن مسجل الحالة، وتتمتع كل مقاطعة بشعاع مقاطعة منفصل في جدول أشعة المقاطعات، ويكون لكل مقاطعة أولوية متناسبة مع موقع شعاعها ضمن الجدول، فكلما كان عنوان شعاع المقاطعة أدنى كلما كانت ذات أولوية أعلى.

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

الشكل5-5 عناوين أشعة المقاطعات في المتحكم ATmega32A

الشكل5-5 يبين القائمة الكاملة للمقاطعات في المتحكم ATmega32A وعناوين الأشعة لهذه المقاطعات، كما تحدد هذه القائمة أيضاً مستويات الأولوية للمقاطعات، فالعنوان الأخفض (000$) هو الشعاع ذو الأولوية الأعلى، فلمقاطعة التصفير مثلاً الأولوية الأعلى ومن ثم المقاطعة الخارجية INT0 وهكذا...

عندما تحدث مقاطعة ما فإنه يتم **تلقائياً** تصفير خانة تمكين المقاطعة العامة (I = 0)، وبالتالي تحجب جميع المقاطعات الأخرى إلى حين الانتهاء من المقاطعة الحالية، إلا أن المبرمج يستطيع أن يُفعّل خانة تمكين المقاطعة العامة (I = 1) داخل برنامج خدمة المقاطعة في حال أريد الإبقاء على المقاطعات الأخرى. وعندما ينفذ المتحكم تعليمة العودة RETURN الواقعة في نهاية برنامج خدمة المقاطعة، فإنه يتم **تلقائياً** تفعيل خانة تمكين المقاطعة العامة (I = 1).

7-5 ملاحظات هامة حول المقاطعات في متحكمات AVR:

- عند استخدام أي مقاطعة فإنه يجب تفعيل شعاع المقاطعات العام I والذي يمكن تمثيله كقاطع رئيسي لجميع المقاطعات.
- عندما تحدث مقاطعة ما يتم تلقائياً تصفير شعاع المقاطعات العام I في مسجل الحالة وبذلك يتم إلغاء الاستجابة لجميع المقاطعات الأخرى، وتتم عملية إعادة تفعيل الخانة I بعد الانتهاء من تنفيذ أية مقاطعة تلقائياً أيضاً.
- إذا تحقق شرط إحدى المقاطعات أو أكثر، وكانت خانة تمكين المقاطعة العامة غير مفعلة (I = 0)، فإن أعلام المقاطعة التي حدثت ستفعل (''1'') تلقائياً وتبقى كذلك إلى أن يتم تأهيل خانة المقاطعة العامة (I = 1)، فإذا ما تم تفعيل خانة المقاطعة العامة (I = 1) عندها يبدأ المتحكم بتنفيذ برامج خدمة المقاطعة بحسب أولويات أشعتها.
- في حال كان المتحكم يقوم بتنفيذ برنامج خدمة مقاطعة ما، وفي نفس الوقت حصلت مقاطعة أخرى، فإن المتحكم سوف يكمل المقاطعة الجارية ويقوم بتخزين المقاطعة الطارئة حتى إذا انتهى من المقاطعة الجارية عاد إلى البرنامج الرئيسي ونفذ تعليمة واحدة على الأقل من البرنامج الرئيسي ثم سيستدعى المقاطعة الطارئة ويقوم بتنفيذها. وأما في حال حصلت عدة مقاطعات أثناء عمل المتحكم في برنامج خدمة مقاطعة ما، فإنه يقوم بمراكمتها حسب أولويتها ويقوم بتنفيذها وفق تسلسل الأولوية بعد انتهائه من برنامج خدمة المقاطعة الجارية.
- عندما ينتهي تنفيذ برنامج خدمة مقاطعة ما، فإن سيتم العودة إلى البرنامج الرئيسي وينفذ المتحكم تعليمة واحدة على الأقل قبل أن ينتقل لتنفيذ مقاطعة أخرى في حال وجود مقاطعات متراكمة أثناء برنامج خدمة المقاطعة الأخيرة.
- ينصح بأن يكون برنامج خدمة المقاطعة قصيراً جداً (يمكن تفعيل علم تحقق المقاطعة وتفحص العلم في البرنامج الرئيسي وتنفيذ جملة تعليمات تبعاً لحالة علم المقاطعة) وجميع المعالجَات تتم في البرنامج الرئيسي من أجل الاستجابة المباشرة للمقاطعات الأخرى حال حصولها.

8-5 زمن استجابة المقاطعة (Interrupt Response Time):

إن الاستجابة الزمنية عند تنفيذ المقاطعات بالنسبة لمتحكمات عائلة AVR هي على الأقل أربع دورات ساعة (4-Cycle) يتم خلالها دفع (Push) محتوى عداد البرنامج PC (Program Counter) إلى المكدس SP ويستهلك الدفع 2-Cycle، ومن ثم يقفز البرنامج إلى برنامج خدمة المقاطعة ويستهلك القفز 2-Cycle. وإذا حدثت المقاطعة أثناء تنفيذ إحدى التعليمات التي زمن تنفيذها أكبر من دورة واحدة، فإنه يتم استكمال تنفيذ التعليمة قبل الانتقال إلى برنامج خدمة المقاطعة. إن العودة من برنامج خدمة المقاطعة تستهلك أربع دورات ساعة (4-Cycle) أيضاً يتم خلالها سحب (Pull) قيمة عداد البرنامج PC من المكدس SP ويستهلك السحب 2-Cycle، ومن ثم يقفز إلى البرنامج الرئيسي ويستهلك القفز 2-Cycle، وينفذ ابتداءً من التعليمة التالية للتعليمة التي حدثت عندها المقاطعة.

9-5 المقاطعات الخارجية في متحكمات AVR (AVR MCU External Interrupts):

تمتلك متحكمات العائلة AVR أقطاب مخصصة للمقاطعات الخارجية والتي يرمز لها INT0, INT1, ….., INT7. الهدف من هذه المقاطعات الخارجية هو الاستجابة لأحداث معينة تطبق على أقطاب هذه المقاطعات. تملك هذه المقاطعات الخارجية أنماط استجابة متعددة للجبهات المطبقة عليها وكذلك يمكن تفعيلها أو إلغاء تفعيلها من خلال مجموعة من مسجلات التحكم الخاصة بهذه المقاطعات.

9-5-1 مسجلات التحكم بالمقاطعات الخارجية في متحكمات AVR (AVR MCUs External Interrupt Registers):

تملك المقاطعات الخارجية ثلاث مسجلات تحكم وهي:

- مسجل التحكم بنمط عمل المقاطعة الخارجية MCUCR (MCU Control Register).
- مسجل التحكم بالمقاطعات الخارجية GICR (Global Interrupt Mask Register).
- مسجل أعلام المقاطعات الخارجية GIFR (Global Interrupt Flag Register).

1) **مسجل التحكم بنمط عمل المقاطعة الخارجية MCUR**: يتم من خلاله التحكم بحساسية أو نمط استجابة المقاطعة للحدث الخارجي المطبق على قطب المقاطعة ويوجد أربع حالات وهي:
   1) تقدح عن الجبهة الصاعدة (Rising Edge).
   2) تقدح عن الجبهة الهابطة (Falling Edge).
   3) تقدح عن مستوى الجبهة (Low Level).
   4) تقدح عن تغير المستوى (Level Change).

Bit	7	6	5	4	3	2	1	0	
	SE	SM2	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	

الشكل5-6 مسجل التحكم بنمط عمل المقاطعة الخارجية **MCUR** للمقاطعات [INT0/INT1]

على اعتبار وجود أربعة حالات لنمط عمل كل مقاطعة من المقاطعات الخارجية فإن هذا سيحتاج إلى خانتين في مسجل التحكم بنمط عمل المقاطعة الخارجية لكل مقاطعة حيث تمثل الخانتين ISC01|ISC00 خانتي التحكم بنمط عمل المقاطعة الخارجية INT0، وتمثل الخانتين ISC11|ISC10 خانتي التحكم بنمط عمل المقاطعة الخارجية INT1 وهكذا باقي المقاطعات الأخرى.

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.

الشكل5-7 اختيار حالات نمط استجابة المقاطعة للحدث للمقاطعة INT1

2) **مسجل التحكم بالمقاطعات الخارجية GICR**: تمثل كل خانة من الخانات الثلاث (5,6,7) في المسجل خانة لتفعيل طلب مقاطعة خارجية من المقاطعات الخارجية الثلاثة (INT0, INT1, INT2) للمتحكم ATmega32A، حيث أنه عند وضع القيمة "**1**" في خانة المسجل **GICR**.n فإنه يتم تمكين المقاطعة الموافقة لهذه الخانة بشرط أن تكون الخانة I في مسجل الحالة SREG مفعلة ("1")، أما عند وضع "**0**" في خانة المسجل **GICR**.n فإنه يتم إلغاء تمكين المقاطعة الموافقة (n).

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	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	

الشكل5-8 مسجل التحكم المقاطعات الخارجية GICR

3) **مسجل أعلام المقاطعات الخارجية GIFR**: تمثل كل خانة من الخانات الثلاث (5,6,7) في المسجل علم يشير لحدوث مقاطعة خارجية (GIFR.n = "1") من المقاطعات الثلاث (INT0, INT1, INT2) للمتحكم ATmega32، وبالتالي سوف يقفز المتحكم إلى شعاع المقاطعة المتوضع عند العنوان المحدد في ذاكرة البرنامج لينفذ برنامج خدمة المقاطعة، وعند العودة من برنامج خدمة المقاطعة سيتم تصفير هذا العلم بشكل تلقائي من قبل الكيان الصلب.

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

الشكل5-9 مسجل أعلام المقاطعات الخارجية GIFR

10-5 برمجة المقاطعات الخارجية في BASCOM-AVR (Programming External Interrupts in BASCOM):

بشكل عام فإنه من أجل برمجة المقاطعات الخارجية فإنه يجب:

✓ تحديد نمط عمل (State: `Rising | Falling | Low | Level`) المقاطعة الخارجية (`INTx`).

```
Config INTx = State
```

✓ تحديد اسم البرنامج الفرعي (Label) للمقاطعة (`INTx`).

```
On INTx Label
```

✓ تفعيل (`Enable`) شعاع المقاطعة المطلوبة تشغيلها.

```
Enable INTx
```

✓ تفعيل شعاع المقاطعات العام.

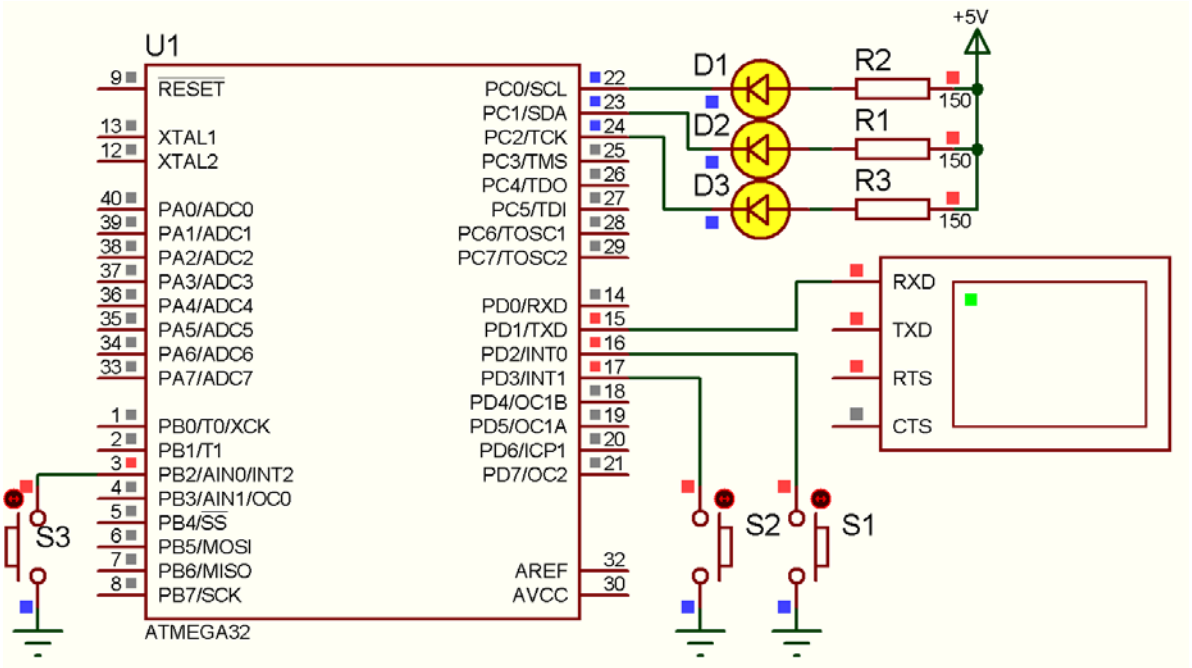
```
Enable Interrupts
```

ملاحظة1: يمكن أثناء عمل البرنامج إلغاء تفعيل أي من المقاطعات الخارجية من خلال التعليمة: `Disable INTx`

ملاحظة2: يمكن أثناء عمل البرنامج إلغاء تفعيل شعاع المقاطعات العام من خلال التعليمة: `Disable Interrupts`

ملاحظة3: التعليمة `Print` تستخدم لطباعة البيانات على النافذة التسلسلية (UART) في حال الوصل مع الحاسب.

التجربة الثانية عشرة: المطلوب تعديل التجربة الثامنة لتعمل المفاتيح اللحظية الثلاث (S1, S2, S3) الموصولة إلى أقطاب المقاطعات الخارجية INT2, INT1, INT0، على تغيير حالة عمل (Toggle) الثنائي D1 عن الضغط على المفتاح S1، وتغيير حالة D2 عند الضغط على S2، وتغيير حالة D3 عند الضغط على S3 – باستخدام المقاطعات الخارجية بدلاً من الفحص الدوري لحالة المفاتيح.



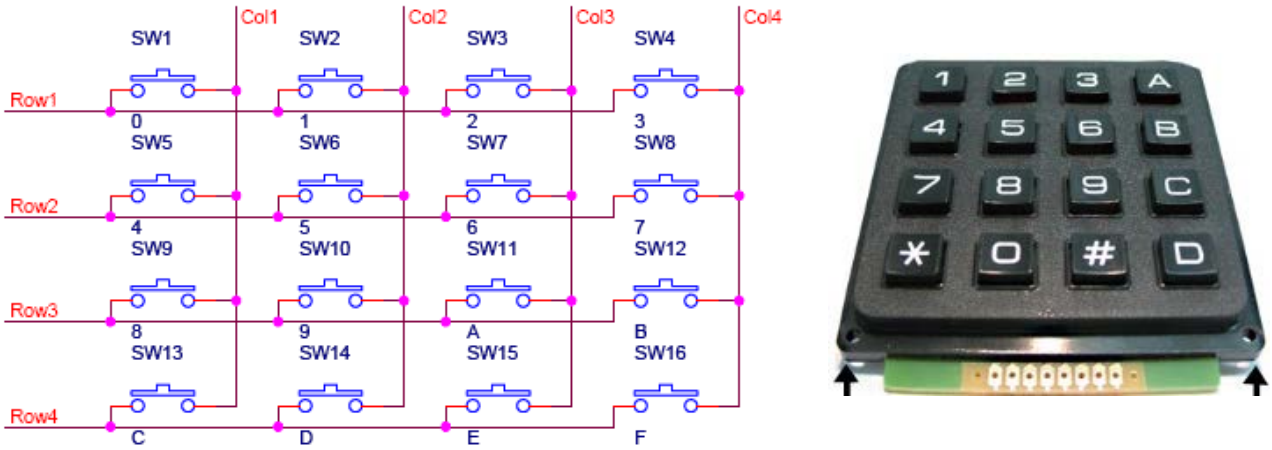
الشكل5-10 توصيل المفاتيح اللحظية الثنائيات مع المتحكم ATmega32A على اللوحة Mini-Phoenix للتجربة 12

البرنامج Exp.12.bas في بيئة BASCOM-AVR:

```
' ********************************************************************************
' * Title         : Exp.12.bas                                                   *
' * Target Board  : Mini-Phoenix - REV 1.00                                      *
' * Author        : Walid Balid                                                  *
' * IDE           : BASCOM AVR 2.0.7.3                                           *
' * Peripherals   : Pull-Up Resistors                                            *
' * Description   : External Interrupts                                          *
' ********************************************************************************
' Set the SW Jumbers to GND (Active Low)
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'-----------------------[Definitions]
$regfile = "m32def.dat"
$crystal = 8000000
$baud = 9600
'-----------------------
'-----------------------[GPIO Configurations]
Config Portc = &B00000111
Led1 Alias Portc.0 : Led2 Alias Portc.1 : Led3 Alias Portc.2
Set Led1 : Set Led2 : Set Led3
'-----------------------
'-----------------------[External Interrupts Configurations]
Config Int0 = Falling : On Int0 Sw_r1 : Enable Int0 : Portd.2 = 1 'PU Resistor
Config Int1 = Falling : On Int1 Sw_r2 : Enable Int1 : Portd.3 = 1
Config Int2 = Falling : On Int2 Sw_r3 : Enable Int2 : Portb.2 = 1

Enable Interrupts
'-----------------------[Variables]
Dim Count1 As Byte , Count2 As Byte , Count3 As Byte
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[Main Program]
Print "Hello!"
Do

Loop
End
'---<[End Main]
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[Print]
Sw_r1:
   Toggle Led1 : Count1 = Count1 + 1
   Print "Sw1 has Pressed! > " ; Count1
Return
'---<
Sw_r2:
   Toggle Led2 : Count2 = Count2 + 1
   Print "Sw2 has Pressed! > " ; Count2
Return
'---<
Sw_r3:
   Toggle Led3 : Count3 = Count3 + 1
   Print "Sw3 has Pressed! > " ; Count3
Return
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

11-5 توصيل وبرمجة لوحة مفاتيح مصفوفية مع متحكم AVR (Interfacing AVR MCU with Matrix-Keypad):

من أجل ربط عدد كبير من المفاتيح اللحظية مع متحكم مصغر فإنه ليس من المجدي ربط كل مفتاح إلى قطب كما مر معنا في التجارب السابقة لأن عدد الأقطاب المستهلكة من المتحكم ستساوي عدد المفاتيح التي تم ربطها مع تلك الأقطاب. لذلك يتم ربط المفاتيح مع بعضها بطريقة مصفوفية – أي يتم توصيل النقطة الأولى للمفاتيح المتوضعة على سطر واحد مع بعضها لتشكل قطب واحد يمثل السطر، كذلك يتم توصيل النقطة الثانية للمفاتيح المتوضعة على عمود واحد مع بعضها لتشكل قطب واحد يمثل العمود... وهكذا كما هو مبين على الشكل5-11.

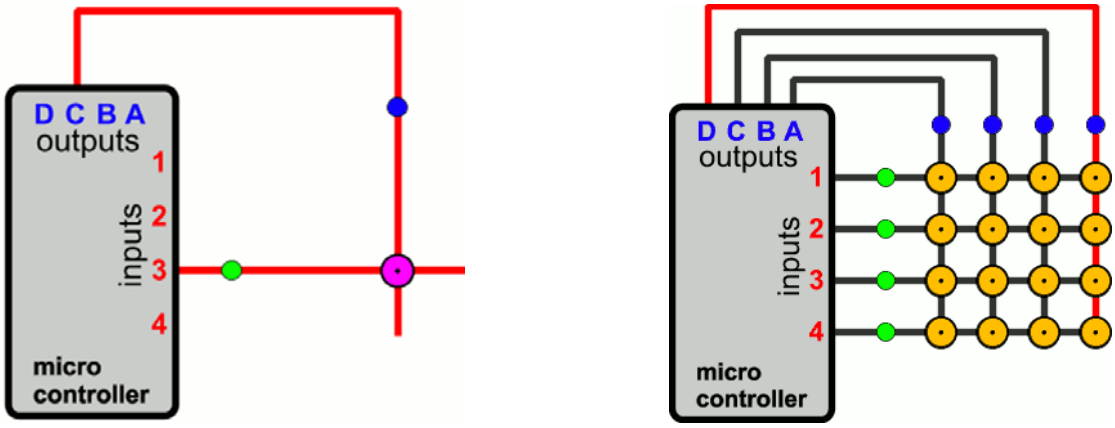


الشكل5-11 لوحة المفاتيح المصفوفية المؤلفة من 16-key

تستخدم لوحة المفاتيح المصفوفية بشكل أساسي في الهواتف، ويمكن أن تكون مؤلفة من 9 مفاتيح (3×3) أو 16 مفتاح (4×4) أو أكثر... ويكون دائماً عدد أقطاب توصيل اللوحة مساوياً إلى مجموع الأسطر والأعمدة (8-Lines < 4×4).

يتم توصيل لوحة المفاتيح مع أقطاب المتحكم المصغر مباشرة، ومنهجية مسح اللوحة لمعرفة المفتاح المضغوط تتم على الشكل التالي:

1) يجب وصل العمود الأول إلى القطب الأول من البوابة (مثلاً: PINB.0) والعمود الثاني إلى القطب الثاني وهكذا... ثم يتم توصيل السطر الأول إلى القطب التالي من نفس البوابة... ففي حال لوحة مفاتيح 4×4 فإن التوصيل سيكون كما هو مبين على الشكل5-13.
2) يتم تعريف أقطاب المتحكم الموصولة مع الأعمدة كأقطاب خرج، ويتم تعريف الأقطاب الموصولة مع الأسطر كأقطاب دخل.
3) يبدأ المسح بكتابة القيمة "1" على العمود الأول (على اعتبار أن الأقطاب الموصلة مع الأعمدة هي أقطاب خرج) وقراءة القيمة الظاهرة على الأسطر (على اعتبار أن الأقطاب الموصلة مع الأسطر هي أقطاب دخل). في حال لم يكن هناك أي مفتاح مضغوط فإن القيمة على الأسطر ستكون "0000". وفي حال كان هناك مفتاح مضغوط فإن السطر الذي ضغط فيه المفتاح ستظهر عليه القيمة المطبقة على العمود "1" وبالتالي يمكن معرفة المفتاح المضغوط. ثم ينتقل المسح إلى العمود الثاني ويكرر العملية السابقة ثم الثالث فالرابع وهكذا حتى يعود للعمود الأول ضمن دورة مسح لانهائية كما في الشكل5-12.



الشكل5-12 توصيل مجموعة المفاتيح مع المتحكم المصغر وحالة المسح

من أجل قراءة لوحة مفاتيح ست عشرية في البيئة BASCOM-AVR فإننا نحتاج إلى تعليميتين أساسيتين:

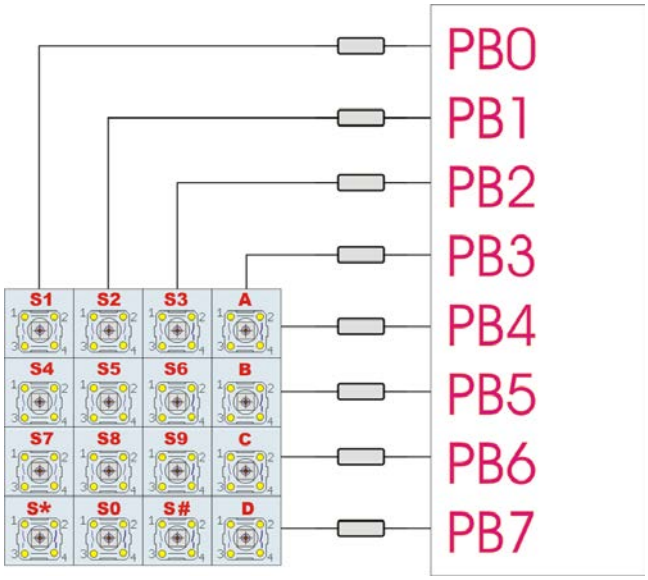
1) تعريف البوابة الموصول معها لوحة المفاتيح وتعريف زمن التأخير (**Debounce**) لتفادي أثر العطالة الميكانيكية للمفاتيح.

```
Config Kbd = Portb , Debounce = 100 , Delay = 100
```

2) قراءة حالة المفاتيح.

```
Var = Getkbd()
```

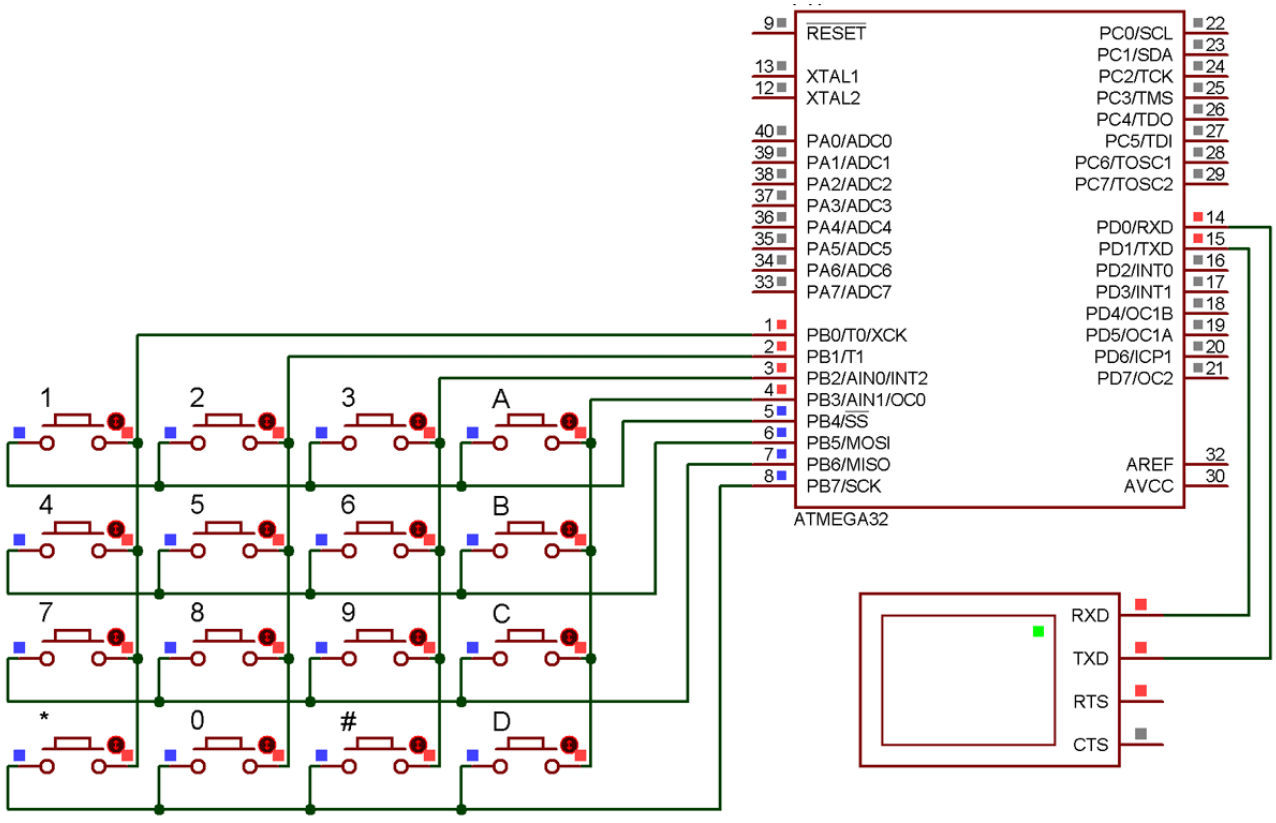
التابع "**Getkbd**" سيعود بقيمة عددية تتراوح بين **0 – 16** تمثل المفتاح المضغوط. حيث يعود هذا التابع بالقيمة **Var = 16** إذا لم يكن هناك أي مفتاح مضغوط. وأما إذا كان هناك مفتاح مضغوط فسيعود بقيمة المفتاح المضغوط كما يلي:




```
Var(S1) = 0
Var(S2) = 1
Var(S3) = 2
Var(A ) = 3
Var(S4) = 4
Var(S5) = 5
Var(S6) = 6
Var(B ) = 7
Var(S7) = 8
Var(S8) = 9
Var(S9) = 10
Var(C ) = 11
Var(S*) = 12
Var(S0) = 13
Var(S#) = 14
Var(D ) = 15
```

الشكل5-13 توصيل مجموعة المفاتيح مع المتحكم المصغر والقيم التي يعود بها التابع "**Getkbd**"

التجربة الثالثة عشرة: المطلوب بكتابة برنامج لقراءة حالة لوحة مفاتيح موصلة بشكل مصفوفي إلى البوابة PORTB لمعرفة المفتاح المضغوط وطباعة اسم المفتاح المضغوط على النافذة التسلسلية UART كما هو مبين على الشكل5-14 – ثم يطلب تطبيقها على اللوحة التعليمية مباشرة.



الشكل5-14 توصيل مجموعة من المفاتيح مع المتحكم ATmega32A لتشكيل لوحة مفاتيح 16-key/Keypad للتجربة 13

البرنامج Exp.13.bas في بيئة BASCOM-AVR:

```
' ************************************************************************
' * Title          : Exp.13.bas                                          *
' * Target Board   : Mini-Phoenix - REV 1.00                             *
' * Target MCU     : ATMega32A                                           *
' * Author         : Walid Balid                                         *
' * IDE            : BASCOM AVR 2.0.7.3                                  *
' * Peripherals    : Keypad                                              *
' * Description    : GPIOs as Input/Keypad                               *
' ************************************************************************
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'-----------------------[Definitions]
$regfile = "m32def.dat"
$crystal = 8000000
$baud = 9600
'-----------------------
'-----------------------[Keypad Configurations]
Config Kbd = Portb , Debounce = 100 , Delay = 100
'-----------------------
'-----------------------[Variables]
Dim Var As Byte
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
'--->[Main Program]
Do
   Var = Getkbd()
   If Var < 16 Then Gosub Check_number
Loop
End
'---<[End Main]
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[Print the Key Number]
Check_number:
   Select Case Var
      Case 00 : Print "Key Pressed is (1)"
      Case 01 : Print "Key Pressed is (2)"
      Case 02 : Print "Key Pressed is (3)"
      Case 03 : Print "Key Pressed is (A)"
      Case 04 : Print "Key Pressed is (4)"
      Case 05 : Print "Key Pressed is (5)"
      Case 06 : Print "Key Pressed is (6)"
      Case 07 : Print "Key Pressed is (B)"
      Case 08 : Print "Key Pressed is (7)"
      Case 09 : Print "Key Pressed is (8)"
      Case 10 : Print "Key Pressed is (9)"
      Case 11 : Print "Key Pressed is (C)"
      Case 12 : Print "Key Pressed is (*)"
      Case 13 : Print "Key Pressed is (0)"
      Case 14 : Print "Key Pressed is (#)"
      Case 15 : Print "Key Pressed is (D)"
   End Select
Return
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

. . . ﴿انتهت الجلسة العملية الخامسة﴾ . . .

وليد بليد

- دمتم بخير ومودة ونور -