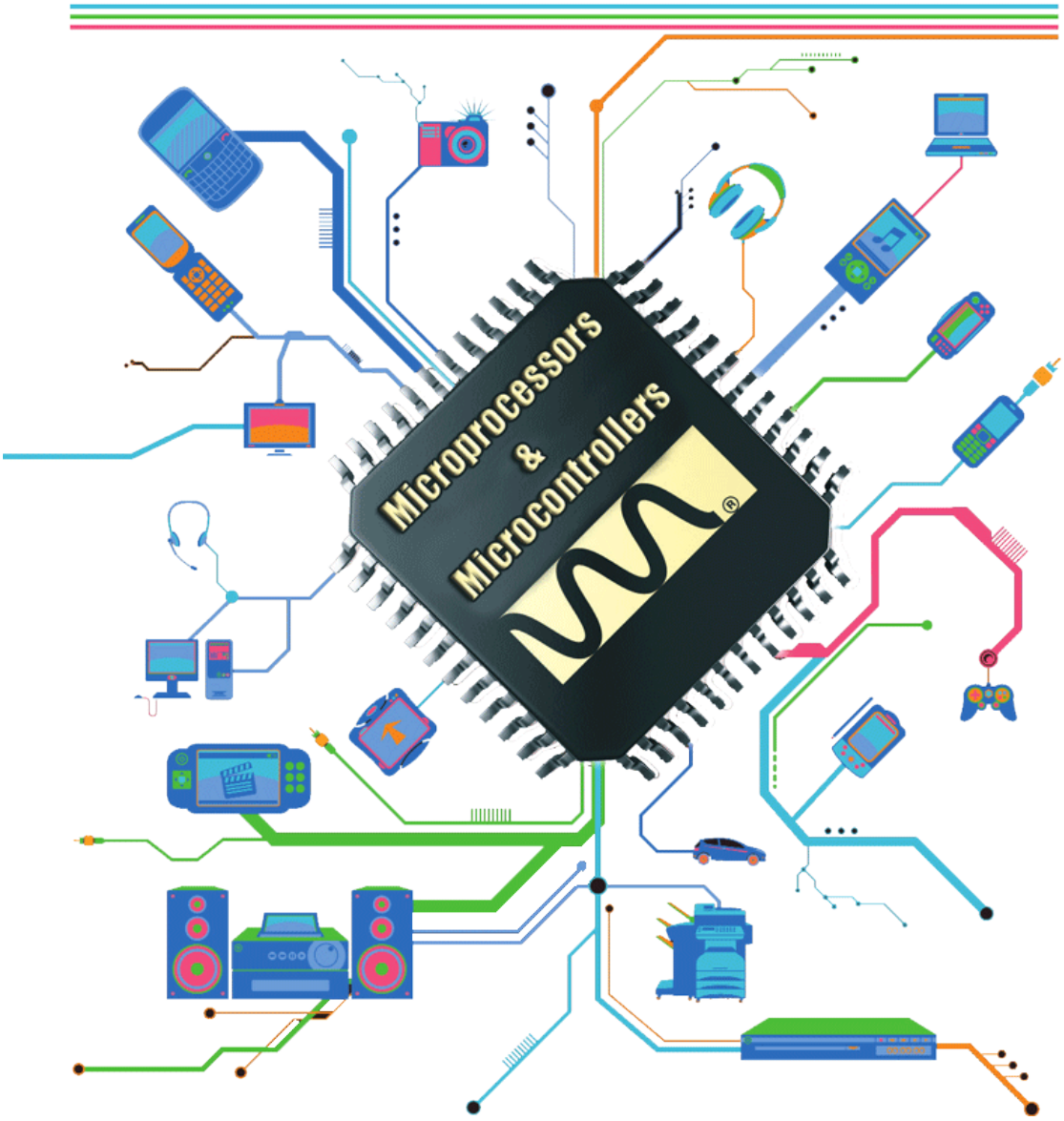
جامعـــــــــــة حلـــــــــــب
كلية الهندسة الكهربائية والإلكترونية
قسـم هندسـة التحكـم والأتمتـة

﴿ الجلسات العملية لمادة المعالجات والمتحكمات المصغرة ﴾

Microprocessors & Microcontrollers Lab Sessions

السنة الثالثة | قسم اتصالات

﴿ الجلسة العملية الثالثة ﴾



م. وليد بليد

Copyright © 2012 Walid Balid - All rights reserved.

Wednesday, March 14, 2012

الجلسة العملية الثالثة

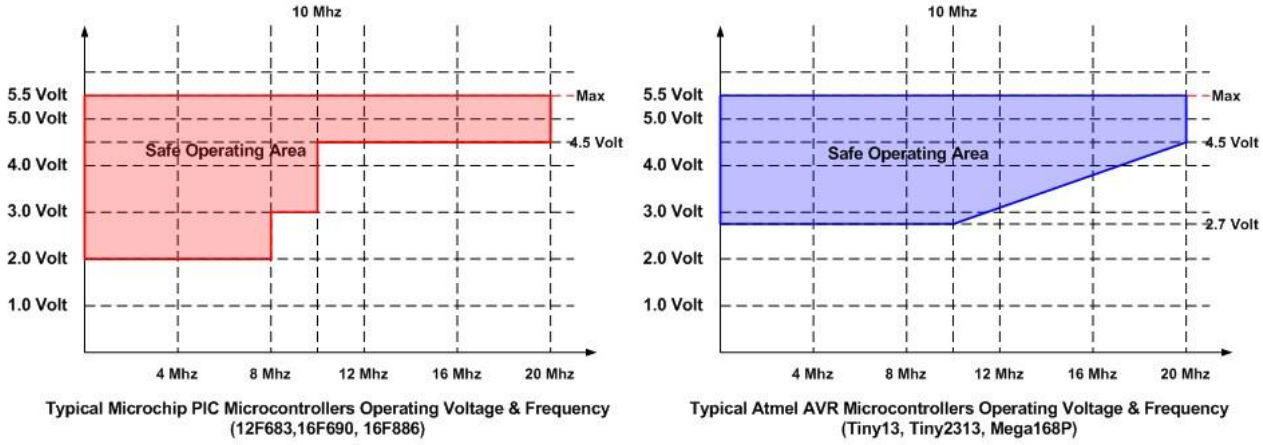
نظرة عامة (Overview):

هذه المحاضرة تشرح بنية بوابات الدخل والخرج لمتحكمات AVR وتشرح المسجلات الداخلية لبوابات الدخل والخرج. ثم تقدم تطبيقاً عملياً لاستثمار أقطاب الدخل والخرج لمتحكمات AVR. وبرمجتها في البيئة BASCOM-AVR ومحاكاتها في البيئة Proteus.

1-3 العلاقة بين تغذية المتحكم وتردد التشغيل الأعظمي (Power vs. Frequency):

بقدر ما تكون التغذية الرئيسية – لأي دارة إلكترونية – مصممة بشكل جيد وفق اعتبارات تصميمية قياسية، بقدر ما يكون عمل العناصر الإلكترونية في الدارة مستقراً وقريباً من منحني العمل الأمثلي. إن التغذية الكهربائية التي توصل للمتحكم المصغر هي بمثابة الروح التي تبث الحياة والحركة في المتحكم المصغر، كما أن استهلاك التغذية في المتحكم يتعلق مباشرة بسرعة عمل المتحكم المصغر، حيث أنه كلما ازداد تردد عمل المعالج، ازداد استهلاك التغذية في المعالج.

الشكل3-1 بين منحني العمل الآمن للمعالج نسبة إلى التغذية المطبقة من أجل كل تردد عمل. من أجل متحكم مصغر من العائلة "AVR" فإن التغذية 4.5V ستؤمن عمل آمن للمعالج عند كامل مجال تردد الهزاز الكريستالي، أما من أجل جهد تغذية "3V" فإن أقصى سرعة عمل للمتحكم يجب أن لا تزيد عن "8MHZ" لكي يبقى المعالج ضمن منطقة العمل الآمنة.



الشكل3-1 منحني العمل الآمن للمعالج نسبة إلى التغذية المطبقة من أجل كل كامل مجال تردد التشغيل

2-3 اعتبارات قيم التشغيل الأعظمية (Powering MCU & the Maximum Ratings):

أحد أهم الاعتبارات التي يجب أن تؤخذ بعين الاعتبار عن ربط أقطاب المتحكم إلى الأحمال هو التيار الأعظمي المستهلك من قطب المتحكم (Vcc-to-Gnd). إن قيمة التيار التي يمكن سحبها أو تصريفها لقطب دخل/خرج من أقطاب المتحكم تتراوح عادة من

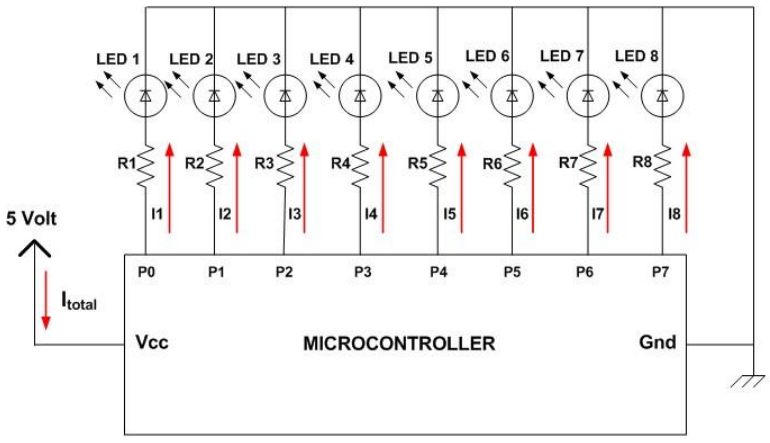
20~40mA حسب المواصفات الكهربائية للمتحكم المصغر. كما أن التيار الأعظمي الذي يمكن سحبه أو تصريفه عن طريق المتحكم بشكل كلي بالنسبة لمتحكمات AVR هو 200mA. الشكل3-2 يبين معدلات القيم الكهربائية الأعظمية لمتحكمات العائلة AVR.

Absolute Maximum Ratings

Operating Temperature	-55·C to +125·C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 - 400.0mA

الشكل3-2 معدلات قيم التشغيل الأعظمية لمتحكمات العائلة AVR

إن التيار الأعظمي الذي يمكن استجراره من المتحكم هو مجموع تيارات الأقطاب إضافةً إلى تيار التشغيل للمتحكم، وإن زيادة التيار فوق الحدود العظمى سوف يؤدي إلى عطل دائم في المتحكم ويتوجب بعدها تغييره. في الشكل3-3 تم استخدام ثمانية أقطاب من متحكم مصغر كأقطاب خرج لتشغيل ثنائيات ثمانية ضوئية.



Typical LEDs Display on the Microcontroller I/O Ports

الشكل3-3 توصيل ثنائيات ضوئية إلى أقطاب متحكم مصغر

إن التيار الأعظمي المسحوب من المتحكم هو مجموع تيارات الثنائيات الثمانية بالإضافة لتيار عمل المتحكم ويمكن حسابه بالشكل:

$$I_{total} = I_{operating_current} + (8 \times I_{LED})$$

بافتراض أن جهد عمل الثنائي الضوئي هو "2V" وقيمة المقاومة التسلسلية (مقاومة تحديد تيار عمل الثنائي الضوئي) هي "150Ω"، فيمكن حساب قيمة التيار المستجر من كل قطب من العلاقة التالية:

$$I_{LED} = V / R = (5 - 2) / 150 = \mathbf{20mA}$$

كما أن تيار عمل المتحكم من العائلة AVR في النمط الفعال هو 2.4mA، وبالتالي يمكن حساب التيار الكلي من العلاقة:

$$I_{total} = 2.5mA + 8 \times I_{LED} = 8 \times 20mA = \mathbf{162.5mA}$$

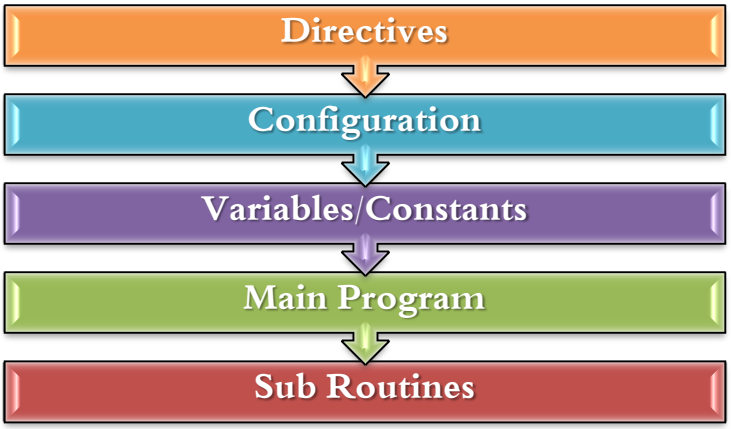
كما هو واضح فإن هذه القيمة تقترب من القيمة العظمة للتيار المسموح استجراره من متحكمات العائلة AVR والذي هو 200mA، بينما تفوق القيمة العظمة للتيار المسموح استجراره من متحكمات العائلة PIC والذي هو 90mA. وبالتالي فإن حساب التيارات المسحوبة من أقطاب المتحكم يعتبر من أهم الأمور التي يجب دراستها في بداية أي مشروع وهو ما سوف نناقشه فيما يأتي.

ملاحظة: عملياً ينصح بأن لا يتجاوز التيار المسحوب من المتحكم نصف قيمة التيار الأعظمي المسموح به لتخفيض ضجيج العمل وللتأكد من أن المتحكم قادر على تيار لعمل الأحمال الموصولة معه بشكل جيد.

3-3 تسلسل كتابة برنامج في Bascom-AVR (Writing a Scalable Code in Bascom-AVR):

من أجل كتابة كود برمجي متماسك ومفهوم مع إمكانية تطويره بسهولة مستقبلاً، فإنه يجب الالتزام بالهيكلية التالية في مراحل كتابته:

1) كتابة التوجيهات (Directives) المخصصة للمترجم (Compiler).
2) كتابة الإعدادات وأوامر التهيئة للمحيطيات (Configurations).
3) تعريف المتحولات (Variables) والثوابت (Constants).
4) كتابة حلقة البرنامج الرئيسي الدورية (Main Program).
5) كتابة البرنامج الفرعية (Sub-Routines).

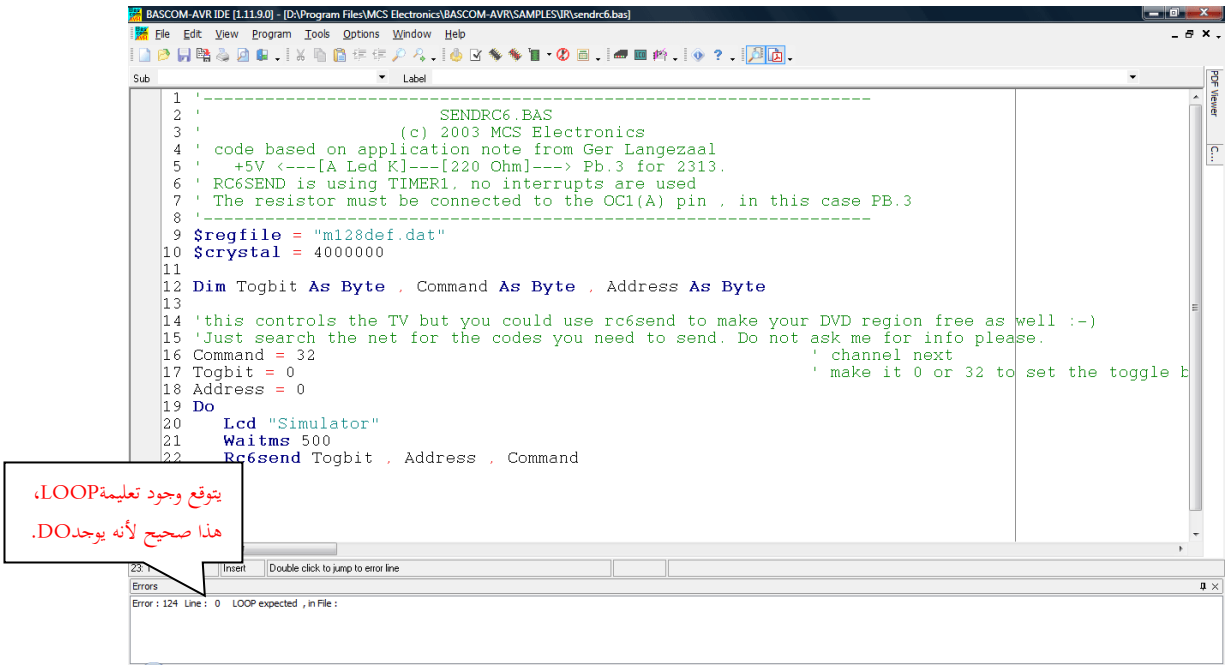


الشكل3-4 تسلسل كتابة برنامج لمتحكم مصغر

4-3 خطوات كتابة كود برمجي في البيئة Bascom-AVR (Writing a Code in Bascom-AVR):

1) من أجل كتابة كود برمجي قم باختيار New من القائمة File وابدأ بكتابة البرنامج وفق التسلسل المبين على الشكل3-4.

2) بعد الانتهاء من كتابة الكود البرمجي قم بحفظه في مجلد ثم قم باختيار أمر تفحص الأخطاء "Syntax Check" من القائمة Program. في حال وجود خطأ برمجي سوف تشير نافذة الأخطاء (أسفل الواجهة الرئيسية) إلى موقع الخطأ وسببه.



الشكل3-5 محرر التعليمات ونافذة تتبع الأخطاء في البيئة Bascom-AVR

3) بعد الانتهاء من تفحص الأخطاء، قم باختيار أمر الترجمة "Compile" من القائمة Program ليقوم البرنامج بتوليد الملفات البرمجية اللازمة للمبرمجة والتي سيتم توليدها في نفس المجلد، والملف الذي تحتاجه المبرمجة هو ذو امتداد "*.hex".

4) يمكن تشغيل نافذة المحاكاة الخاصة بالبيئة Bascom-AVR ومحاكاة التطبيق.

5) قم باختيار أمر الإرسال إلى المبرمجة (Send to programmer) من القائمة Program.

5-3 التعليمات الأساسية في Bascom-AVR (Essential Instructions in Bascom-AVR):

تعليمات التوجيهات الأساسية:

وظيفة التعليمة	شكل التعليمة
تحديد اسم المعالج المستخدم (ATmega128)	`$regfile = "m128def.dat"`
تحديد تردد المزاز الكريستالي الذي يعمل عليه المعالج	`$crystal = 1000000`
تحديد معدل بود النقل لنافذة الاتصال التسلسلي	`$baud = 9600`

تعليمات التأخير الزمني:

وظيفة التعليمة	شكل التعليمة
تأخير زمني (قيمة التأخير Value تعطى بالثانية)	**Wait** value
تأخير زمني (قيمة التأخير Value تعطى بالميلي ثانية)	**Waitms** value
تأخير زمني (قيمة التأخير Value تعطى بالميكرو ثانية)	**Waitus** value

تعليمات تعريف الأقطاب (دخل/خرج) ومقاومات الرفع الداخلية:

وظيفة التعليمة	شكل التعليمة
تعريف البوابة C كبوابة خرج	**Config** PORTC = **Output**
تعريف القطب رقم 5 من البوابة C كقطب خرج	**Config** PINC.5 = **Output**
تعريف البوابة C كبوابة دخل	**Config** PORTC = **Input**
تعريف القطب رقم 5 من البوابة C كقطب دخل	**Config** PINC.5 = **Input**
تفعيل مقاومات الرفع الداخلية للبوابة C	PORTC = 255
تفعيل مقاومة الرفع الداخلية للقطب رقم 5 من البوابة C	PINC.5 = 1
إلغاء تفعيل مقاومة الرفع الداخلية للقطب رقم 5 من البوابة C	PINC.5 = 0
تفعيل بعض مقاومات الرفع الداخلية للبوابة C	PORTC = &B11110000
يمكن استخدام هذا الشكل لتعريف الأقطاب من البوابة كدخل/خرج حيث أن (0) تعني قطب دخل، والـ (1) تعني قطب خرج.	**Config** PORTC = &B11110000
يصرح إلى أن البوابة (C) سوف يشار إليها أثناء البرنامج بالاسم (Leds)	Leds **Alias** PORTC
يصرح إلى أن القطب (5) سوف يشار إليه أثناء البرنامج بالاسم (Led)	Leds **Alias** PORTC.5

تعليمات التعامل على مستوى البت (Set/Reset):

وظيفة التعليمة	شكل التعليمة
جعل قيمة (البت/بت من متحول المتحول) واحد منطقي	**Set** bit
جعل قيمة (البت/بت من متحول المتحول) صفر منطقي	**Reset** bit
تغيير قيمة (البت/بت من متحول المتحول) إلى الحالة المعاكسة	**Toggle** bit

تعليمات الحلقات:

وظيفة التعليمة	شكل التعليمة
يستمر بالدوران في الحلقة وتنفيذ التعليمات الموجودة في جسم الحلقة حتى تحقق الشرط أو الخروج القسري من الحلقة.	**Do** Statements **Loop** [**until** Expression]

While Condition Statements **Wend**	تنفيذ جملة من التعليمات طالما أن الشرط محقق.
For Var = **Start To End** [**step** Value] Statements **Next** Var	تنفيذ جملة من التعليمات عدداً من المرات يبدأ من القيمة Start وينتهي عند القيمة End. يمكن تحديد خطوة العد بالمتحول step.
Exit For **Exit Do** **Exit While**	خروج قسري من الحلقة For خروج قسري من الحلقة Do خروج قسري من الحلقة While

التعليمات الشرطية:

شكل التعليمة	وظيفة التعليمة

If Expression1 **Then**
Statements1
...
Elseif Expression2 **Then**
Statements2
...
Else
Statements3
...
End If
	اختبار حالة أو قيمة متحول وتنفيذ تعليمات معينة تبعاً لنتيجة شروط الاختبار.
	إذا تحقق الشرط1 فنفذ التعليمات1
	وإلا إذا تحقق الشرط2 فنفذ التعليمات2
	وغير ذلك نفذ التعليمات3

SELECT CASE var **Case** Test1 : Statements1 **Case** Test2 : Statements2 **Case Else** : Statements3 **END SELECT**	اختبار حالة أو قيمة متحول وتنفيذ تعليمات معينة تبعاً لنتيجة شرط الاختبار المتحقق. إذا كان var = Test1 فنفذ التعليمات1 إذا كان var = Test2 فنفذ التعليمات2 وغير ذلك نفذ التعليمات3

تعليمات تعريف المتحولات في الذاكرة SRAM:

شكل التعليمة	وظيفة التعليمة
Dim Var1 **As Bit**	تعريف متحول عددي نوع بت (0 or 1).
Dim Var2 **As Byte**	تعريف متحول عددي نوع بايت (0 to 255).
Dim Var3 **As Integer**	تعريف متحول عددي صحيح (-32,768 to +32,767).
Dim Var4 **As Word**	تعريف متحول عددي نوع وورد (0 to 65535).
Dim Var5 **As Long**	تعريف متحول عددي طويل (-2147483648 to 2147483647).
Dim Var6 **As Single**	تعريف متحول عددي مؤشر (1.5 x 10^-45 to 3.4 x 10^38).
Dim Var7 **As Double**	تعريف متحول عددي مؤشر مضاعف.
Dim Var8 **As String** * 1	تعريف متحول نوع محرفي محدد المحارف بـ (chr_num *).

تعريف مصفوفة بثمان بايتات.	`Dim Array(8) As Byte`
تعريف متحول رقمي ثابت.	`Const Symbol = Numconst` `Ex. Const Pi = 3.14159265358979`
تعريف متحول محرفي ثابت.	`Const Symbol = Stringconst` `Ex. Const S = "TEST"`
تعريف تعبير رياضي ثابت.	`Const Symbol = Expression` `Ex. Const E =(b1 * 3) + 2`
تعريف متحول محلي في برنامج فرعي أو برنامج فرعي وظيفي.	`Local Var As Type`

تعليمات قراءة حالة مفاتيح موصولة مع أقطاب دخل:

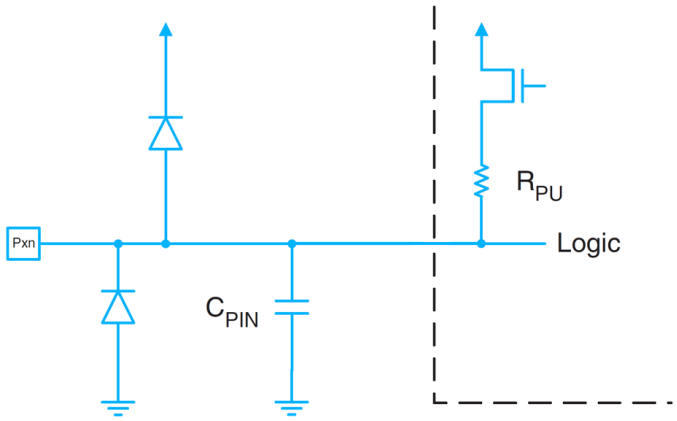
وظيفة التعليمة	شكل التعليمة
يراقب حالة القطب المحدد في Px.y كلما مر عليه وعندما تصبح حالته موافقة للحالة المحددة في state، سوف يقفز إلى البرنامج الفرعي عند اللافتة label وينفذ البرنامج ويعود.	`Debounce` $P_{x.y}$ `, state , label , Sub` `Ex. Debounce Key1 , 0 , Sw1 , Sub`
تهيئة زمن تأخير (ميلي ثانية) عن استعمال تعليمة Debounce للتخلص من العطالة الميكانيكية للمفتاح.	`Config Debounce = time`
سوف يقف البرنامج عند هذه التعليمة وينتظر أن تصبح حالة البت (القطب) صفر أو واحد منطقي عندها يكمل البرنامج.	`Bitwait x , Set/reset` `Ex. Bitwait Pinb.7 , reset`

يمكن الاطلاع على مبادى وأساسية البرمجة في البيئة Bascom من خلال ملف المساعدة (BASCOM-AVR IDE Help) وضمن "Language Fundamentals".

6-3 بوابات الدخل والخرج في متحكمات AVR (GPIOs in AVR MCUs):

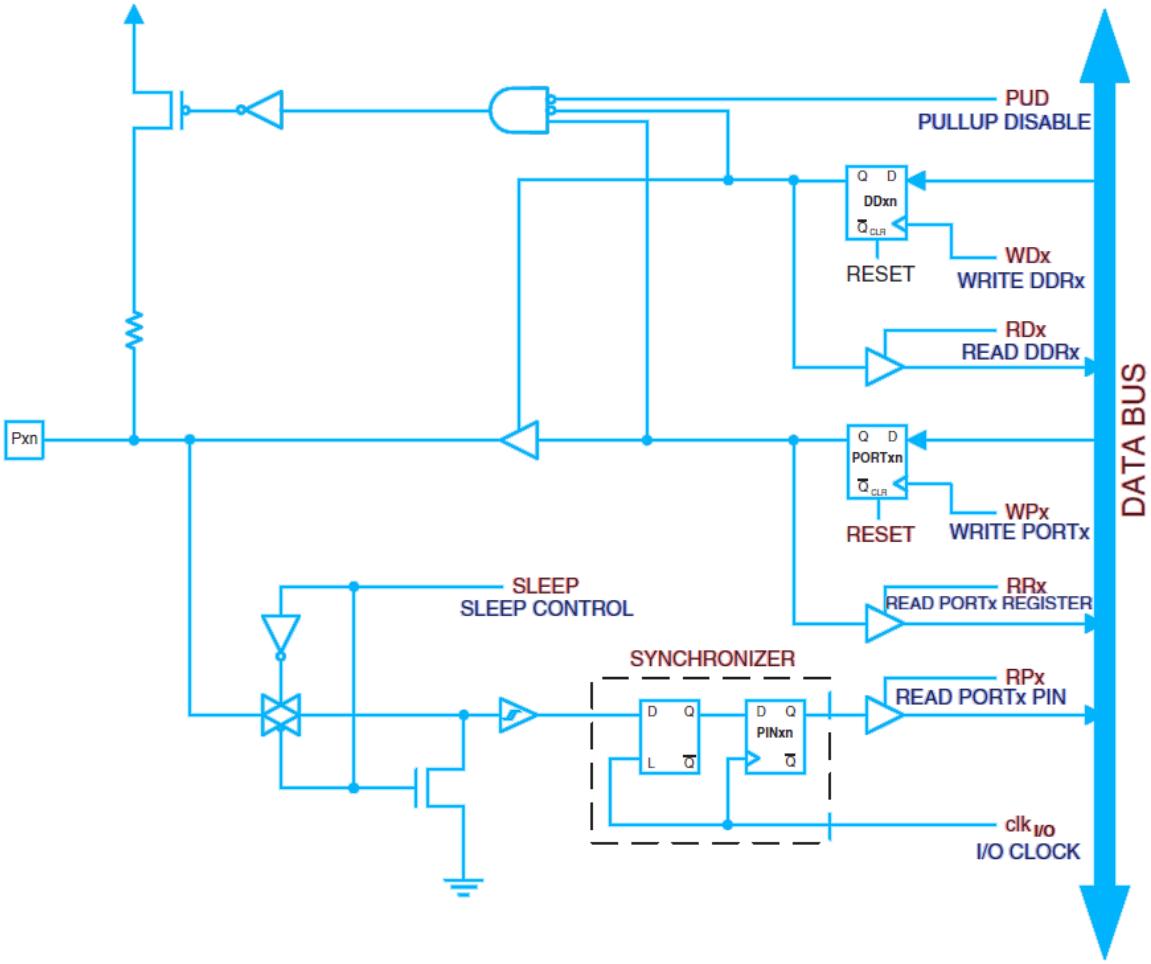
تتمتع جميع أقطاب بوابات متحكمات العائلة AVR بأنها أقطاب ثنائية الاتجاه وظائف قراءة وكتابة وتعديل عند استخدامها كأقطاب دخل/خرج للأغراض العامة (GPIOs)، كما يمكن تغيير اتجاه أحد أقطاب بوابة بشكل منفصل – خلافاً لمتحكمات 8051 – فيمكن تعريف كل قطب من الأقطاب على حدى كقطب دخل أو خرج. كذلك تمتلك الأقطاب عند تعريفها كأقطاب دخل مقاومات رفع داخلية – إلى التغذية – يمكن تفعيلها أو إلغاء تفعيلها لكل قطب بشكل منفصل.

إن بنية الأقطاب هي من النوع "Push-pull" أي أنها قادرة على قيادة الخرج على المستوى المنطقي "0" والمستوى "1" حيث أن التيار الذي يمكن أن يزوده القطب قادر على قيادة ثنائي ضوئي (LED) بشكل مباشر دون الحاجة إلى دارة مفتاح ترانزستوري. كما أن جميع الأقطاب مزودة بدارة حماية من تفريغ الشحنات الستاتيكية (ESD) مؤلفة من ثنائيين شوتكي أحدها موصل إلى التغذية (للحماية من شحنات التفريغ الموجبة) والآخر موصل إلى النقطة الأرضية (للحماية من شحنات التفريغ السالبة) كما هو مبين في الشكل3-6.



الشكل3-6 دارة الحماية من شحنات التفريغ الستاتيكية لقطب متحكم AVR

يتم تصنيف أقطاب الدخل/الخرج العامة (GPIOs) في مجموعات تسمى بوابات (PORTs) كل بوابة تتألف من ثمانية أقطاب (PINs)، ويختلف عدد البوابات باختلاف عدد أقطاب المتحكم حيث يمكن أن يصل عدد البوابات في متحكمات AVR المتقدمة إلى أحد عشر بوابة ويشار إليها بالأحرف **PORT**A, B, C, D, E, F, G, H, J, K, L.



الشكل3-7 البنية الداخلية الكاملة والدارات المنطقية لقطب متحكم AVR

3-7 مسجلات بوابات الدخل والخرج في متحكمات AVR (AVR MCUs GPIO Registers):

تمتلك كل بوابة من بوابات المتحكم ثلاث مسجلات تحكم، حيث يمثل الرمز **x** رمز البوابة (A, B, C, D, E, F, G, H, J, K, L) والرمز y قطب البوابة (7 : 0).

1) **مسجل التحكم باتجاه المعطيات للبوابة DDRx** (Data Direction Register): يتم من خلاله تحديد وظيفة كل قطب – دخل أو خرج. تمثل كل خانة من خانات مسجل اتجاه المعطيات الثمانية قطباً من أقطاب البوابة الموافقة (المسجلات أدناه هي للبوابة A)، حيث أنه عند وضع القيمة "**1**" في خانة المسجل DDRx.y فإن القطب الموافق لهذه الخانة يصبح قطب خرج، أما عند وضع القيمة "**0**" في خانة المسجل DDRx.y فإن القطب الموافق لهذه الخانة يصبح قطب دخل.

Bit	7	6	5	4	3	2	1	0
DDRA	**DDA7**	**DDA6**	**DDA5**	**DDA4**	**DDA3**	**DDA2**	**DDA1**	**DDA0**
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

الشكل3-8 مسجل اتجاه المعطيات للبوابة PORTA

2) **مسجل الخرج للمعطيات PORTx** (Data Output Register): يوجد له وظيفتان:

- في حال كان القطب معرفاً في مسجل DDRx كقطب خرج ("1" = DDRx.y)، فإن مسجل خرج المعطيات سيحدد الحالة المنطقية المطبقة على القطب بحيث إما أن يكون منبع للتيار ("1") أو مصرف للتيار ("0").
- في حال كان القطب معرفاً في مسجل DDRx كقطب دخل ("0" = DDRx.y)، فإن مسجل خرج المعطيات سيتحكم بوصل ("1") أو فصل ("0") مقاومة الرفع الداخلية للقطب المعني.

Bit	7	6	5	4	3	2	1	0
PORTA	**PORTA7**	**PORTA6**	**PORTA5**	**PORTA4**	**PORTA3**	**PORTA2**	**PORTA1**	**PORTA0**
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

الشكل3-9 مسجل خرج المعطيات للبوابة PORTA

3) **مسجل الدخل للمعطيات PINx** (Data Input Register): يستخدم لقراءة الحالة الخارجية المطبقة على القطب المعني ("0" | "1") عند تعريف القطب في مسجل اتجاه المعطيات كقطب دخل ("0" = DDRx.y).

Bit	7	6	5	4	3	2	1	0
PINA	**PINA7**	**PINA6**	**PINA5**	**PINA4**	**PINA3**	**PINA2**	**PINA1**	**PINA0**
Read/Write	R	R	R	R	R	R	R	R
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

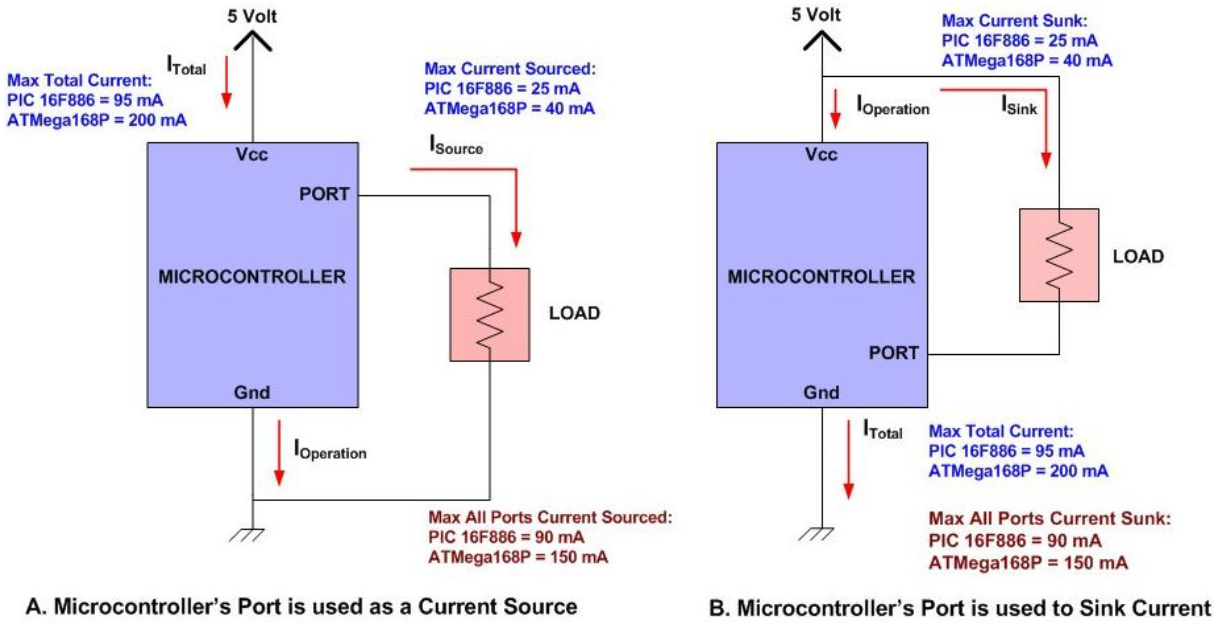
الشكل3-10 مسجل دخل المعطيات للبوابة PORTA

3-8 طرق توصيل الأحمال مع أقطاب المتحكم (Interfacing Loads with MCU):

إن بنية الأقطاب في متحكمات AVR ومتحكمات PIC هي من النوع "Push-pull"، أي أنها قادرة على قيادة الخرج على المستوى المنطقي "0" والمستوى "1" حيث أن التيار الذي يمكن أن يزوده القطب قادر على قيادة ثنائي ضوئي (LED) بشكل مباشر دون الحاجة إلى دارة مفتاح ترانزستوري. وبالتالي يمكن وصل الأحمال مع أقطاب المتحكم بطريقتين:

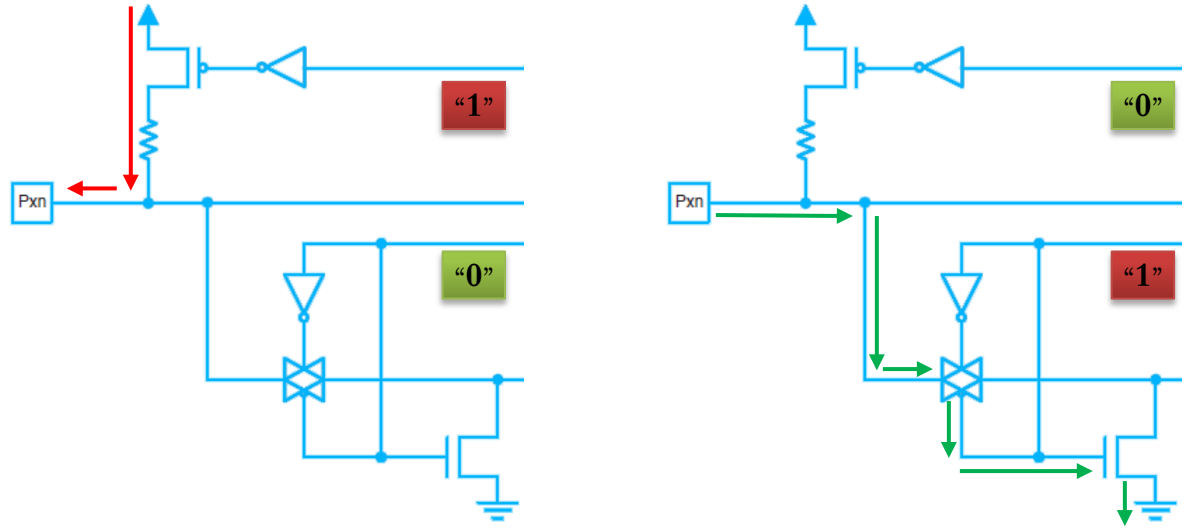
- القطب يعمل كمنبع لتيار تشغيل الحمل (Source) – الشكل3-11 (A).
- القطب يعمل كمصرف لتيار تشغيل الحمل (Sink) – الشكل3-11 (B).

في الحالة الأولى – القطب يعمل كمنبع (Source) لتيار تشغيل الحمل – يتم تزويد التغذية للحمل عن طريق التغذية الداخلية للمتحكم والتصريف يكون من خلال النقطة الأرضية مباشرة؛ أما في الحالة الثانية – القطب يعمل كمصرف (Sink) لتيار تشغيل الحمل – فإنه يتم تزويد التغذية للحمل مباشرة من التغذية الرئيسية ويتم التصريف من خلال المتحكم والنقطة الأرضية له. في كلا الحالتين سيكون الأداء للمتحكم واحداً إلا أنه يوصى عادة بالطريقة الثانية وذلك لتخفيض ضجيج التغذية VCC داخل المتحكم، كما أن توزع مسارات النقطة الأرضية GND داخل المتحكم أكبر وبالتالي التصريف سيكون موثوقاً ومناعته للضجيج أكبر.



الشكل3-11 طرق توصيل الأحمال مع أقطاب المتحكم – كمنبع أو مصرف للتيار

إن مبدأ سير التيار في الحالة الأولى (المسار باللون الأحمر على اليسار) والثانية (المسار باللون الأخضر على اليمين) داخل البنية الداخلية لأقطاب المتحكم مبين على الشكل3-12.

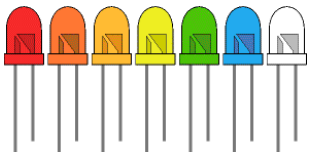


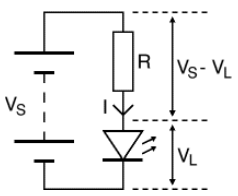
الشكل3-12 سير التيار داخل البنية الداخلية لقطب متحكم AVR كمصرف ومنبع للتيار

3-9 حساب قيمة واستطاعة مقاومة تحديد التيار (Calculating Current Resistor Value):

إن قيمة مقاومة تحديد التيار للحمل تتعلق مباشرة بجهد تشغيل الحمل وتياره ومقاومته الأمامية. من أجل حساب قيمة مقاومة تحديد التيار لثنائي ضوئي (LED) على سبيل المثال فإنه يجب معرفة تيار وجهد التشغيل للثنائي. إن تيار وجهد العمل للثنائيات الضوئية يختلف حسب لون الثنائي الضوئي، الجدول التالي يوضح المواصفات الكهربائية للثنائيات الضوئية.

Type	Colour	I_F max.	V_F typ.	V_F max.	V_R max.	Luminous intensity	Viewing angle	Wavelength
Standard	Red	20mA	2.0V	2.3V	5V	5mcd @ 10mA	60°	660nm
Super bright	Bright red	25mA	3.0V	3.4V	5V	80mcd @ 10mA	60°	625nm
Standard	Yellow	20mA	2.1V	2.3V	5V	32mcd @ 10mA	60°	590nm
Standard	Green	20mA	3.2V	3.5V	5V	32mcd @ 10mA	60°	565nm
High intensity	Blue	20mA	3.4V	3.6V	5V	60mcd @ 20mA	50°	430nm
Super bright	White	20mA	3.4V	3.6V	5V	500mcd @ 20mA	60°	660nm





IF max: التيار الأعظمي الأمامي المار في الثنائي.

VF typ: الجهد الأمامي النموذجي من اجل تشغيل الثنائي.

VF max: الجهد الأمامي الأعظمي الذي يمكن للثنائي أن يتحمله.

VR max: الجهد العكسي الأعظمي الذي يمكن للثنائي أن يتحمله.

Luminous intensity: شدة السطوع للثنائي.

Viewing angle: زاوية انعكاس الرؤية للإضاءة.

Wavelength: طول موجة الضوء الصادر.

وبالتالي من أجل ثنائي ضوئي ذو لون أحمر فإن جهد وتيار العمل هو 2V/20mA،

وبالتالي يمكن حساب مقاومة تحديد التيار من العلاقة:

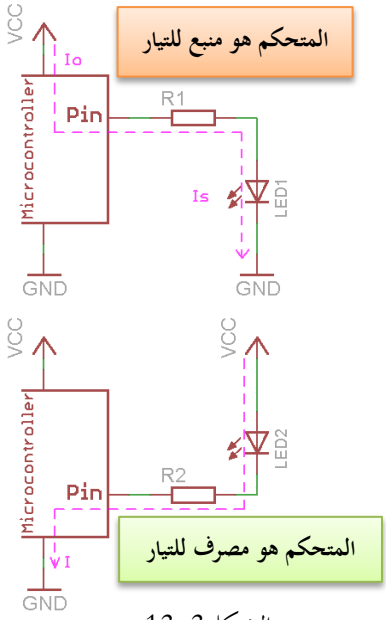
$$R_{LED} = \frac{V_{cc} - V_{LED}}{I_{LED}}$$

$$R_{LED} = \frac{5-2}{20} = \frac{3}{20} = 150\Omega$$

$$PR_{LED} = V_R \times I_R = (V_{cc} - V_{LED}) \times I_{LED}$$

$$PR_{LED} = (5-2) \times 20 = 60mW$$

وبالتالي فإن الذي نحتاجه هو مقاومة 150Ω ذات استطاعة $1/4_{Watt}$.

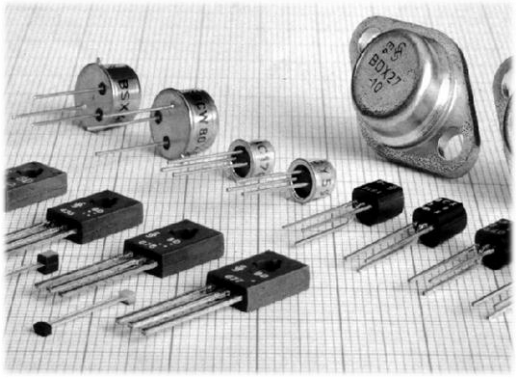


الشكل3-13

3-10 مفاتيح التحكم الترانزستورية (Transistors as Control Switches):

من أجل التحكم بأحمال ذات تيارات كبيرة (محركات، ريله، سخانات) فإن تيار الخرج لقطب المتحكم (20mA) لا يمكنه قيادة هذه الأحمال، لذا يتم استخدام الترانزستورات كمفاتيح إلكترونية (On/Off) للتحكم بهذه الأحمال. بشكل عام يوجد نوعين من الترانزستورات:

- الترانزستورات ثنائية القطبية (BJT).
- الترانزستورات أحادية القطبية (FET).



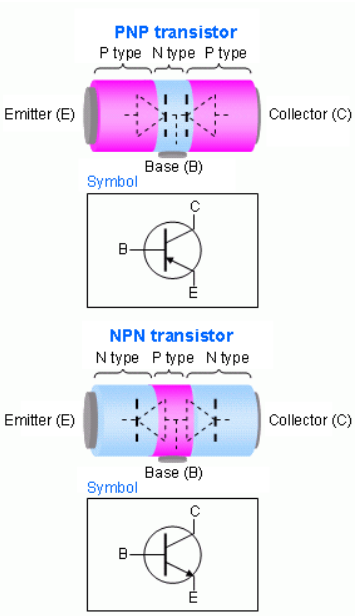
عملياً، إن الاستخدام لكل منها يختلف بحسب طبيعة الحمل المقاد، الجدول التالي يبين الفرق بين كلا النوعين:

	BJT	FET/MOSFET
طريقة التحكم	يتم التحكم به عن طريق تيار القاعدة ويحتاج تيار 1 – 10mA بالإضافة إلى VBE=0.6V	يتم التحكم به عن طريق جهد البوابة ويختلف الجهد حسب استطاعة الترانزستور.
سرعة الفتح والإغلاق	أبطئ لا يتجاوز 200MHZ (uS)	10 مرات أسرع (nS)
العمل	تأثر كبير بالحرارة	أقل تأثراً بالحرارة
المقاومة الأمامية	المقاومة الأمامية (هبوط جهد أمامي) صغيرة جداً	مقاومة أمامية كبيرة نسبياً
التأثر	لا يتأثر بالشحنات الساكنة	يمكن أن يتأثر ويدمر بالشحنات الساكنة
ممانعة الدخل	متوسطة	كبيرة جداً (10^{12} Ω)
مجالات جهود العمل	صغيرة لا تتجاوز 100V	كبيرة جداً
تيار الحمل	يعمل من أجل تيارات أحمال صغيرة	يمكنه أن يقود أحمال بتيارات عالية (محرك)
ضجيج العمل	ضجيج عالي	ضجيج منخفض

يتم استخدام الترانزستورات ثنائية القطبية من أجل التحكم بأحمال ذات تيارات صغيرة. بينما تستخدم الترانزستورات الحقلية من أجل التحكم بأحمال ذات تيارات وجهود متوسطة وكبيرة.

إن مجال استخدام الترانزستورات في أنظمة التحكم الرقمي يقتصر على استخدام هذه الترانزستورات كمفاتيح إلكترونية تحكمية (On/Off) – يعمل في منطقتي القطع والإشباع، وبالتالي فإن اختيار الترانزستور نسبة إلى الحمل سيعتمد على ثلاث عوامل أساسية:

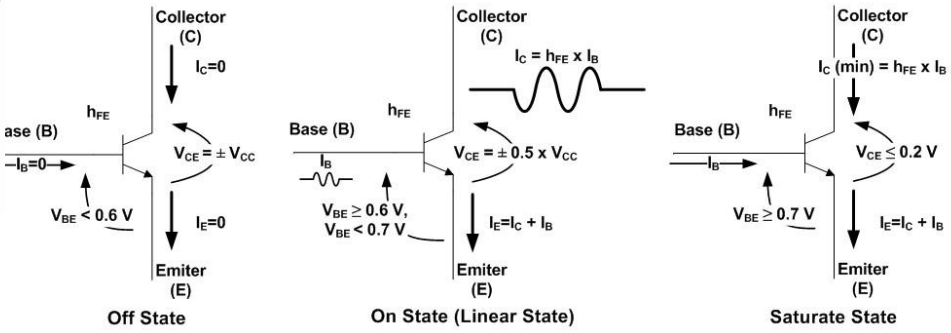
- التيار المار في الترانزستور.
- الاستطاعة المبددة في الترانزستور.
- سرعة الفتح والإغلاق للترانزستور.



في حالة القطع (Off state): يكون تيار القاعدة IB=0.

في الحالة الفعالة (On active state): يكون فيها تيار المجمع $\mathbf{I_C = I_B \times h_{FE}}$ وهي الحالة التي يستخدم فيها الترانزستور كمضخم فعال – أي زيادة في تيار القاعدة ينتج عنه زيادة في تيار المجمع.

في حالة الإشباع (On saturate state): في هذه الحالة يمرر الترانزستور كامل التيار.

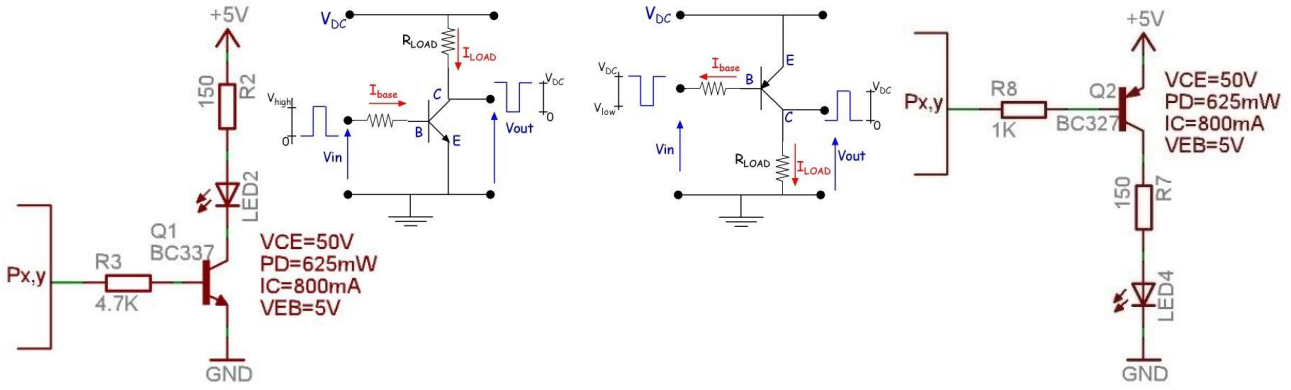


الشكل3-14 عمل الترانزستور وما يقابل كل حالة من شروط للجهد والتيار

3-11 استخدام مفاتيح التحكم الترانزستورية ثنائية القطبية (Using BJT Transistors as Control Switches):

يمكن توصيل المفاتيح الترانزستورية بطريقتين:

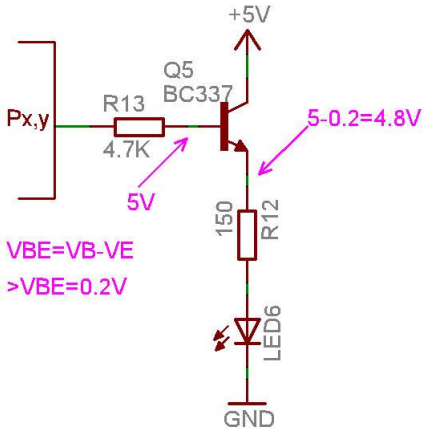
1) متحكم بها لتكون فعالة عند المنطق العالي "**1**": وبالتالي فإن الترانزستور سوف يعمل كمفتاح لوصل/فصل النقطة الأرضية (GND) للحمل، وفي هذه الحالة سوف يستخدم ترانزستور من نوع NPN – الشكل3-15.
2) متحكم بها لتكون فعالة عند المنطق المنخفض "**0**": وبالتالي فإن الترانزستور سوف يعمل كمفتاح لوصل/فصل نقطة التغذية (VCC) للحمل، وفي هذه الحالة سوف يستخدم ترانزستور من نوع PNP – الشكل3-16.



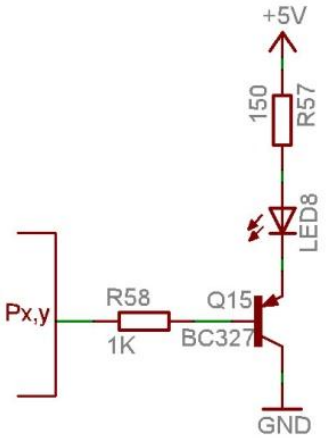
الشكل3-15 مفتاح ترانزستوري فعال عند المنطق "0"

الشكل3-16 مفتاح ترانزستوري فعال عند المنطق "0"

في بعض الأحيان يحصل خطأ في تصميم دارة المفتاح الإلكتروني باستخدام الترانزستور ثنائي القطبية، وهو من خلال استخدام الترانزستورات من نوع NPN كمفتاح لوصل/فصل نقطة التغذية (VCC) للحمل – الشكل3-17، أو استخدام الترانزستور من نوع PNP كمفتاح لوصل/فصل النقطة الأرضية (GND) للحمل – الشكل3-18.



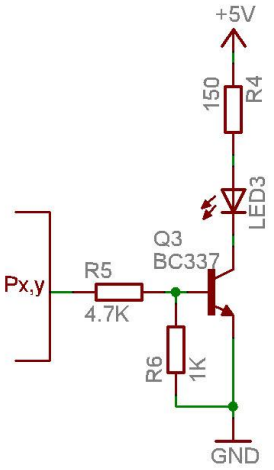
الشكل3-17 توصيل خاطئ لمفتاح ترانزستوري نوع NPN



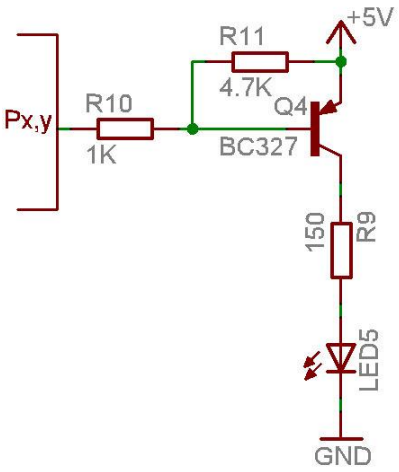
الشكل3-18 توصيل خاطئ لمفتاح ترانزستوري نوع PNP

لنوضح الخطأ من خلال الحسابات التالية: حتى يفتح الترانزستور بشكل كامل (الإشباع)، فيجب أن يكون الجهد $V_{BE}=0.7V$. بالنظر إلى الدارة على الشكل6-10 نجد أن الجهد الموجود على المشع (E) هو: $V_E= V_{CC}-V_{CE} = 5 - 0.2 = 4.8V$. كما أن الجهد على قاعدة الترانزستور هو: $V_B = V_{PIN} = 5V$. وبالتالي فإن: $V_{BE}= V_B - V_E = 5 - 4.8 = 0.2V$!! هذا يعني أن الترانزستور يعمل في المنطقة الفعالة ولن يكفي تيار مجمع الترانزستور (I_C) لتشغيل الحمل وسيعمل الثنائي الضوئي بشكل خافت.

إن الجهد المطبق على قاعدة الترانزستور في الدارة المبينة في الشكل3-15 والشكل3-16 يساوي 5V وهو نفسة جهد منطق بوابة المتحكم المصغر، بنفس الوقت من أجل الفتح الكامل للترانزستور فإنه يكفي تطبيق 0.7V، وإن هذا الجهد الزائد على القاعدة يؤدي إلى سحب تيار زائد وضياع في الاستطاعة، وبالتالي يمكن إضافة مقاومة مع مقاومة القاعدة ليتشكل لدينا مقسم كمون خرجه يتراوح بين 0.7~1V.
الشكل3-19 والشكل3-20 يوضحان طريقة إضافة المقاومة وحساب الجهد VBE.



الشكل3-20 دارة مفتاح ترانزستوري فعال عند المنطق "1"

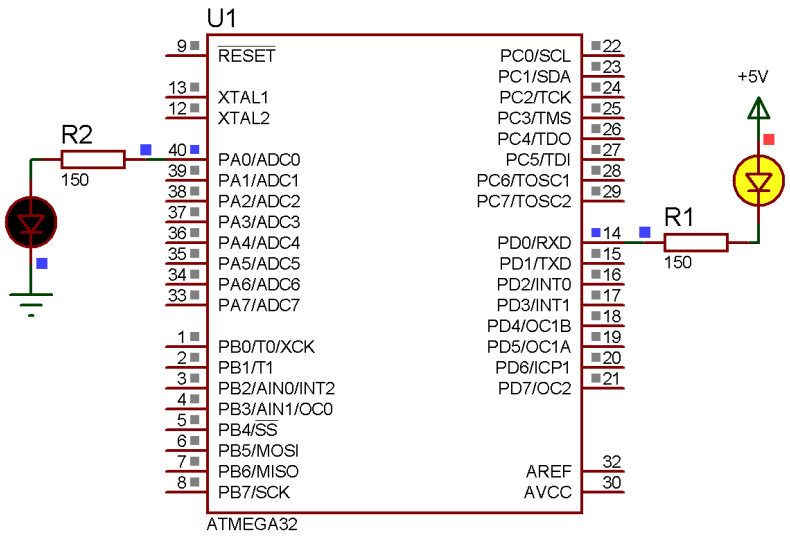


الشكل3-19 دارة مفتاح ترانزستوري فعال عند المنطق "0"

$$VBE = Vcc \times \frac{R6}{R6 + R5} = 5 \times \frac{1}{4.7 + 1} = 0.87V$$

12-3 برمجة بوابات الدخل والخرج في متحكمات AVR (Programming AVR MCUs GPIOs):

التجربة الأولى: تم وصل ثنائيين ضوئيين إلى متحكم ATmega32A، الثنائي الأول (LED1) موصول إلى القطب PINA.0 بحيث أن قطب المتحكم هو منبـع للتيار (فعال عند المستوى المنطقي "1") ، الثنائي الثاني (LED2) موصول إلى القطب PIND.0 بحيث أن قطب المتحكم هو مصرف للتيار (فعال عند المستوى المنطقي "0")، والمطلوب: كتابة برنامج خفقان بالتناوب لكلا الثنائيين كل 0.5S.

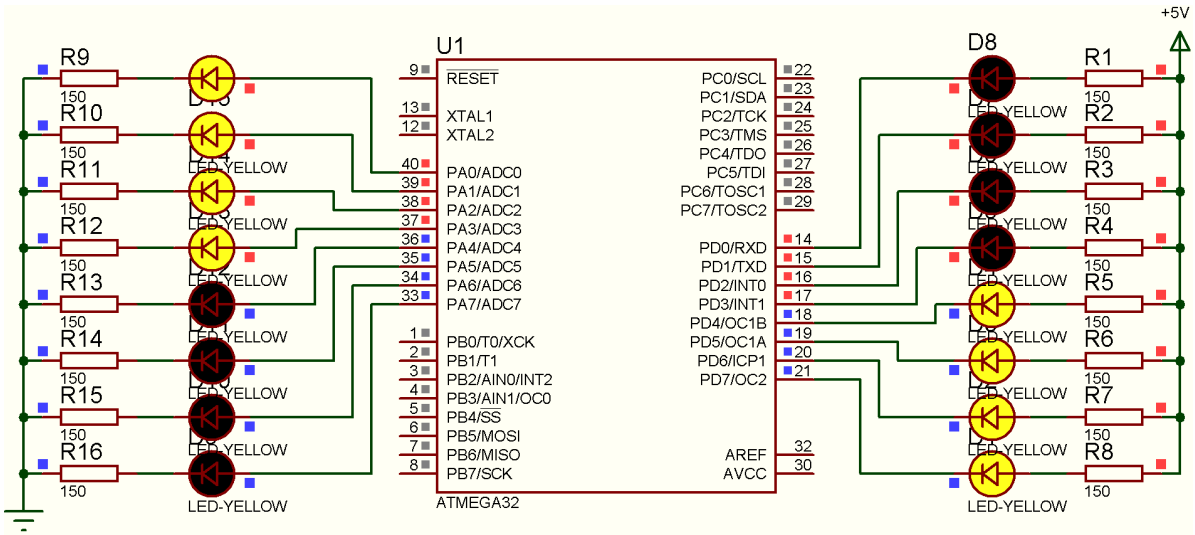


الشكل3-21 توصيل الثنائيات مع المتحكم للتجربة 1

البرنامج Exp.01.bas في بيئة BASCOM-AVR:

```
' ********************************************************************************
' * Title          : Exp.01.bas                                                  *
' * Target MCU     : ATMega32A                                                   *
' * Author         : Walid Balid                                                 *
' * IDE            : BASCOM AVR 2.0.7.3                                          *
' * Peripherals    : LEDs                                                        *
' * Description    : GPIOs as Outputs                                            *
' ********************************************************************************
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'-----------------------[Definitions]
$regfile = "m32def.dat"
$crystal = 8000000
'------------------------
'-----------------------[GPIO Configurations]
Config Pina.0 = Output : Led1 Alias Porta.0
Config Pind.0 = Output : Led2 Alias Portd.0
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[Main Program]
Do
    '>[Turn Led1 on & Led1 off]
   Set Led1 : Set Led2 : Waitms 500
    '>[Turn Led1 off & Led2 on]
   Reset Led1 : Reset Led2 : Waitms 500
Loop
End
'---<[End Main]
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

التجربة الثانية: تم وصل ثمانية ثنائيات ضوئية (LEDs_A) إلى البوابة PORTA للمتحكم ATmega32A بحيث أن بوابة المتحكم هي في حالة منبـع للتيار، وتم وصل ثمانية ثنائيات ضوئية أخرى (LEDs_D) إلى البوابة PORTD بحيث أن بوابة المتحكم هي في حالة مصرف للتيار، والمطلوب: كتابة برنامج بحيث تعمل الثنائيات بالتتابع ذهاباً وإياباً باستخدام الحلقات.

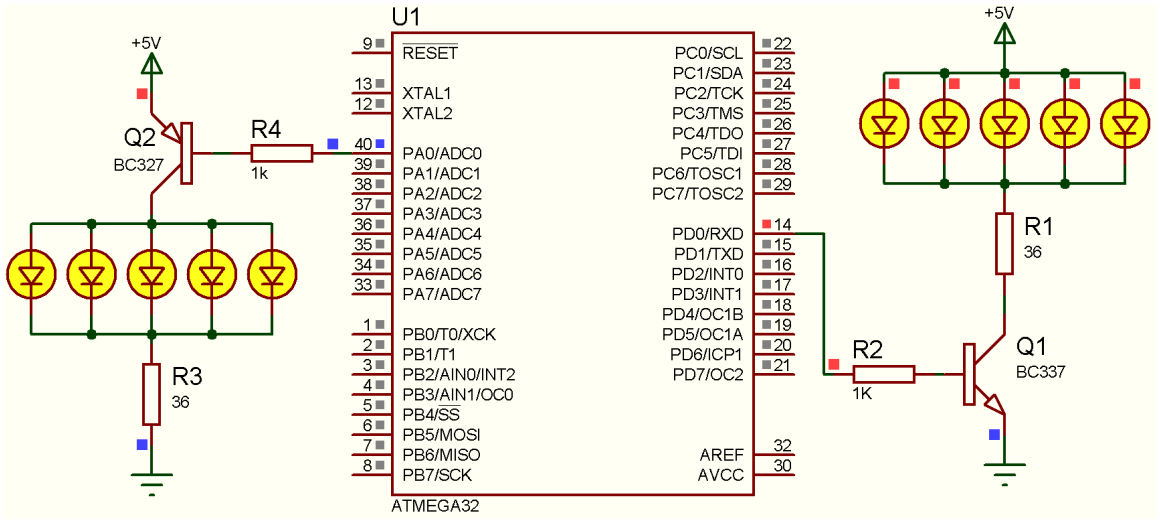


الشكل3-22 توصيل الثنائيات مع المتحكم للتجربة 2

البرنامج Exp.02.bas في بيئة BASCOM-AVR:

```
' ************************************************************************************
' * Title          : Exp.02.bas                                                      *
' * Target MCU     : ATMega32A                                                       *
' * Author         : Walid Balid                                                     *
' * IDE            : BASCOM AVR 2.0.7.3                                              *
' * Peripherals    : LEDs                                                            *
' * Description    : GPIOs as Outputs                                                *
' ************************************************************************************
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'-----------------------[Definitions]
$regfile = "m32def.dat"
$crystal = 8000000
'-----------------------
'-----------------------[GPIO Configurations]
Config Porta = Output : Leds_a Alias Porta : Leds_a = 0      'Set All LEDs off
Config Portd = Output : Leds_d Alias Portd : Leds_d = 0      'Set All LEDs on
'-----------------------
'-----------------------[Variables]
Dim I As Byte
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[Main Program]
Do
   Gosub Leds_fw : Gosub Leds_rw
Loop
End
'---<[End Main]
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[To Turn-on LEDs_A  & Turn-off LEDs_D]
Leds_fw:
   For I = 0 To 7
      Set Leds_a.i : Set Leds_d.i
      Waitms 100
   Next I
Return
'-----------------------
'--->[To Turn-off LEDs_A  & Turn-on LEDs_D]
Leds_rw:
   For I = 7 To 0 Step -1
      Reset Leds_a.i : Reset Leds_d.i
      Waitms 100
   Next I
Return
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

التجربة الثالثة: تم وصل مجموعة من خمسة ثنائيات ضوئية (LEDs_A) إلى متحكم ATmega32A مع القطب PINA.0 عن طريق مفتاح ترانزستوري وبحيث أن الترانزستور هو من النوع PNP (فعال عند المستوى المنطقي "0")، كما تم وصل خمسة ثنائيات ضوئية أخرى (LEDs_B) مع القطب PIND.0 عن طريق مفتاح ترانزستوري وبحيث أن الترانزستور هو من النوع NPN (فعال عند المستوى المنطقي "1")، والمطلوب: كتابة برنامج خفقان بالتناوب لكلا المجموعتين كل 0.5S.



الشكل3-23 توصيل الثنائيات مع المتحكم للتجربة 3

البرنامج Exp.03.bas في بيئة BASCOM-AVR:

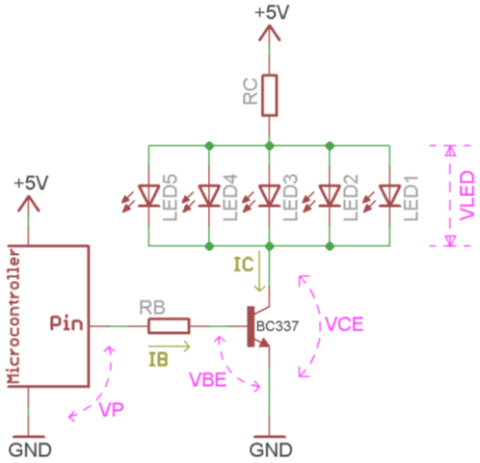
```
' ********************************************************************************
' * Title          : Exp.03.bas                                                  *
' * Target MCU     : ATMega32A                                                   *
' * Author         : Walid Balid                                                 *
' * IDE            : BASCOM AVR 2.0.7.3                                          *
' * Peripherals    : LEDs                                                        *
' * Description    : GPIOs as Outputs                                            *
' ********************************************************************************
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'-----------------------[Definitions]
$regfile = "m32def.dat"
$crystal = 8000000
'-----------------------
'-----------------------[GPIO Configurations]
Config Pina.0 = Output : Leds_a Alias Porta.0
Config Pind.0 = Output : Leds_d Alias Portd.0
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'--->[Main Program]
Do
   '>[Turn-on LEDs_A & Turn-off LEDs_D]
   Set Leds_a : Set Leds_d : Waitms 500
   '>[Turn-on LEDs_A & Turn-off LEDs_D]
   Reset Leds_a : Reset Leds_d : Waitms 500
Loop
End
'---<[End Main]
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

3-13 تصميم دارة مفتاح ترانزستوري ثنائي القطبية (BJT Control Switch Circuit Design):

لنأخذ مثالاً عملياً ونحسب قيم المقاومات والتيارات للدارة المبينة في الشكل3-24 والتي تستخدم في التجربة السابقة (Exp.03).

لتصميم دراة المفتاح قمنا باختيار الترانزستور BC337 والذي له المواصفات التالية:

I_{C_max}=800mA, $V_{BE_saturate}$=0.65V, $V_{CE_saturate}$=0.2V, h_{FE} = 100, V_{CE_max}=50V



الشكل3-24 التحكم بمجموعة ثنائيات من قطب متحكم باستخدام مفتاح ترانزستوري BJT

1) نحسب تيار الحمل (IC) مع العلم أن تيار كل ثنائي ضوئي هـو: I_{LED}= 20mA.

I_C = 5 x 20mA = 100mA

2) نحسب مقاومة تحديد التيار مع العلم أن جهد عمل الثنائي هو: V_{LED}=2V.

$$Rc = \frac{V_{cc} - V_{LED}}{I_C} = \frac{5-2}{100} = 30\Omega$$

3) نحسب استطاعة مقاومة تحديد التيار.

$$P_{RC} = (V_{cc} - V_{LED}) \times I_C = (5-2) \times 100 = 300mW$$

4) نحسب قيمة التيار الأصغري اللازم لقيادة الترانزستور عن طريق بوابة المتحكم:

$$I_C = hfe \times I_B \rightarrow I_B = \frac{I_C}{hfe} = \frac{100}{100} = 1mA$$

5) نحسب الاستطاعة المبددة في الترانزستور:

$$Pc_{max} = U_{CE} \times I_C = 0.2 \times 100 = 20mW$$

6) نحسب قيمة مقاومة القاعدة واستطاعتها:

$$R_B = \frac{V_P - V_{BE}}{I_B} = \frac{5-0.7}{1} = 4.3K\Omega$$

$$P_{RC} = (V_P - V_{BE}) \times I_B = (5-0.7) \times 1 = 4.3mW$$

. . . ﴿انتهت الجلسة العملية الثانية﴾ . . .

وليد بليد

- دمتم بخير ومودة ونور -