الباب الأول

مقدمة عامة عن المتحكمات الدقيقة

المتحكم الدقيق :

هو عبارة عن جهاز كمبيوتر صغير مصمم خصيصا ليقوم بأعمال معينة ، ويستخدم الذاكرة لتخزين الأوامر المبرمجة و القيام بتنفيذ هذه الأوامر مثل : (التشغيل و الاطفاء ، العد ، الحساب ، التوقيت و غيرها من العمليات) .

أنواع المتحكمات الدقيقة :

الشركة المصنعة	العائلـــــة
Atmel	AVR
Microchip	PIC
Motorolla	HS908
National	PSOC
Intel	8051, 8052, 8096

استخدامات الميكروكنترولر :

طابعة الليزر

الربوتات .

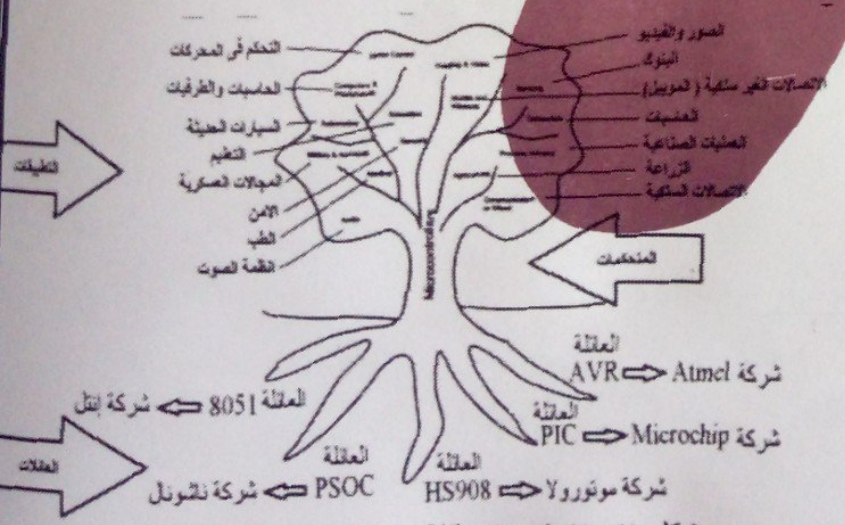
المساحات الضوئية .

الغسالة الأوتوماتيك .

أجهزة التكييف .

أجهزة الارسال والاستقبال اللاسلكي .

تطبيقات التحكم الحركي في سرعة واتجاه وعزم دوران المحركات .



شكل (1- 3) شجرة عائلات وتطبيقات الميكروكونترولر

الخصائص العامة للميكروكنترولر :

١- يكون الميكروكنترولر عادة بداخل جهاز آخر لتحكم بذلك الجهاز .

٢- يكون في الميكروكنترولر ما يحتاجه من الذاكرة مثل الروم و الرام فهو ليس بحاجة الى شرائح ذاكرة خارجية .

٣- يكون عمل الميكروكنترولر محدد بمهمة واحدة وتنفيذ الأوامر في برنامج واحد يكون مخزنا فى ذاكرة المايكروكنترولر .

٤- يستهلك طاقة صغيرة جدا مقارنه بالكمبيوتر ، فبعض المتحكمات تستهلك 50 مللى وات بينما يستهلك الكمبيوتر 50 وات .

٥- زمن تنفيذ التعليمة ثابت ومقداره دورة واحدة من دورات الآلة باستثناء تعليمات التفرع التى تستخدم دورتان .

٥- كل تعليمة تمثل موقع من ذاكرة البرنامج الوميضية .

٦- دورة الآلة هى أربع دورات للساعة .

العوامل التى تحدد اختيار الميكروكنترولر :

١_ عدد أطراف الدخل و الخرج المطلوبة In/ out pins .

٢_ الملحقات المطلوبة مثل (Motor control , LCD , USB) .

٣_ حجم ذاكرة البرنامج .

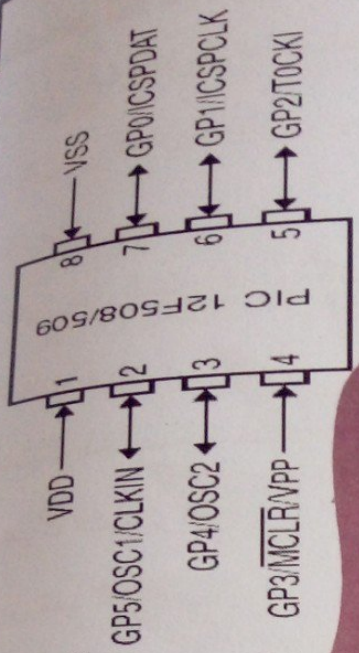
٤_ حجم ذاكرة الوصول العشوائى .

٥_ نوع الذاكرة ما اذا كانت وميضية أو ثابتة .

٦_ السرعة .

٧_ التكلفة .

عائلات الميكروكنترولر (PIC) طبقا لطول التعليمة .



١ – **عائلة 12 بت** : وأهم ما يميزها :

أ – النوع الأكثر استخداما هو PIC 12F508 ذو ثمانية أطراف

عدد منافذ الدخل / الخرج منفذ واحد G و له ستة أطراف Gp 0 : Gp 5

ب _ مجموعة التعليمات 33 تعليمة .

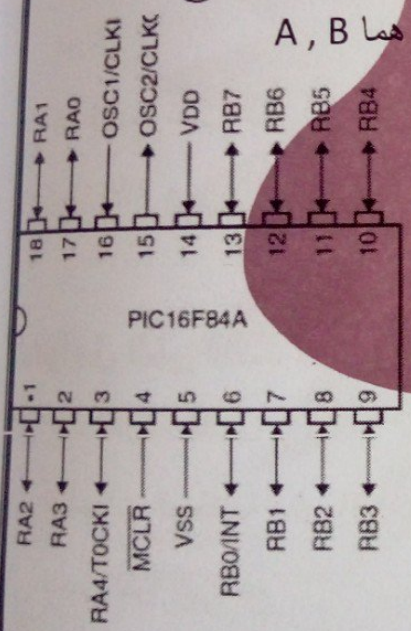
جـ _ مذبذب الساعة الداخلي تردده 4 M Hz .

ج _ سعة ذاكرة البرنامج 512 x 12 من نوع EPROM .

د _ سعة ذاكرة البيانات 25 bytes من نوع RAM .

هـ _ يحتوي على مؤقت 8 bit ، رخيصة الثمن .

٢ – **عائلة 14 بت** : وأهم ما يميزها :



أ – النوع الأكثر استخداما هو PIC 16f84 عدد منافذ دخل / خرج منفذان و هما A , B

منفذ A له خمسة أطراف $A_0 : A_4$ و منفذ B له ثمانية أطراف $B_0 : B_7$

و بذلك يحتوي على 13 أطرف دخل / خرج .

ب – مجموعة التعليمات 35 تعليمية .

ج – مذبذب الساعة الداخلي تردده 10 M Hz .

د – سعة ذاكرة البرنامج 14 x 1024 bytes من النوع الوميضي flash

هـ – سعة ذاكرة البيانات 64 bytes من نوع EEPROM الغير متطايرة.

و – يحتوي على مؤقت 8 bit و مؤقت للحراسة و مؤقت للمقاطعة .

ز – كل الأوامر دورة واحدة ما عدا أوامر القفز و التفرع فهي دورتان .

٣ – **عائلة 16 بت** : وأهم ما يميزها :

أ – النوع الأكثر استخداما هو 18C452 عدد المداخل / المخارج 34 طرف .

ب _ مجموعة التعليمات 77 تعليمة .

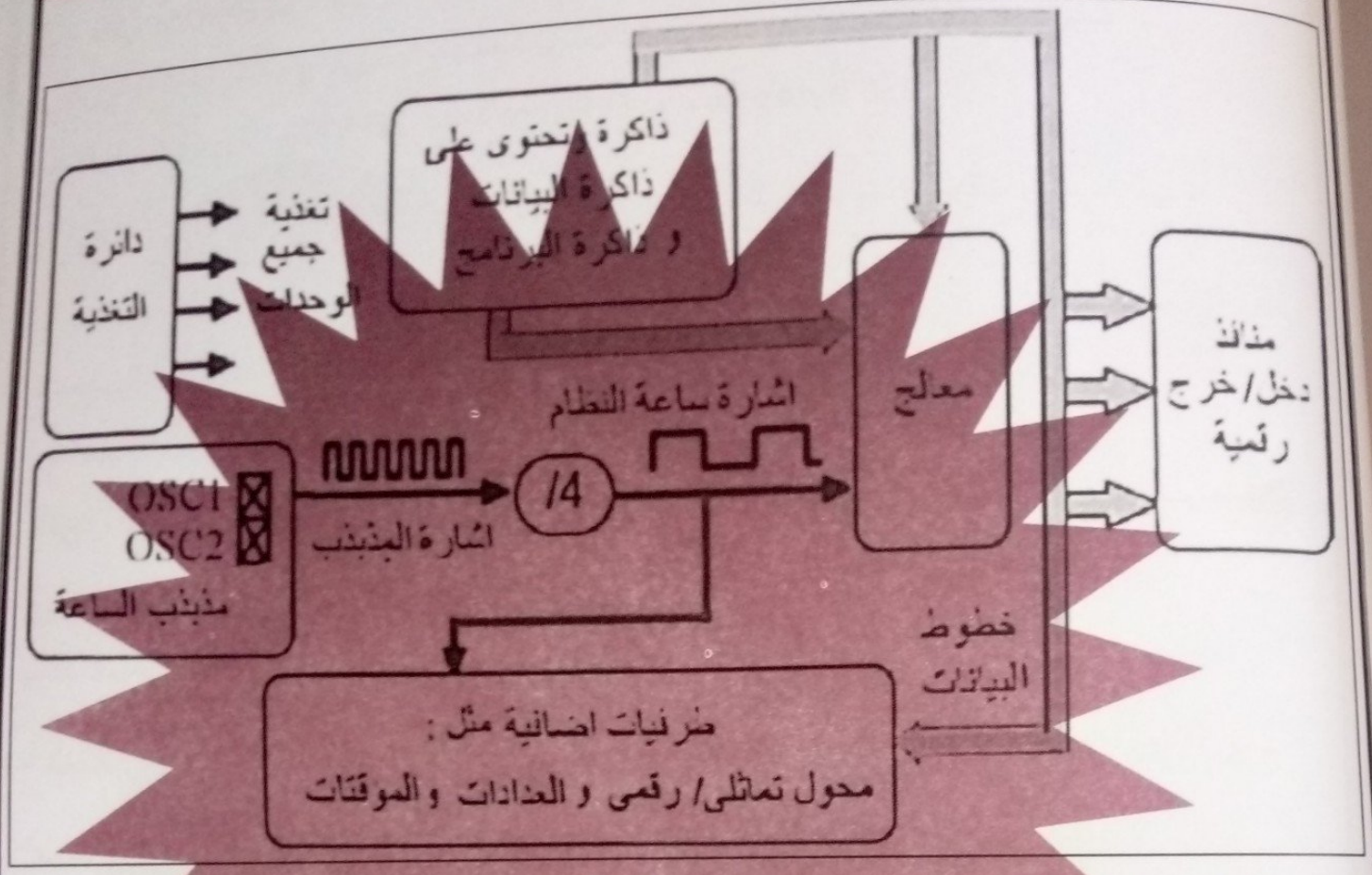
ج _ سعة ذاكرة البرنامج 16 x 16384 من النوع ROM .

د _ سعة ذاكرة البيانات 1536 bytes .

هـ _ مذبذب الساعة الداخلي تردده 40 M Hz .

و _ تحتوي على 8 محولات (من تماثلي إلى رقمي) .

المكونات الأساسية للمتحكم الدقيق :



البناء المعماري للمتحكمات الدقيقة 16F877A :

١- وحدة المعالجة الرئيسة CPU :

وحدة المعالجة المركزية مصنعة بتقنية RISC وتعني مجموعة التعليمات المخفضة و هذة التقنية تعطي للمعالج ميزتين هما :

أ- **المعالج لا يتعرف إلا على 35 تعليمة بسيطة .**

ب- **زمن تنفيذ التعليمة واحد لمعظم التعليمات ويستغرق 4 دورات من نبضات الساعة .**

فاذا كان الميكروكنترولر يعمل عن **تردد 20 MHZ** فان البرنامج سوف يقوم ب 5 مليون تعليمة في الثانية .

٢ – الذاكرة **MEMORY** : يمتلك الميكروكنرولر ثلاثة أنواع من الذاكرة :

أ – **ذاكرة القراءة فقط ROM (ذاكرة البرنامج) :**

تستخدم ذاكرة Rom في الحفظ الدائم للبرنامج المطلوب تنفيذه ، و يمتلك الميكروكنرولر 16F877A **ذاكرة Rom ذو سعة 8kb باجمالي 8192 موقع و هي من النوع الوميضي Flash .**

ب – ذاكرة EEPROM :

و هي تشبه ذاكرة البرنامج في انها يتم حفظ بياناتها بشكل دائم و لكن يمكن تغيير محتوياتها .

و يمتلك الميكروكنرولر 16F877A **ذاكرة EEPRom ذو سعة 256 bytes** .

جـ _ ذاكرة الوصول العشوائي RAM : و تستخدم في تخزين البيانات بشكل مؤقت و تنقسم الى :

- سجلات الأغراض العامة : (GPR)

و تستخدم في الحفظ المؤقت للبيانات و النتائج المتولدة خلال العمل .

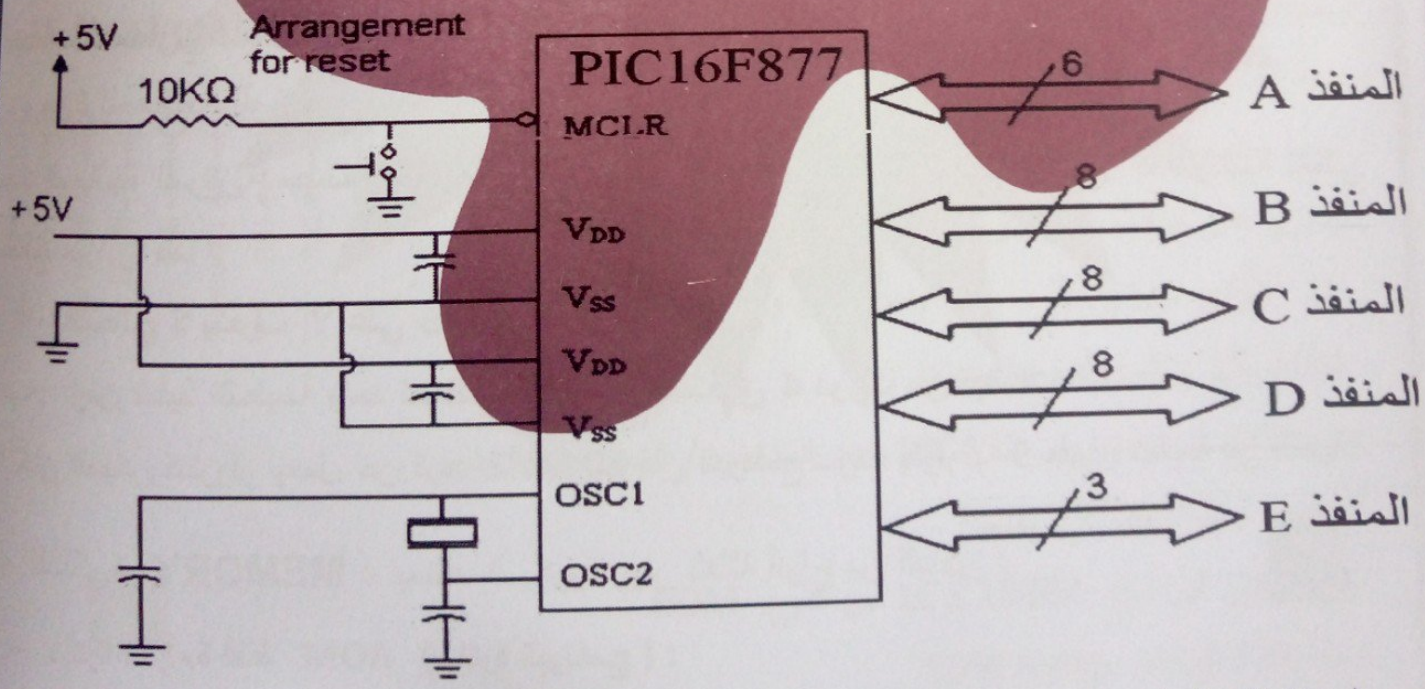
- سجلات الوظائف المخصصة : (SFR)

و الغرض منها محدد مسبقا من خلال عملية التصنيع و لا يمكن تغييره و أى تغيير في محتواها يؤثر مباشرة على عمل المتحكم أو بعض دوائره .

و يمتلك الميكروكنرولر 16F877A **ذاكرة Ram ذو سعة 368 bytes** .

٣– المنافذ (بوابات الدخل / الخرج) : يمتلك الميكروكنرولر 16F877A **33 طرف دخل / خرج**

و هى وسيط بين العالم الخارجى و المتحكم الدقيق ، وله خمس منافذ متوازية وهى :



٤– المنفذ التسلسلي :**Serial port** :

وهذا المنفذ يسمح بتبادل البيانات بين المتحكم و الأجهزة الأخرى مثل الكمبيوتر و المتحكمات الأخرى .

٥- الساعة / المذبذب :

يقوم بضبط الفواصل الزمنية بين مختلف عمليات المتحكم ، لتحدث تزامن بين المعالج و باقي الوحدات و لتنفيذ البرنامج التعليمات واحدة تلو الاخرى .

٦- المؤقت **Timer** :

يسمح المؤقت للمتحكم بالقيام بالمهام لفترات زمنية محددة ، و **يمتلك المتحكم ثلاثة مؤقتات** و هم :

TMR 0 : و هو عبارة عن سجل 8 bits و يستخدم كعداد .

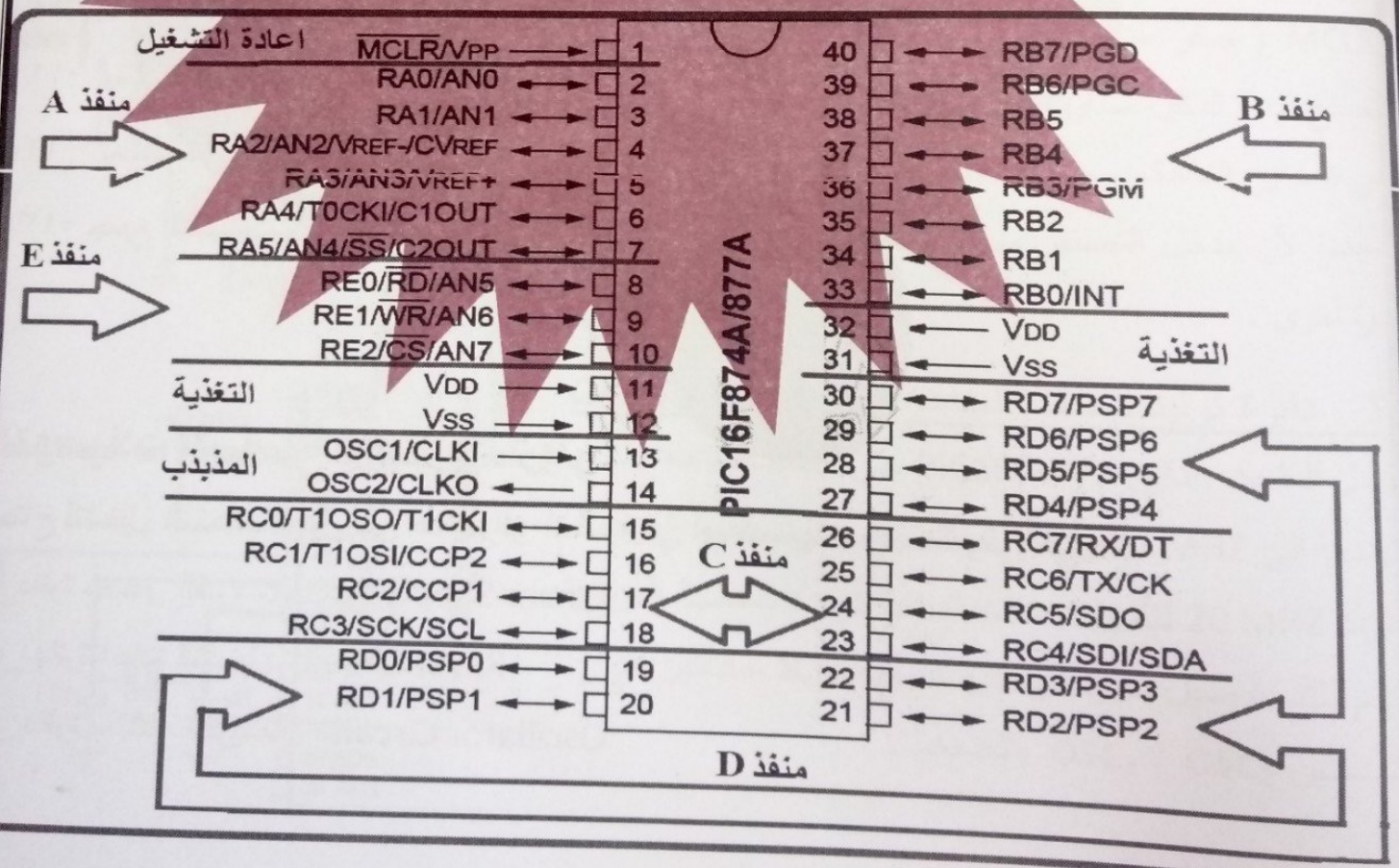
TMR 1 : و هو عداد ذو سعة 16 bits يتكون من سجل بيانات علوي TMR 1H وسجل بيانات سفلي TMR 1L .

TMR 2 : و هو عداد ذو سعة 8 bit يستخدم في توليد خرج **بتعديل عرض النبضة PWM** و يستخدم في تشغيل محركات التيار المستمر و محركات السرفو.

٧- محول تماثلي / رقمي :

و يقوم بترجمة بيانات الدخل من الصورة التماثلية الى الصورة الرقمية حتى يستطيع المتحكم التعامل معها .

أطراف المتحكم الدقيق 16F877A :



خصائص ومميزات الميكروكنترولر PIC 16F877A :

١- تمتلك شريحة الميكروكنترولر PIC 16F877A 40 طرف منها 33 طرف مخصصة كمداخل و مخارج .

٢- يستخدم 35 تعليمة فقط .

٣- سعة ذاكرة EEPROM : هي 256 btyes .

٤- سعة ذاكرة RAM : هي 368 btyes .

٥-سعة ذاكرة ROM : هي 8 KB وهي من النوع FLASH .

٦- جهد مصدر التغذية من 2 V الى 5.5 V .

٧- تردد العمل من 0 Hz الى 20 M Hz .

٨- ذات تركيب بنائي من نوع RISC ذو التعليمات المخفضة .

٩- يمكن برمجة الشريحة حتى ١٠٠٠٠٠ مرة .

١٠- يمتلك محولات من تناظري إلى رقمي .

١١- يمتلك ٣ مؤقتات مستقلة .

١٢- يمتلك خرج قيادة تحكم بتعديل عرض النبضة PWM .

١٣- جميع التعليمات تستغرق دورة تعليمات واحدة فيما عدا تعليمات التفرع .

التوصيلات الأساسية للميكروكنترولر :

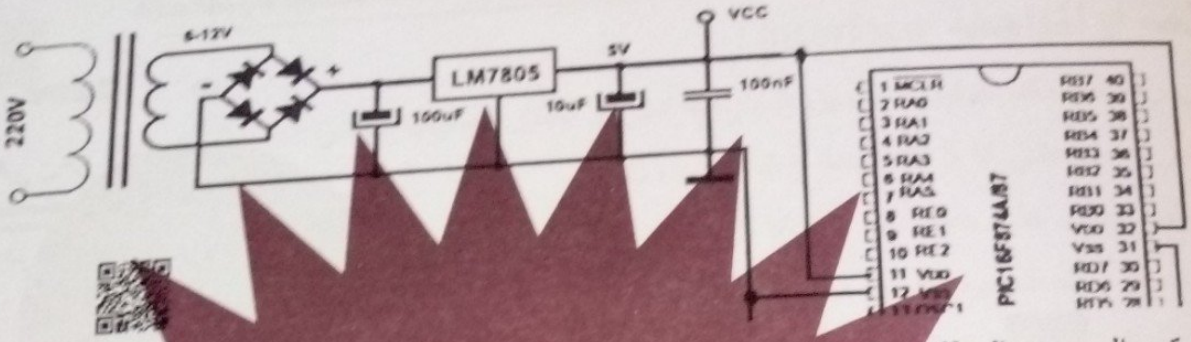
نحتاج لتشغيل المتحكم الدقيق الى عدة دوائر أساسية في أي تصميم ، وهذه الدوائر هي :

أ_ دائرة مصدر القدرة .Power Supply Circuit

ب_ دائرة اعادة التشغيل .Reset Circuit

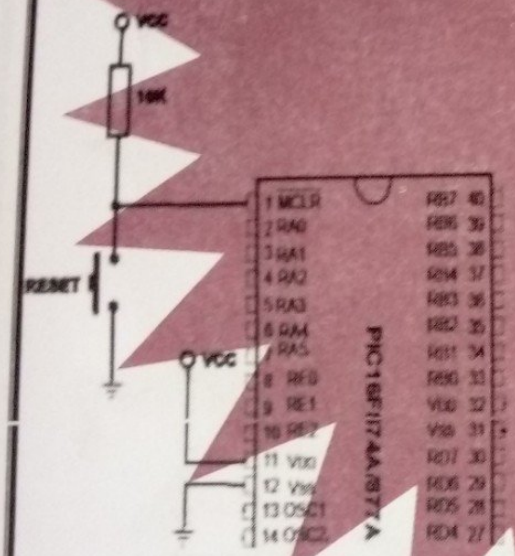
ج_ دائرة مذبذب الكريستال .Oscillator Circuit

١ _ دائرة مصدر القدرة Power Supply Circuit



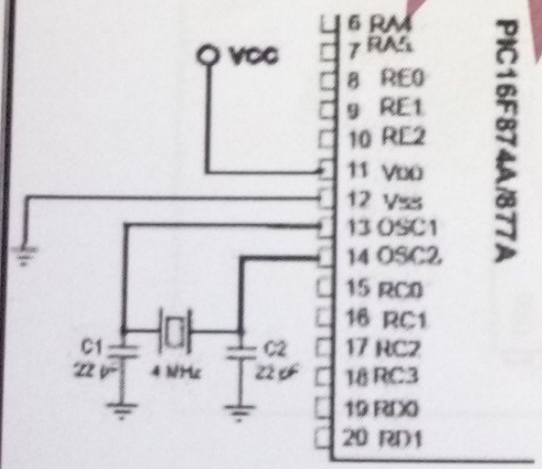
تختلف قيم المجهود المطلوب لتغذية المتحكمات من نوع لأخر ، و المتحكم PIC 16F877A يحتاج الى جهد تغذية من 4.5 V الى 5.5 V ولذلك تستخدم دائرة منظم جهد من نوع LM7805 لتوفير استقرار جهد وتيار كافي لتغذية المتحكم لأن التغذية الغير مستقرة للمتحكم ستؤدي الي عمل غير مستقر للمتحكم .

٢ _ دائرة اعادة التشغيل Reset circuit



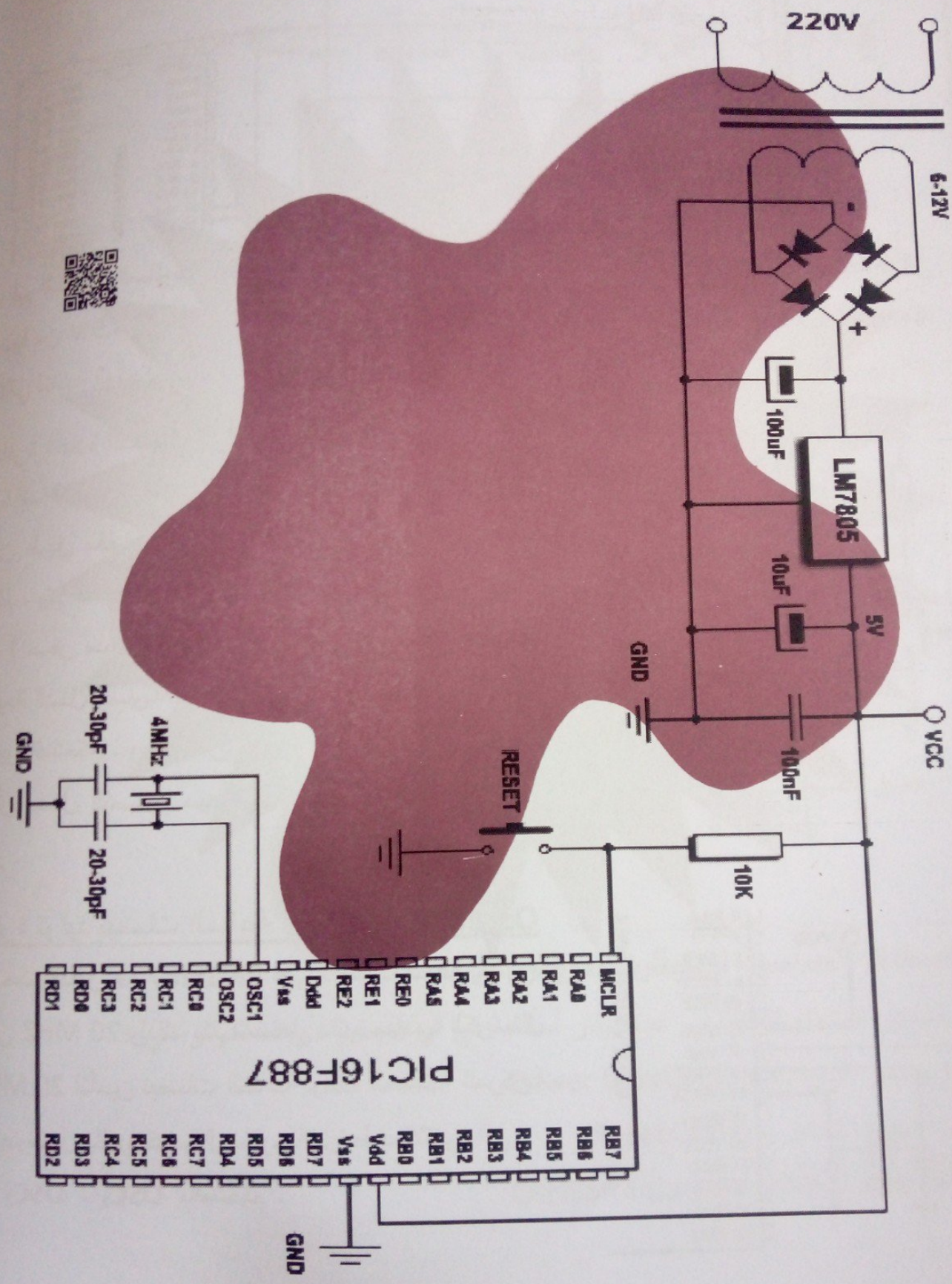
يوصل طرف التصفير Master Clear (MCLR) دائما بقيمة جهد 5 V عن طريق مقاومة قيمتها 10 k لكي يبقى المتحكم في حالة عمل دائم ، يتم عملية التصغير بتطبيق مستوى منخفض على مدخل MCLR (صفر منطقي) بالضغط على المفتاح اللحظي بحيث تصبح قيمة الدخل مساوية لجهد الأرضي أي (0 V) والذي يتسبب في تصغير المتحكم . وعند تحرير المفتاح سيعود المتحكم الى العمل مجددا لأن مدخل التصفير سيعود للمستوى العالي (واحد منطقي) مرة أخرى .

٣ دائرة توليد نبضات الساعة Oscillator Circuit



يعمل المتحكم الدقيق PIC16f877a على نبضات ساعة يصل ترددها الى 20 MHZ ولذلك يتم استخدام دائرة مذبذب الكريستال بتردد 20 MHZ لتأمين نبضات الساعة لوحدة المعالجة المركزية . ويتم ذلك بتوصيل بللورة كريستال خارجية ترتبط بمكثفتين الى المدخلين OSC_2 ، OSC_1 بالمتحكم .

التوصيلات الأساسية الثلاثة للميكروكنترولر



الباب الثانى : مقدمة عن مواجهة المتحكمات الدقيقة

الموائمة أو المواجهة Interface :

و تعني ربط المتحكم الدقيق بالأجهزة الخارجية التي لا يمكن توصيلها مباشرة معه مثل :

أ_ موتور يعمل على 5 V لكنه يحتاج تيار 100 mA .

ب_ موتور يعمل على جهد أكبر من 5 V .

ج_ الأحمال ذات الجهود العالية مثل 220 V تيار متردد .

ولأن خرج المتحكم هو اما 0 V أو 5 V وتيار 25 ma لذا لزم عمل دائرة هاردوير مصممة للربط بين الميكروكنترولر والأحمال المتصلة معه .

دوائر المواجهة القياسية Standard Interfacing Circuits :

* دائرة الترانزستور ثنائي القطبية كمفتاح **Transistor Interfacing**
* دائرة القيادة المتكاملة **Driver IC Interfacing**
* دائرة المرحل **Relay Interfacing Circuit**

أ المواجهة مع أجهزة الخرج **output devices interfacing**

١_ الموحد الباعث للضوءLED

٢_ لمبة الآشارة lamp

٣_ الزنان Buzzer

٤_ الساعة speaker

٥_ محركات لعب الأطفال motors

٦_ محركات الخطوة stepper motors

٧_ وحدة العارضات السباعية 7-sigment

ب المواجهة مع أجهزة الدخل **Input Devices Interfacing** :

١_ المفاتيح switch

٢_ المقاومات المتغيرة . potentioneter

٣_ المقاومات الضوئية LDR.

٤_ المقاومات الحرارية thermistor.

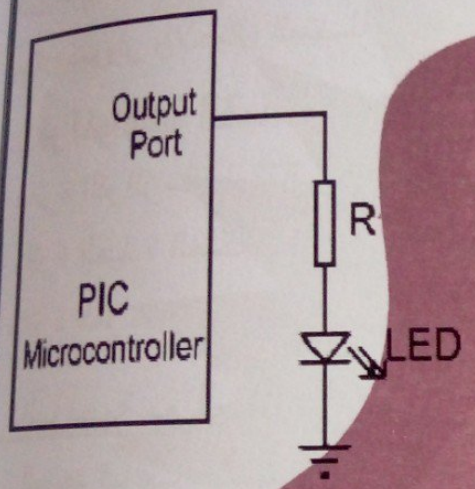
ج المواجهة مع الأجهزة المتقدمة **Advanced Components** :

١_ وحدة شاشة الكريستال السائل .LCD

٢_ وحدة الاتصال التسلسلى مع الكمبيوتر . serial comminication

أولا : دوائر المواجهة القياسية Standard Interfacing Circuits :

من أبسط عناصر المواجهة مع المتحكم هي المقاومة . فهي تستخدم للربط بين المتحكم و أجهزة الخرج مثل الموحد الباعث للضوء و وحدة العارضات السباعية .



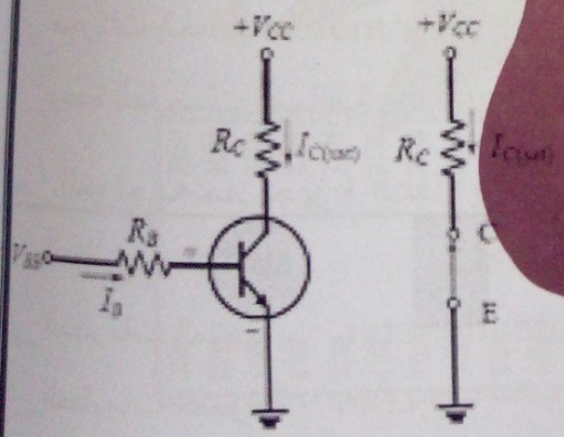
ربط الليد مع المتحكم :

جهد خرج المتحكم هو 5 V وتيار خرج 25 mA بينما يحتاج الموحد الباعث للضوء LED الى جهد تشغيل يتراوح من 2v : 1.5 وتيار في حدود 10 mA حسب نوع مادته ونظرا لهذا التفاوت بين قيم جهد وتيار خرج المتحكم وجهد وتيار تشغيل الليد فانه يتم استخدام " **المقاومة** " كعنصر ربط بين المتحكم والليد لتقليل قيمة الجهد من الـ 5 V الى 2 V اللازمة لتشغيل الليد .

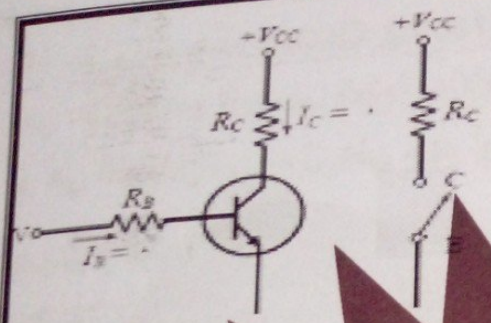
١- دائرة الترانزستور ثنائي القطبية **BJT Transistor** :

تستخدم دئرة ترانزستور ثنائي القطبية كمفتاح للربط بين المتحكم **والاحمال الثابتة التي تعمل على جهد أو تيار مستمر** . مثل توصيل محرك تيار مستمر يعمل علي DC 12 V وتيار قيمته 100 mA مع المتحكم .

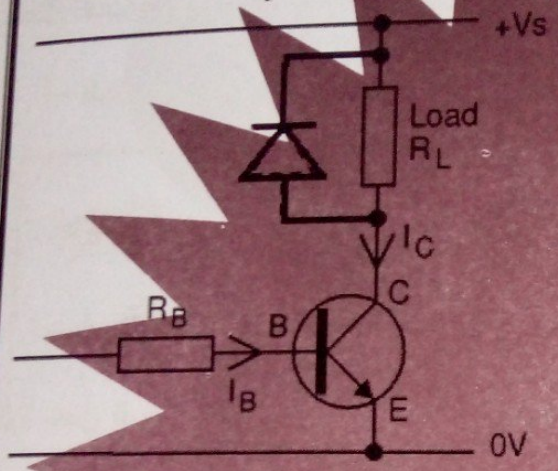
فكرة عمل الدائرة :



- إذا طبقنا قيمة جهد اكبر من قيمة جهد انحياز قاعدة الترانزستور (0.7v) فإن الترانزستور يكون في حالة التشبع وبالتالي يمر تيار من المجمع إلي المشع اي يتم توصيل بين النقطتين E,C ويكون جهد وصلة المجمع–المشع VCE تساوي صفر .



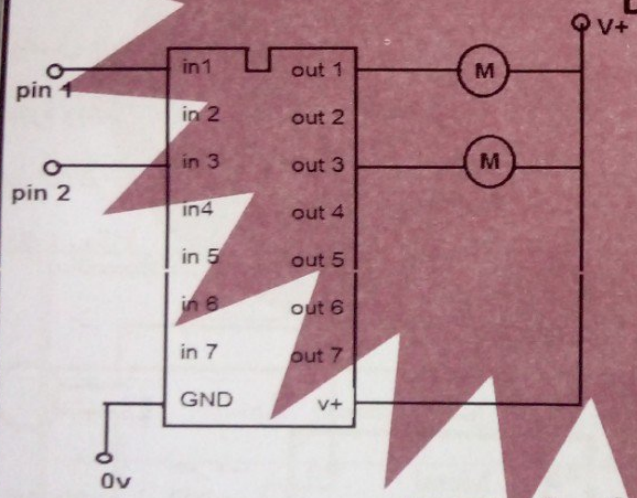
• إذا طبقنا قيمة جهد اقل من قيمة جهد انحياز قاعدة الترانزستور (0.7v) فإن الترانزستور يكون في حالة القطع وبالتالي لا يمر تيار من المجمع إلي المشع اي يصبح بين المجمع والمشع open circiut ويكون قيمة جهد وصلة المجمع المشع VCE تساوي 5 V .



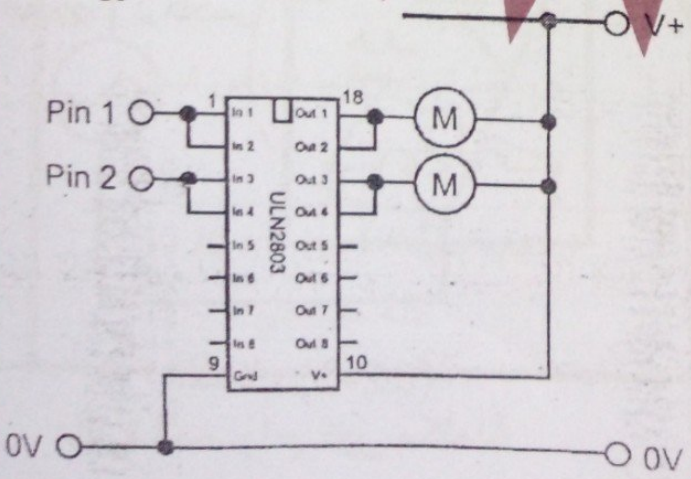
دائرة استخدام الترانزستور كمفتاح (**switch**) :

يلزم توصيل موحد علي التوازي مع اي حمل حثي أو اي حمل يحتوى علي ملفات مثل المحركات او الريلاي أو غيرهم ، **وذلك لحماية الترانزستور و المتحكم من التيار العكسي** Reverse current الناتج عن الطاقة المخزنة ف ملفات الحمل عند توقفه عن العمل.

٢- دائرة القيادة المتكاملة **Driver IC Interfacing**



يتم استخدام الشريحة **ULN 2003 Driver Ic** كدائرة مواجهة مع المتحكم إذا أردنا التحكم في عمل أكثر من جهاز الدخل او الخرج ، والتي تتكون داخلياً من سبعة ترانزستورات BC x 38B وعدد أطرافها 16 طرف .



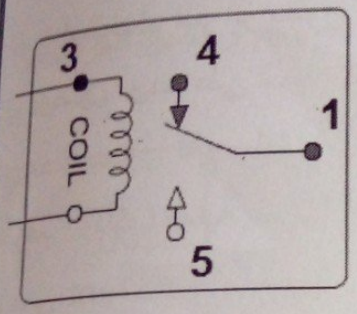
و هناك شريحة اخرى **ULN 2803 Driver Ic** والتي تتكون داخلياً من ثمانية ترانزستورات BC x 38B وعدد أطرافها 18 طرف .

٣- دائرة المرحـل **Relay Interfacing Circuit** :

وهي دائرة تستخدم للربط بين المتحكم والأحمال المترددة **AC Loads**

و هي الأحمال التي تعمل علي جهد أو تيار متردد .

والمرحل (الريلاي) هو:عبارة عن عنصر كهربي مكون من مفتاح ميكانيكي يمكن التحكم به كهربيا من خلال تطبيق جهد علي الملف الموجود بداخله . و يتكون من خمسة رجول مقسمين إلي جزئين :



أ- **الجزء الأول** : خاص بتوصيل جهد 220 v للحمل

وهذه الأرجل هي 5 , 4 , 1

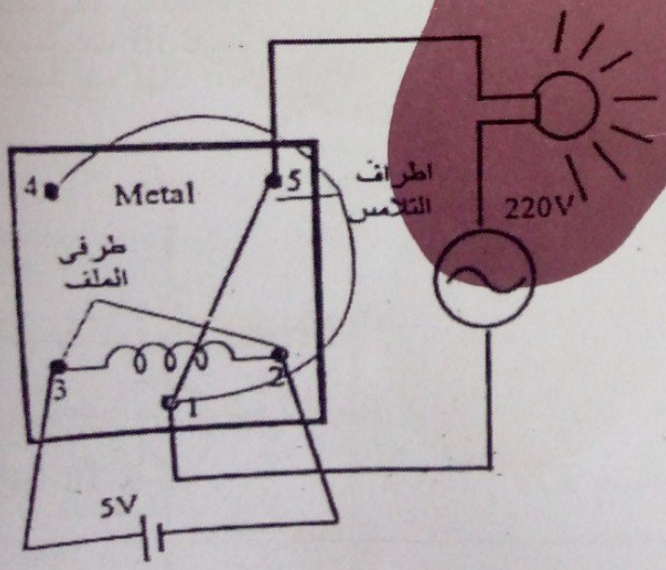
ب- **الجزء الثاني** : خاص بتوصيل جهد اما 5 , 6 , 2 فولت

علي الطرفين 3 , 2

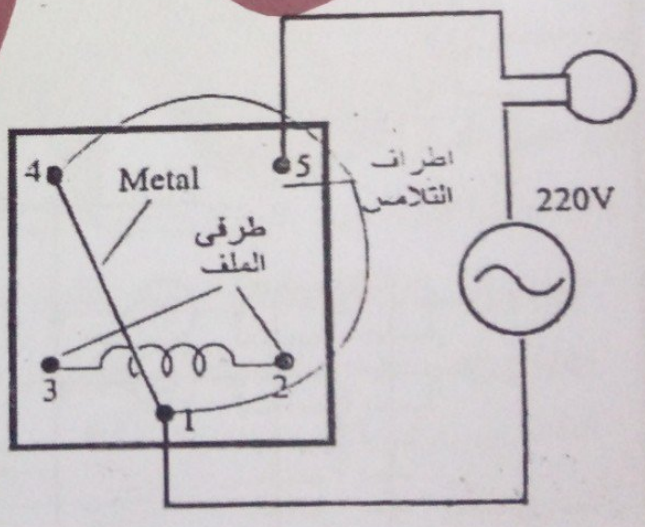
فكرة عمل الريلاي :

عند تطبيق فرق جهد بين الطرفين 3 , 2 فإن الملف يولد مجالاً مغناطيسياً يؤثر علي طرف التلامس الواصل بين الطرف 4 , 1 ويجعله يتحرك من النقطة 4 إلي النقطة 5 ليكمل الدائرة وتوصيل الحمل بمصدر القدرة وبالتالي يضئ المصباح .

أما إذا تم تطبيق فرق جهد صفر بين طرفي الملف فإن دائرة المصباح تكون غير مكتملة ولن يمر تيار في الدائرة وبالتالي لن يضئ المصباح .



الشكل (2-13)
دائرة المصباح مكتملة

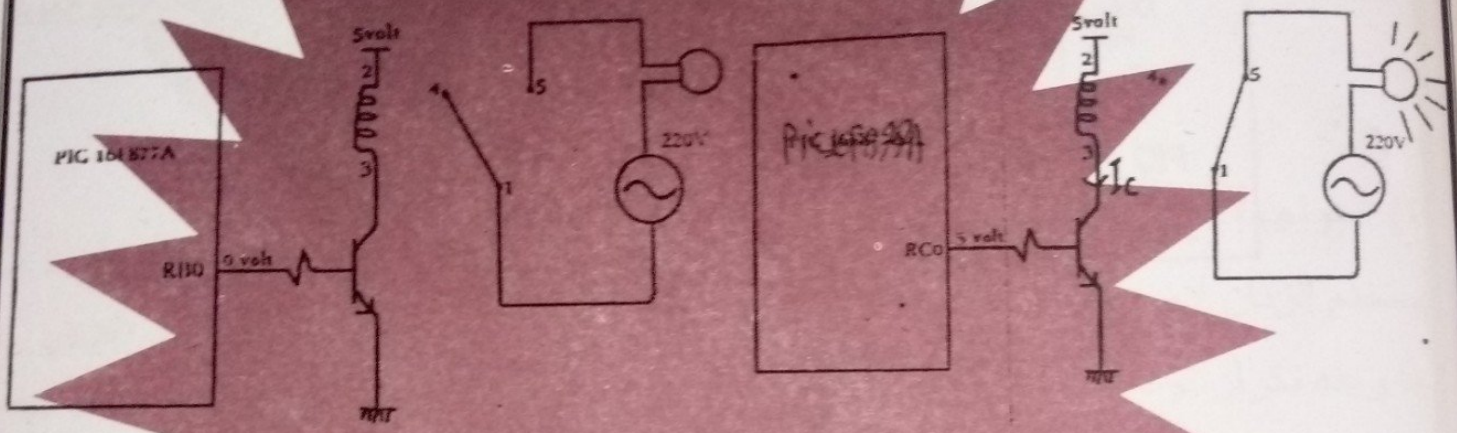


الشكل (2-12)
دائرة المصباح غير مكتملة

استخدام الترانزستور والريلاي كدائرة مواجهة للتيارات العالية :

إذا كان قيمة جهد خرج المتحكم هي صفر (OV) اي صفر منطقي فإن الترانزستور يكون في حالة القطع و لن يمر تيار و بالتالي لن يمر تيار في ملف الريلاي و لن يتم توصيل الحمل .

أما إذا كانت قيمة جهد خرج المتحكم هي (5 V) اي واحد منطقي فإن الترانزستور في حالة التشبع و بالتالي يمرر تيار المجمع ويسمح بمرور تيار في ملف الريلاي مما يجعل طرف التلامس يوصل دائرة المصباح فتكتمل الدائرة و يضئ المصباح .



المصباح لا يضئ ∴ lamp off open circuit T1 off (0 v) Logic 0

المصباح يضئ ∴ relay on and lamp on short circuit T1 on (5 V) Logic 1

مميزات الريلاي الكهروميكانيكي	عيوب الريلاي الكهروميكانيكي
عزل تام بين دائرة التحكم ودائرة الجهد العالي	يلزم إضافة عناصر للحماية
يمكن أن يوجد أكثر من زوج من التلامسات التي تمكننا من التحكم في أكثر من دائرة في نفس الوقت	تتآكل نقاط التلامس بعد فترة من الزمن
يمكن ان تتحمل نقاط التلامس تيارات عالية جداً	استجابة بطيئة ما بين 5 : 10 ms لأنه ميكانيكي

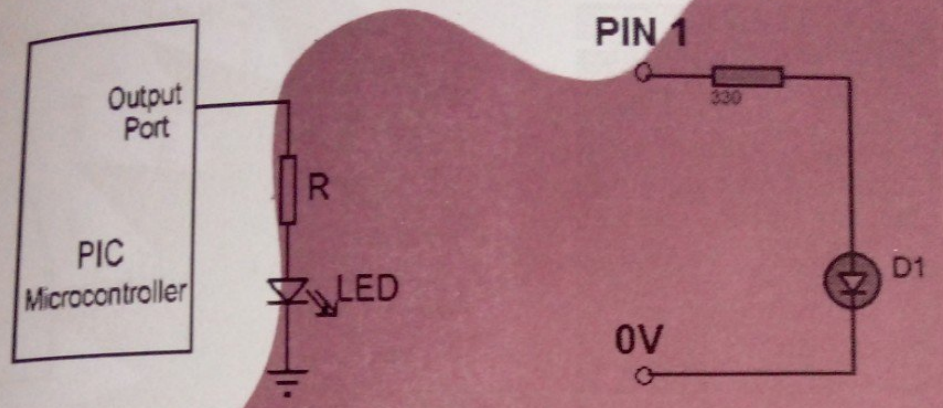
ب- المواجهة مع اجهزة الخرج Out put Devices Interfacing :

١- المواجهة مع الموحد الباعث للضوء **LED** :

يتم الربط بين الليد والمتحكم عن طريق مقاومة توصيل علي التوالي مع الليد ويمكن حساب قيمتها من خلال المعادلة $Rs = (V_s - V_f) / I_f$

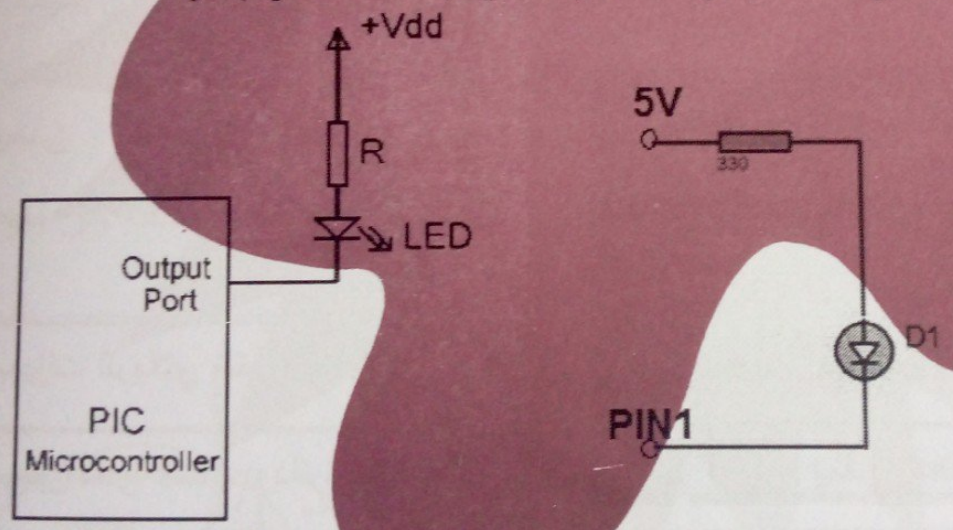
طريق توصيل الموحد الباعث للضوء LED مع المتحكم:

طريقة دفع التيار:



و فيها يضئ الليد عندما يكون خرج المتحكم في المستوي العالي (واحد منطقي) أي 5V .

طريقة سحب التيار:



و فيها يضيئ الليد عندما يكون جهد خرج المتحكم في المستوي المنخفض (صفر منطقي) اي 0 V .

استخدامات الثنائي الباعث للضوء :

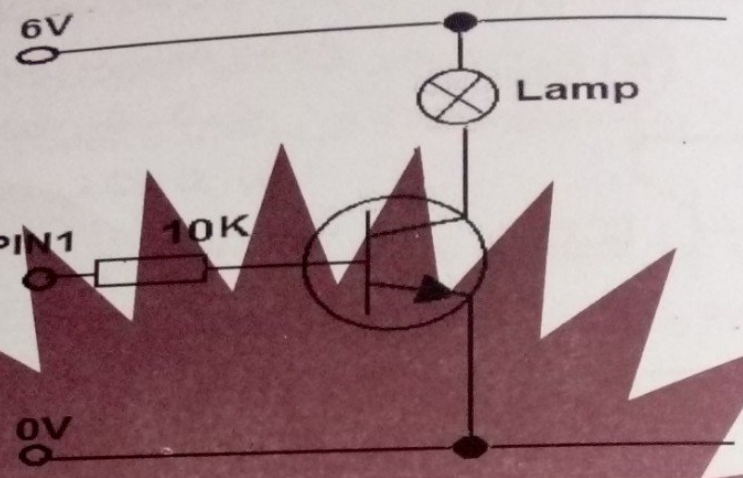
١) في العدادات الرقمية .

٢) في الحاسب الآلي .

٣) في انظمة الاتصالات الضوئية .

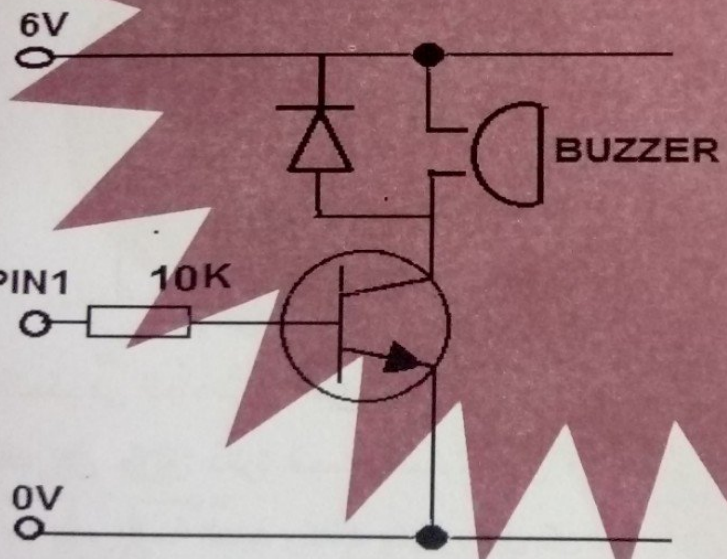
٤) في حاسبات الجيب لإظهار الأرقام و الحروف والاشارات في الشريحة السباعية 7 sigment .

٢) المواجهة مع لمبة الاشارة:

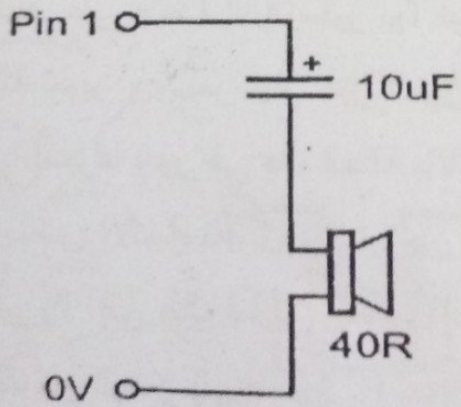


٣) المواجهة مع آلة التنبيه **Buzzer** :

يستخدم الزنان لاصدار اصوات تنبيه ويتكون من رقاقة معدنية تتحرك عند توصيل تيار كهربي لها وعند تكرار توصيل وقطع التيار الكهربي تصدر صوت تنبيه مع اهتزاز في نفس الوقت و لربط آلة التنبية مع المتحكم تستخدم دائرة الترانزستور كمفتاح ويلزم توصيل موحد لحماية دائرة الترانزستور من التيار العكسي.



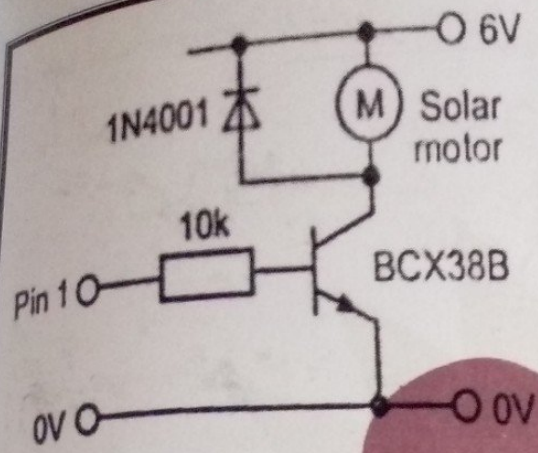
٤- المواجهة مع السماعة **Speaker** :



٥- المواجهة مع محركات لعب الأطفال :

يمكن ربط محركات لعب الأطفال مع المتحكم باستخدام دائرة الترانزوستور كمفتاح لتشغيل وإيقاف المحرك.

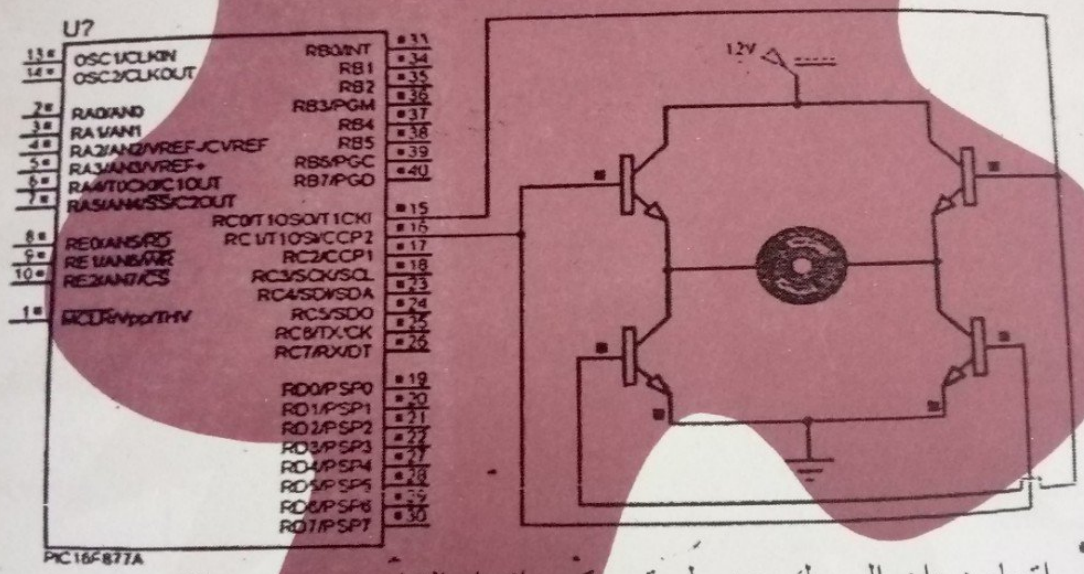
وتتميز هذه المحركات بأنها رخيصة الثمن لكن يعيبها أنها تنتج كثيراً من الضوضاء الكهربية والتي تؤثر علي المتحكم ولذلك يمكن توصيل مكثف قيمته 220 nF بين طرفي المحرك .



٦- المواجهة مع محرك التيار المستمر DC Motor Intefacing :

يستخدم في العديد من التطبيقات مثل الروبوتات .

التحكم في اتجاه دوران المحرك :



يمكن التحكم في اتجاه دوران المحرك عن طريق عكس اتجاه التيار فيه ، ويتم ذلك بعكس قطبي البطارية ويتم ذلك عن طريق دائرة الجسر المكونة من أربع ترانزستورات على شكل حرف H حيث يتم توصيل T_1 و T_2 بنفس قيمة الجهد وكذلك يتم توصيل كل من T_3 , T_4 بنفس القيمة .

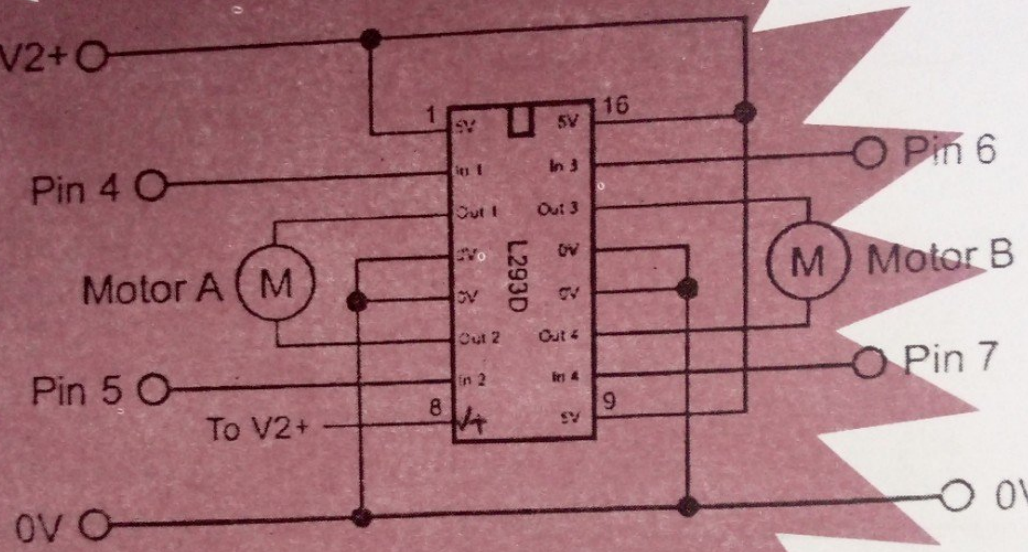
- عند تطبيق الجهد على قاعدة الترانزستور T_3 فينتقل إلى حالة تمرير ويسمح ذلك للترانزستور T_4 بأن ينتقل أيضا لحالة تمرير ، و يمر تيار عبر المحرك من اليمين إلى اليسار .
  (و لنفترض أن المحرك يدور في هذه الحالة بالاتجاه الأمامي) .
- لتدوير المحرك بعكس الاتجاه نفصل الاشارة عن دخل الترانزستور T_3 وتطبق على قاعدة الترانزستور T_2 فينتقل الترانزستور T_2 إلى حالة تمرير ويسمح ذلك للترانزستور T_1 بأن ينتقل أيضا لحالة تمرير ، و يمر تيار عبر المحرك من اليسار إلى اليمين . (يدور المحرك بالاتجاه العكسي) .

التحكم في سرعة دوران المحرك :

يتم التحكم في سرعة دوران المحرك عن طريق تغيير قيمة الجهد المطبق عليه ، فمثلاً نفترض أن جهد تشغيل الحرك هو 12 v واذا وصلناه بقيمة جهد 6 v فان سرعته سنقل النصف و كذلك اذا وصلناه بقيمة جهد 3v سنقل أيضاً وهكذا.

وبذلك يتم التحكم في سرعة دوران المحرك عن طريق عمل تعديل أو تغيير في قيمة جهد التغذية من خلال تعديل عرض النبضة PWM وذلك للحصول علي قيم مختلفة للجهود .

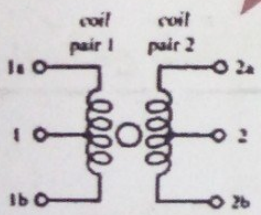
للتحكم في محركين تستخدم شريحة درايفر L293D :



٧- محرك الخطوة **Stepper Motor** :

هو محرك يتم المتحكم فيه رقمياً و يتحرك خطوه step في كل مرة ، و كل خطوة تتحدد بزاوية الخطوة و تتميز هذه المحركات بالتحكم بالموضع بدقة عالية و سرعات منخفضة لذلك تستخدم في :

فتح الأبواب أوتوماتيكيا ، و في الطابعات و الراسمات و التلسكوبات .

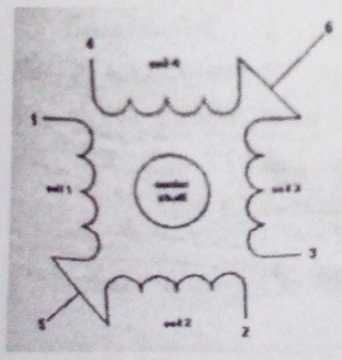


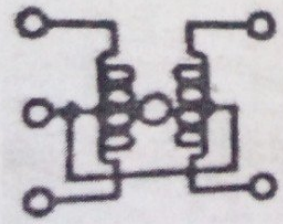
أنواع محركات الخطوة :

أ- محرك الخطوة أحادي القطبية :

وتحتوي علي أربعة ملفات و يوجد منه النوع ذو الخمس أسلاك و نوع ذو ستة أسلاك .

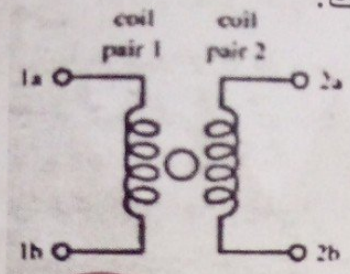
وتسمي الحركات أحادية القطبية لأن القدرة دائما تمر في اتجاه واحد و تأتي من توصيله المنتصف .





ب- محرك الخطوة ثنائي القطبية :

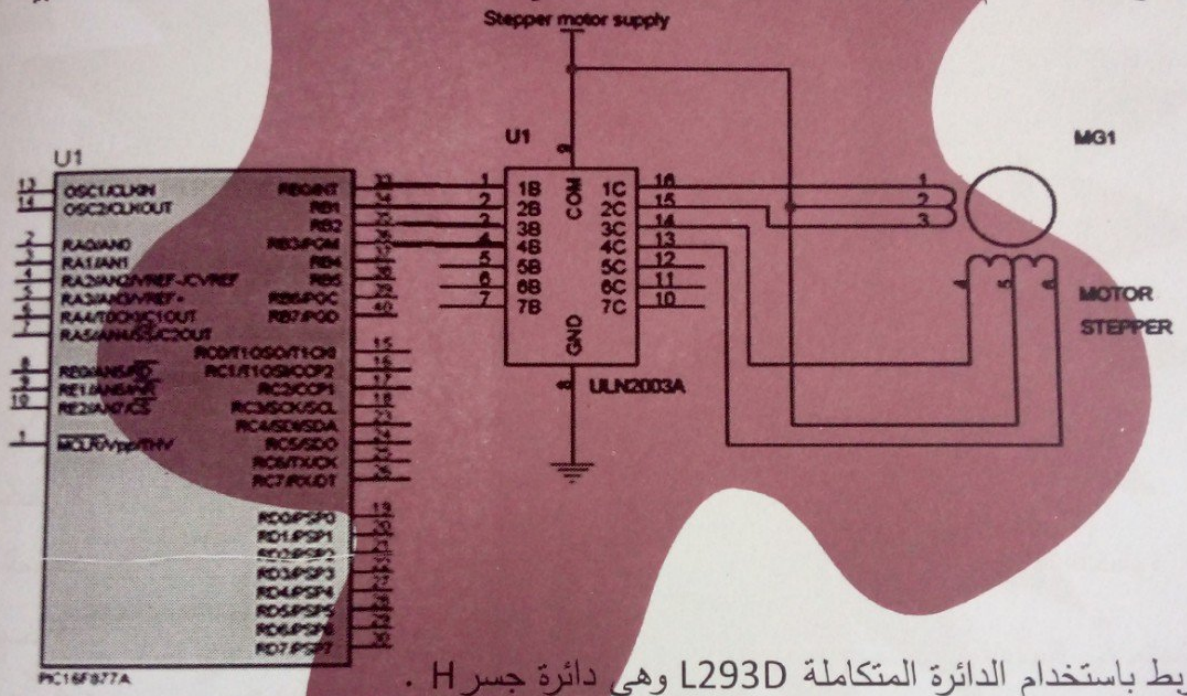
وهي محركات تحتوي علي ملفين وذو أربعة أسلاك.



المواجهة مع المحركات الخطوية الأحادية :

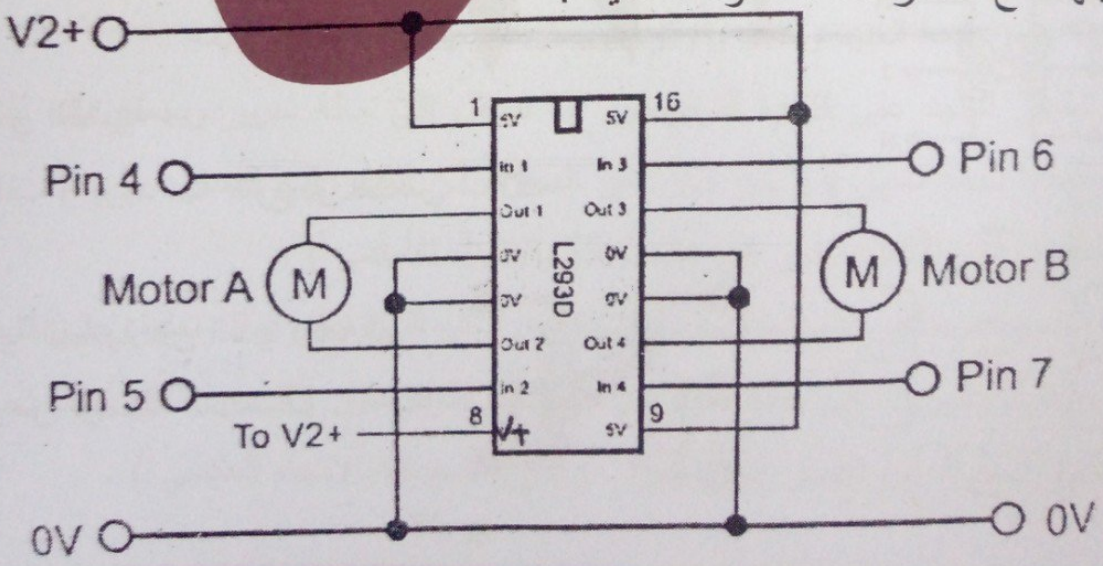
يتم الربط بين محرك الخطوة أحادي القطبية والمتحكم بطريقتين:

أ- الربط باستخدام الدائرة المتكاملة ULN 2003 وهي مصفوفة من ترانزستورات ثنائية القطبية .



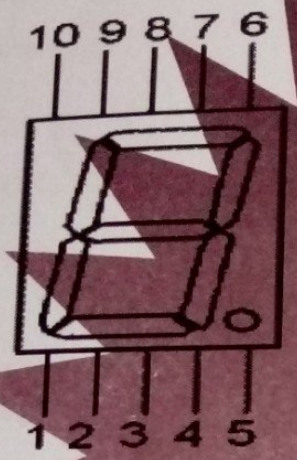
ب- الربط باستخدام الدائرة المتكاملة L293D وهي دائرة جسر H .

المواجهة مع المحركات الخطوة الثنائية :



المواجهة مع وحدة العارضات السباعية **7-Sigment Display Interface** :

(7- Sigment) : هي عبارة عن سبع ليدات مرتبة بطريقة لتظهر الأرقام وبعض الحروف ، ويوضع ليد إضافي يمثل العلامة العشرية (dot) يستخدم عند إظهار الأرقام التي تحتوي علي علامة عشرية ، وكل ليد له حرف يمثله و يجب معرفته عند توصيل الشريحة بالمتحكم الدقيق و هذه الحروف هي dp, g , f , E , D , C , D , C , B , A حيث انها لها عشرة (10) رجول .



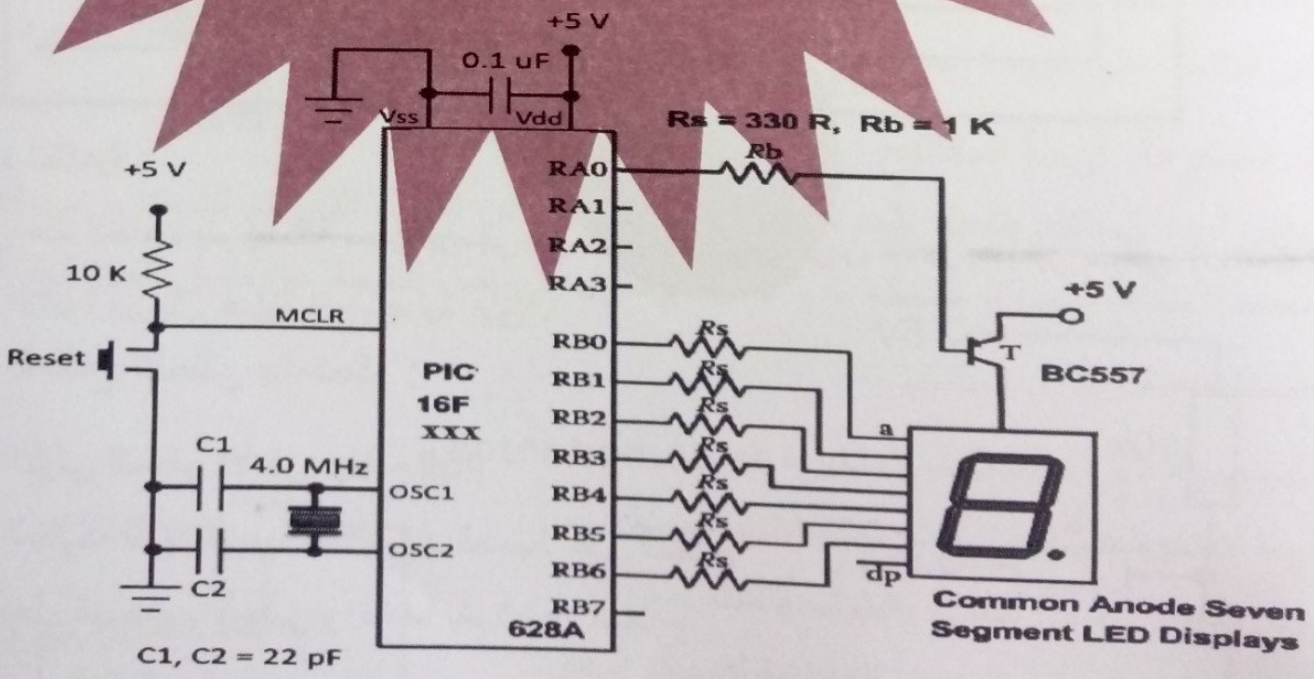
استخدامات شريحة العارضات السباعية :

- تستخدم لعرض درجة الحرارة التي يقوم المتحكم بقياسها .
- تستخدم في المصاعد للدلالة علي رقم الدور .
- تستخدم في البنوك لعرض رقم العميل .
- تستخدم في ساعات الحائط

.أنواع شريحة العارضات السباعية :

١) النوع الأول يسمي **Common Cathode** و يعني الكاثود المشترك و فيها يتم توصيل جميع مهابط الليدات السبعة بالأرضي ، و لإضاءة اي ليد يقوم بإخراج قيمة جهد **5V** علي الرجل المناظرة لها .

٢) النوع الثاني يسمي **Common Anode** و يعني الأنود المشترك ، و فيها يتم توصيل جميع مصاعد الليدات السبعة بالطرف الموجب **5V** ، و لإضاءة اي ليد تقوم بإخراج قيمة جهد **0V** علي الرجل المناظرة لها.



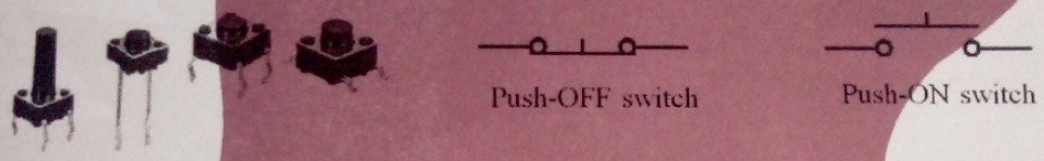
ثانيا: المواجهة مع اجهزة الدخل Input Devices Interface :

١- المواجهة مع المفاتيح Switches :

المفتاح اللحظي (الضاغط) : هو عبارة عن مفتاح ميكانيكي رخيص الثمن عن الضغط عليه يتم التوصيل بين طرفيه أو الفصل بينهما حسب نوع المفتاح و عند تحرير الضغط يعود لوضعه الأصلي .

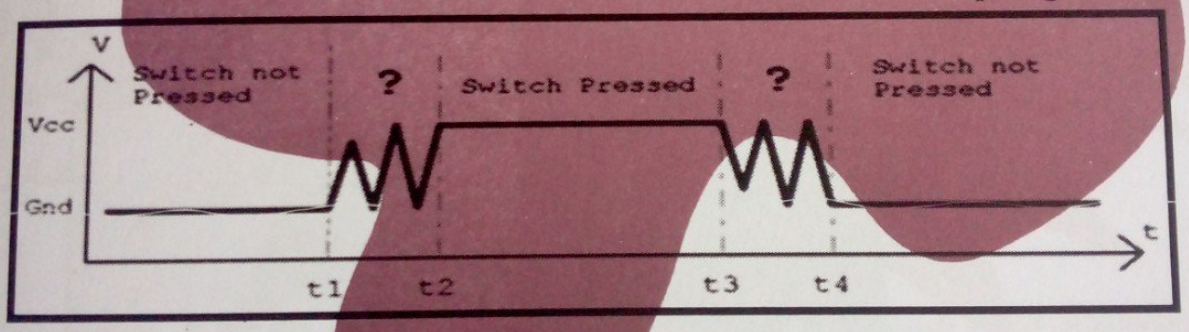
و يوجد منه نوعان : النوع المفتوح طبيعيا Push On و يقوم بالتوصيل بين طرفيه عند الضغط عليه و النوع المغلق طبيعيا Push off و يقوم بالفصل بين طرفيه عند الضغط عليه .

- و لزر الضاغط 4 رجول **لمنحه الاستقرار الميكانيكي** عند التثبيت على اللوحة .



ظاهرة الارتدادات (الوثبات) Bouncing :

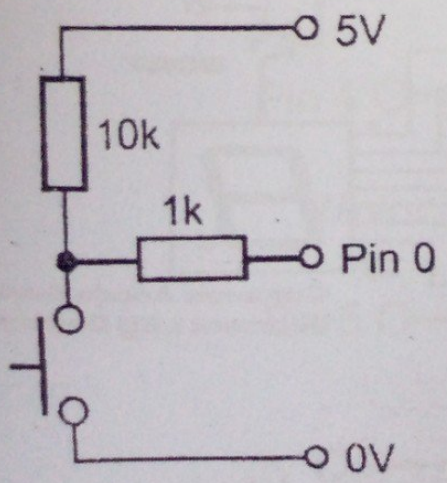
هي ظاهرة غريبة تحدث خلال الملللي ثانية ، حيث يقوم المفتاح بالإرتداد أكثر من مرة نتيجة وجود صفائح معدنية داخل المفتاح بسرعة جداً عند فتح او غلق المفتاح مما يؤدي إلي توصيل و فصل التيار عدة مرات قبل ان يثبت المفتاح علي الحالة الأخيرة .



معالجة هذه الظاهرة :

للتغلب علي هذه الظاهرة يتم إضافة سطر برمجي في الكود عبارة عن زمن تأخير صغير حوالي (25ms) لكي تستقر الاشارة بين كل عملية فتح وعملية غلق .

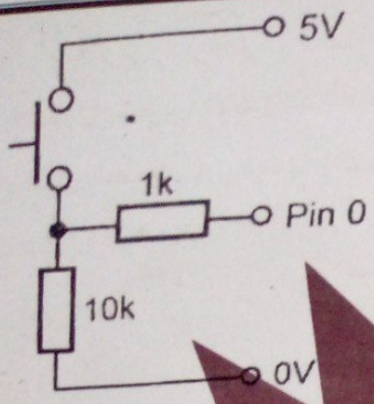
طرق ربط المفتاح اللحظي بالمتحكم :



١) الطريقة الأولي السحب الأعلي Pull – Up

وفيه نتصل المقاومة بجهد التغذية VCC عند الضغط علي المفتاح يتم توصيل رجل المتحكم بالأرضي و عندما يتم تحرير المفتاح يتم توصيل رجل المتحكم بالجهد .

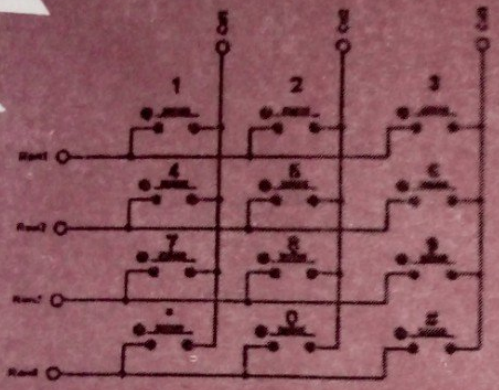
٢) الطريقة الثانية السحب لأسفل Pull - Down

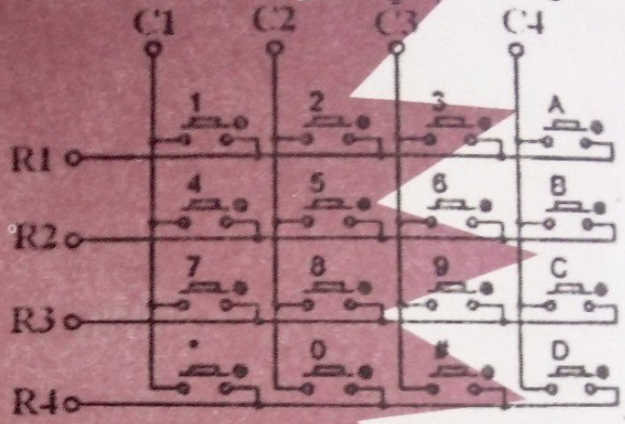


وفيه تتصل المقاومة بالأرضي عند الضغط علي المفتاح يتم توصيل رجل المتحكم بجهد التغذية VCC وعندما يتم تحرير المفتاح يتم توصيل رجل المتحكم بالأرضي .

٢- المواجهة مع لوحة المفاتيح Keypad :

لوحة المفاتيح عبارة عن مصفوفة من المفاتيح اللحظية مرتبة علي هيئة صفوف واعمدة 4 * 4 أو 3 * 4 و تستخدم في تطبيقات مثل الآلة الحاسبة ، الهاتف ، أنظمة الأمان .





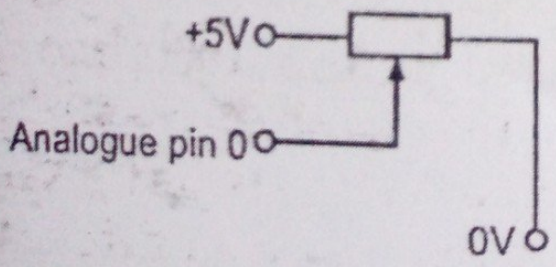
ربط مصفوفة لوحة المفاتيح مع المتحكم :

لربط مصفوفة لوحة المفاتيح من النوع (4*4) مع المتحكم سوف نحتاج إلي 16 طرف من أطراف المتحكم و لكن باستخدام تقنية بسيطة تسمي عملية المسح Scanning يمكننا استخدام ثمانية (8) أطراف فقط ثم عرض النتيجة علي شريحة العارضات السباعية .

عمل مصفوفة لوحة المفاتيح :

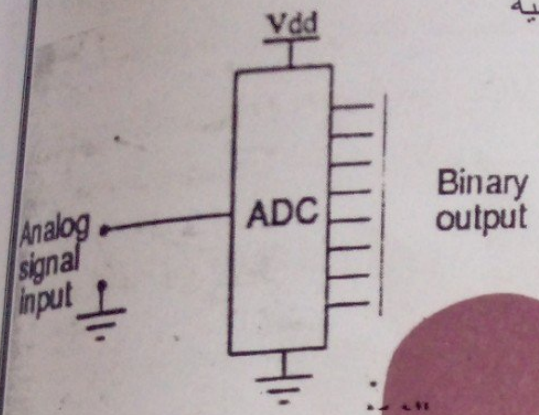
يتم عمل لوحة المفاتيح بتفعيل واحد من الأربعة اعمدة وفحص الصف الذي يتم تنشيطه و من خلال ذلك يتم تحديد المفتاح الذي تم ضغطه ، و يمكن الوصول إلي نفس النتيجة عن طريق تفعيل احد الصفوف وفحص الأعمدة .

٣-المواجهة مع المقاومات المتغيرة Potentiometers :



المقاومة المتغيرة هي المقاومة ذو ثلاثة اطراف يتصل الطرفان الثابتان بطرفي التغذية والطرف الثالث وهو المتغير يتصل بأحد المداخل التناظرية للمتحكم الدقيق.

محول تناظري / رقمي (Analogue / Digital Converter (ADC :



يقوم المحول بتحويل الاشارات التناظرية (التماثلية) إلي إشارات رقمية لكي يتمكن المتحكم من فهمها والتعامل معها بحيث يكون له طرف دخل واحد لاستقبال الاشارة التناظرية ويكون له عدد أطراف خرج رقمي 8 أو 10 او 12 أو 14 حيث ان القيم التي يخرجها المحول تتوقف علي عدد الأرجل الموجودة علي خرجه .

يحتوي متحكم PIC علي محول ذو 10 عشرة أطراف في الخرج و بالتالي أقصي قيمة متاحة بالنظام الثنائي هي (1023) .

القيمة المرجعية للمحول Reference value :

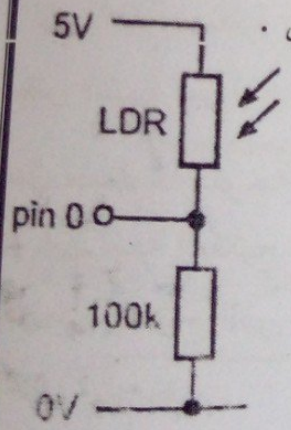
هي القيمة المرجعية تمثل أقصي قيمة يمكن للمحول قراءتها وهي إما تكون تناظرية وتمثل قيم دخل تناظري من 0 V إلي 5 V علي الدخل ، وتكون رقمية وتتراوح من 0 إلي 1023 علي خرج المحول .

أ- القيمة 5 V علي الدخل التناظري يخرجها 1023 وهي بالثنائي (1111111111)

ب- القيمة 2.5 V هي نصف القيمة العظمي علي الدخل التناظري يخرجها 511 أي نصف القيمة العظمي للخرج وهي بالثنائي (0111111111)

جـ- القيمة 1.25 V هي ربع القيمة العظمى علي الدخل التناظري يخرجها 255 أي ربع القيمة العظمي للخرج وهي بالثنائي (00111111111)

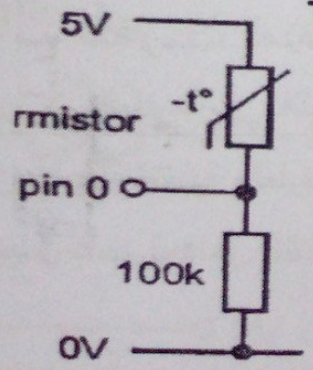
د- القيمة 0 V علي الدخل التناظري يخرجها (0000000000) علي الخرج الرقمي .



٤- المواجهة مع المقاومات الضوئية LDR Interface

المقاومة الضوئية هي مقاومة تتغير قيمتها طبقا لشدة الضوء الساقط عليها حيث كلما زادت شدة الضوء الساقط عليها تقل مقاومتها .

و من اشهر أنواعها ORP-12 التي لها مقاومة عالية جداً



٥- المواجهة مع المقاومات الحرارية Thermistor Interface

المقاومة الحرارية هي مقاومة تتغير قيمتها مع تغير درجة الحرارة T و يتم ربطها مباشرة مع المتحكم علي أحد المداخل التناظرية .

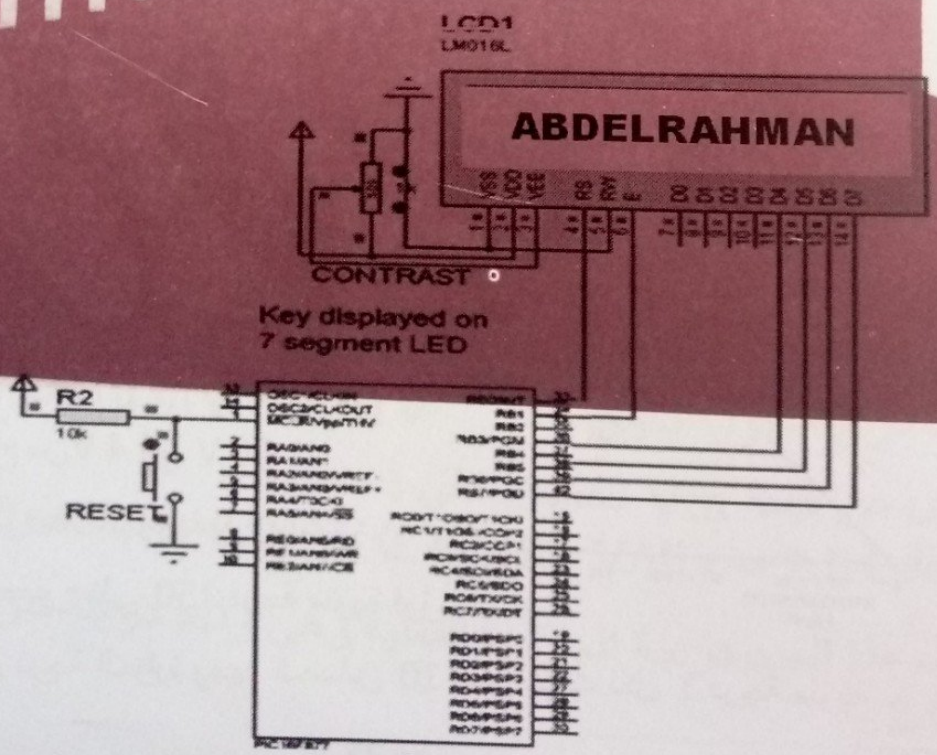
رابعا: المواجهة مع الأجهزة المتقدمة Advanced Component Interface :

١- المواجهة مع لوحة إظهار البيانات liquid Crgstal Display :

هي عبارة عن وحدة عرض الحروف مكونة من وحدات جاهزة التي تعتمد على المتحكم HD 4480 و تتميز أنها رخيصة الثمن و يسهل ربطها بالمتحكم لأنها تحتوي علي ناقل للبيانات و ناقل للتحكم .

- جدول يصف ال 16 طرف ل شاشة LCD :

Pin	Name	Descriptions
1	VSS	الارضى
2	VDD	الموجب
3	V0	تحديد مستوى التباين للشاشة وصلة لل موجب على طول
4	RS	تحديد سجلات داخل الشاشة
5	R/W	اختيار الكتابة على الشاشة ام القراءة منها
6	E	تفعيل البيانات
7	DB0	الرجل رقم 7 الى الرجل رقم 14 فهم اطراف الداتا التى سوف تتناقل بين البك والشاشة
.	.	
14	DB7	
15	BLA	طرف تشغيل الاضاءة الخلفية الطرف الموجب
16	BLK	طرف تشغيل الاضاءة الخلفية الطرف السالب

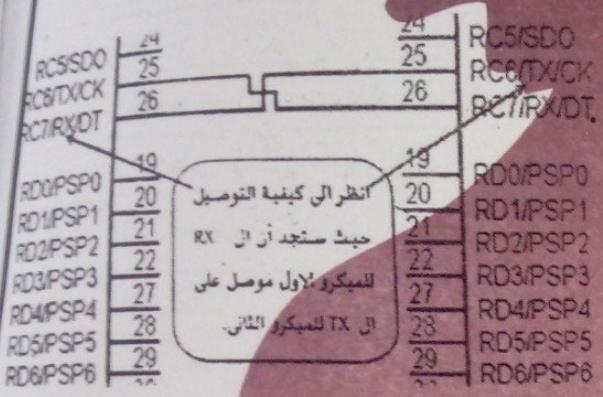


المواجهة مع وحدة الاتصال التسلسلي مع الكمبيوتر Serial Communication :

الاتصال التسلسلي: هو بروتوكول يستخدم لربط جهازين أو أكثر لنقل البيانات بطريقة تتابعية فيما بينهم من خلال وصلة بين الجهازين ، حيث يرسل البت الأول ثم الثاني و هكذا حتى يتم نقل البايت كاملاً .

- مشكلة عدم إمكانية الإرسال و الاستقبال في نفس الوقت علي سلك واحد فإما أن تقوم بالإرسال أو أن تقوم بالاستقبال ، لحل هذه المشكلة يتم وضع سلكين بين الجهازين يستخدم أحدهما للإرسال TX و الآخر للاستقبال Rx و بذلك يتم التوصيل بين جهازين ميكروكنترولر لنقل البيانات فيما بينهم .

سرعة نقل البيانات هي سرعة افتراضية في اي جهاز و هي (9600) بت في الثانية .



المواجهة مع وحدة الاتصال التسلسل مع الكمبيوتر:

يتم استخدام ثلاثة أسلاك لعملية الربط بين المتحكم والكمبيوتر.

TX: يستخدم لإرسال الاشارات من المتحكم إلي الكمبيوتر .

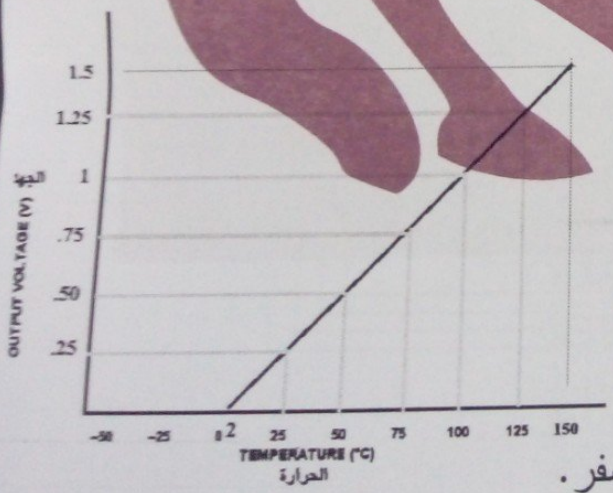
RX : يستخدم لإرسال الاشارات من الكمبيوتر إلي المتحكم .

GND : وهو مشترك بين الجهازين ويوصل بالأرضي .

بناء نظام حساس لقياس درجة الحرارة وإظهارها علي شاشة LCD :

هو نظام يقوم بقياس درجة الحرارة باستخدام الحساس (LM35) المتصل بطرف القناة AN1 مع توصيل مقاومة متغيرة بطرف القناة AN0 و يتم عرض قيمة نتيجة التحويل علي وحدة العرض LCD .

من **الحساس الحراري LM35** : هو عبارة عن دائرة متكاملة لحساس خطي لدرجة الحرارة قيمة جهده تتناسب مباشرة مع درجة الحرارة له جهد خرج 10mv/°C في مدى درجة حرارة يتراوح من (150 °c : 55°c-) .



مواصفات الحساس LM35 :

١) دقة الحساس 0.5 °c

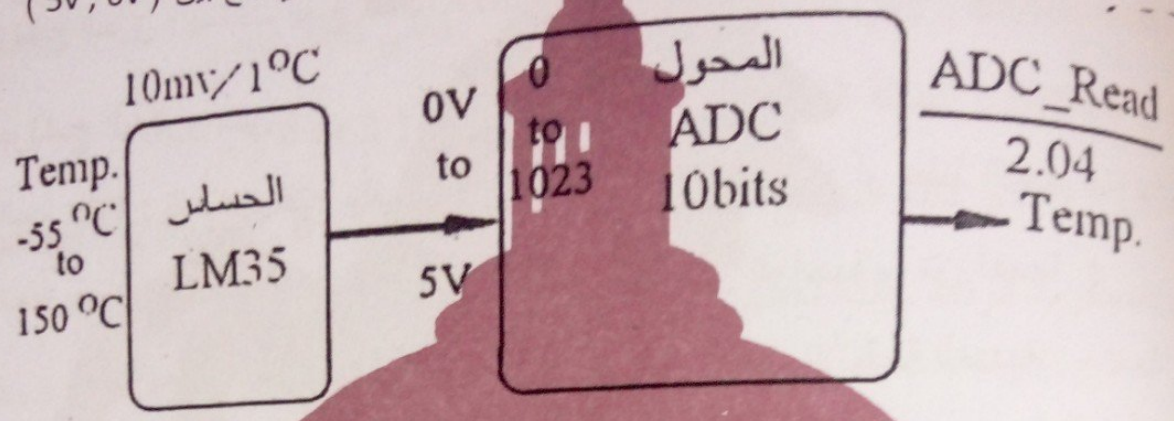
٢) جهد التغذية يتراوح من 4 V إلي 30 V

٣) درجات الحرارة التي يتعامل معها تتراوح من 55 درجة تحت الصفر غلي 150 درجة مئوية فوق الصفر.

٤) العلاقة خطية بين درجة الحرارة وجهد الحساس 10 مللي فولت لكل 1 درجة مئوية .

طرق تصحيح القراءة لتعبر عن درجة الحرارة بالدرجة المئوية :

نفترض أن الحساس Lm35 تم وضعه في مكان درجة حرارته (1 °C) فيكون جهد خرج الحساس هو (10mv) ويسمي الجهد Vin mv جهد دخل المحول ADC الذي يقوم بتحويله إلي عدد يقع بين 0 و 1023 ويسمي بالـ ADC Read قراءة المحول وفق جهد الدخل الذي يقع بين (5v , 0v)



شكل (2- 62)

2 - العلاقه الهامة الأولى هى دقة التحويل أو نسبة التحويل بين الجهد التناظرى والعدد الرقمى المقابل

5V ⟶ 1023 div

Vin V ⟶ ADC_Read div

حيث يمكن وضعها فى صورة نسبة وتناسب كما يلى :

5V / 1023 div = Vin V / ADC_Read div = 0. 00489 V/div

منها نحصل على القيمة العددية لنتيجة التحويل

ADC_Read div = Vin V /0. 00489 V/div

3 - من خواص الحساس

عند واحد درجة مئوية يكون Vin=10 mV = 0.01V

بالتعويض عن قيمة Vin نحصل على :

ADC_Read div=0.01V/ 0. 00489 V/div = 2.04 div

لكى يمثل هذا العدد درجة الحرارة الفعلية وهى واحد درجة مئوية يجب قسمته على 2.04

الباب الثالث : أساسيات البرمجة

البرامج المستخدمة مع الميكروكنترولر :

- **برنامج الميكرو سي :** يستخدم لكتابة الكود على الحاسب .
- **برنامج البروتس :** يستخدم لرسم الدوائر الإلكترونية و عمل محاكاة لها .
- **برنامج الحرق :** يستخدم لنسخ الملف HEX من الحاسب إلى ذاكرة الميكروكنترولر .

واجهة برنامج الميكرو سي :

- **قائمة File :** تستخدم لفتح ملف و إغلاقه و إنشاء ملف جديد أو حفظ التغييرات .
- **قائمة View :** تستخدم لإظهار و إخفاء المربعات في الواجهة و من أهمها :

  مربع الرسائل ٢- مربع التحويلات بين الأنظمة العددية .
- **قائمة Project :** تستخدم لعمل مشروع جديد و إغلاقه .
- **قائمة Build :** تستخدم لبناء كود البرنامج وظهور ملف HEX اذا كان الكود صحيح .

وصف المبرمجة QL-2006 : هي مبرمجة عالية السرعة مصممة خصيصا لبرمجة العديد من متحكمات PIC سلسلة PIC 10 / 12 / 16 / 18 .

مميزات المبرمجة QL-2006 :

- عالية الإداء بالنسبة للسعر .
- لها منفذ تسلسلي RS232 ومنفذ USB .
- سرعة أعلى من غيرها من المنتجات المماثلة .
- الأداء مستقر .
- دعم المزيد من الأجهزة .
- دعم الترقية على شبكة الإنترنت .
- مجهزة مقبس DIP 40 pin زيف سوكت .
- متوافقة مع أنظمة التشغيل OS المختلفة .

- دالة المفتاح :

Button (& Port , Pin , debounce time , active _ state) ;

& Port : إشارة للمنفذ المتصل به المفتاح .

Pin : الطرف المتصل به المفتاح و قيمته من 0 إلى 7 .

debounce time : زمن التأخير لمنع الارتدادات بالملي ثانية ms .

active _ state : الحالة الفعالة - عند الضغط عليه - (0 or 1) .

- **مثال ١ : فسر دالة المفتاح التالية :** **Button (& PortB , 0 , 10 , 0) ;**

المفتاح المطلوب قراءة حالته متصل بالمنفذ PortB بالطرف رقم 0 و هو RB0 و زمن التأخير لعلاج الارتدادات هو 10 ms و الحالة الفعالة - عند الضغط عليه – هي الحالة المنخفضة (0) .

- **مثال ٢ : فسر دالة المفتاح التالية :** **Button (& PortD , 5 , 20 , 1) ;**

المفتاح المطلوب قراءة حالته متصل بالمنفذ PortD بالطرف رقم 5 و هو RD5 و زمن التأخير لعلاج الارتدادات هو 20 ms و الحالة الفعالة - عند الضغط عليه – هي الحالة المرتفعة (1) .

- دالة **ADC – Init ( )** :

تستخدم لتهيئة ال ADC – interface كتحديد الزمن بين كل قيمة يتم قراءتها و الأخرى .

- دالة **ADC – Read (2)** :

تستخدم لقراءة قيم الإشارات التناظرية من أحد رجول الدخل الثمانية وهو الرقم بين الأقواس ، و الرجول من AN0 إلى AN7 .

مثال : فسر دوال المحول ADC من تماثلي إلى رقمي :

int temp ;

حجز متغير ثابت باسم temp .

ADC_init ;

تهيئة المحول للعمل .

Temp=ADC_Read (0) ; قراءة المحول للقيمة من الرجل AN0 و وضع هذه القيمة في المتغير temp .

مكتبة لوحة المفاتيح : توجد ثلاث دوال نحتاجها عند التعامل مع لوحة المفاتيح وهي كلاتي :

- **الدالة الأولى :**

يستخدم هذا الأمر لأعلام الميكرو بالمنفذ المتصل عليه لوحة المفاتيح ويكتب قبل ال main .

```
Char keypadport at portB    ;
```

- **الدالة الثانية :**

تستخدم لتهيئة مخرج الميكرو للاستخدام مع لوحة المفاتيح و تكتب داخل ال main .

```
keypad_init ( )    ;
```

- **الدالة الثالثة :**

دالة تخبرنا بقراءة قيمة الزر الذي تم الضغط عليه .

عبارة عن متغير نستقبل فيه قيمة الزر

```
Char kp    ;
Kp =  keypad_Key_Click  ( )    ;
```

تستخدم لتخزين قيمة الزر في المتغير kp عند الضغط على الزر .

- **وهناك دالة أخرى لقراءة قيمة الزر :**

عبارة عن متغير نستقبل فيه قيمة الزر

```
Char kp    ;
Kp =  keypad_Key_Press  ( )    ;
```

تستخدم لتخزين قيمة الزر في المتغير kp عند الضغط على الزر .

- **و الفرق بين الدالتين :**

أن **الدالة الثانية** : لا تنتظر حتى ترفع يدك من على الزر ، بل بمجرد الضغط عليه ترسل القيمة للبرنامج و يكمل البرنامج تنفيذ أوامره بعد الدالة .

أما **الدالة الاولى** : لا تأخذ القيمة إلا بعد رفع يدك من على الأزرار المضغوط عليها ، ثم بعد ذلك تأخذ فقط أول قيمة قمت بالضغط عليها .

مكتبة وحدة شاشة الكريستال السائل LCD :

توجد ست دوال نحتاجها عند التعامل مع وحدة شاشة الكريستال السائل LCD .

أولا : مجموعة المتغيرات :

عبارة عن مجموعة أوامر يتم كتابتها في بداية كل برنامج يتعامل مع شاشة LCD وقبل الدالة الرئيسية له main و تستخدم لإعلام الميكرو بالرجول المتصلة بالشاشة تقوم بوضع اسمائها في هذه المتغيرات عن طريق مجموعة أوامر .

- **المجموعة الأولى من المتغيرات :**

Sbit LCD_RS at RC2_bit ;

هذا الأمر يخبر الدالة أن الطرف **RS** الموجود في الشاشة متصل بالرجل رقم **2** في المنفذ C .

Sbit LCD_EN at RC3_bit ;

هذا الأمر يخبر الدالة أن الطرف **EN** الموجود في الشاشة متصل بالرجل رقم **3** في المنفذ C .

Sbit LCD_D4 at RC4_bit ;

هذا الأمر يخبر الدالة أن الطرف **D4** الموجود في الشاشة متصل بالرجل رقم **4** في المنفذ C .

Sbit LCD_D5 at RC5_bit ;

هذا الأمر يخبر الدالة أن الطرف **D5** الموجود في الشاشة متصل بالرجل رقم **5** في المنفذ C .

Sbit LCD_D6 at RC6_bit ;

هذا الأمر يخبر الدالة أن الطرف **D6** الموجود في الشاشة متصل بالرجل رقم **6** في المنفذ C .

Sbit LCD_D7 at RC7_bit ;

هذا الأمر يخبر الدالة أن الطرف **D7** الموجود في الشاشة متصل بالرجل رقم **7** في المنفذ C .

- **المجموعة الثانية من المتغيرات :**

تستخدم لتحديد اتجاه الداتا على الأرجل المستخدمة مع الشاشة باستخدام الأمر TRIS ، وهذا سيكون ب ستة أوامر أيضا مشابهة للأوامر السابقة و هي كلاتي :-

Sbit LCD_RS_Direction at TRISC2_bit ;

هذا الأمر يخبر أن اتجاه الداتا من الميكرو إلى الطرف **RS** الموجود في الشاشة متصل بالرجل رقم **2** في المنفذ C .

Sbit LCD_EN_Direction at TRISC3 bit ;

هذا الأمر يخبر أن اتجاه الداتا من الميكرو إلى الطرف **EN** الموجود في الشاشة و المتصل بالرجل رقم **3** في المنفذ C .

Sbit LCD_D4_Direction at TRISC4 bit ;

هذا الأمر يخبر أن اتجاه الداتا من الميكرو إلى الطرف **D4** الموجود في الشاشة و المتصل بالرجل رقم **4** في المنفذ C .

Sbit LCD_D5_Direction at TRISC5 bit ;

هذا الأمر يخبر أن اتجاه الداتا من الميكرو إلى الطرف **D5** الموجود في الشاشة و المتصل بالرجل رقم **5** في المنفذ C .

Sbit LCD_D6_Direction at TRISC6 bit ;

هذا الأمر يخبر أن اتجاه الداتا من الميكرو إلى الطرف **D6** الموجود في الشاشة و المتصل بالرجل رقم **6** في المنفذ C .

Sbit LCD_D7_Direction at TRISC7 bit ;

هذا الأمر يخبر أن اتجاه الداتا من الميكرو إلى الطرف **D7** الموجود في الشاشة و المتصل بالرجل رقم **7** في المنفذ C .

ثانيا : مجموعة الدوال :

- **الدالة الأولى** **Led_Init ( )** ;

ويتم كتابتها داخل الدالة الرئيسية main و قبل ال while و فائدتها تجهيز الموديول الذي تعامل مع الشاشة داخل الميكرو .

- **الدالة الثانية** **Led_out (1,8,"Abdelrhman")** ;

رقم الصف — رقم العمود — السلسلة الحرفية

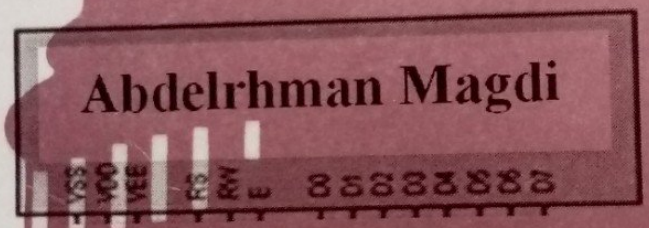
تستخدم لإخراج سلسلة حرفية أو جملة على الشاشة كما موضح :

- **الدالة الثالثة** **Led_out_Cp ("Magdi")** ;

تستخدم هذه الدالة للكتابة عند آخر حرف انتهينا منه ، وبالتالي لو كتبنا الأمرين التالين :

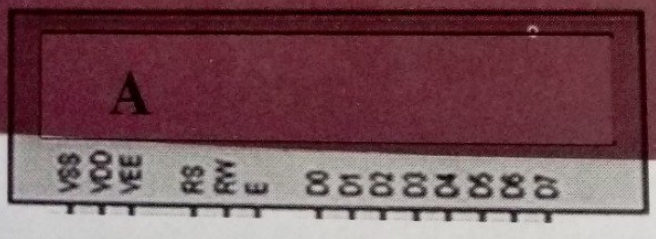
```
Led_out ( 1,8,"Abdelrhman" )      ;
Led_out_Cp ("Magdi" )             ;
```

سيكون الخرج كالآتي :



- **الدالة الرابعة** **Led_Char (1,1,"A")** ;

تستخدم هذه الدالة لإظهار حرف واحد فقط على الشاشة ، وبالتالي لو كتبنا هذه الأمر سيظهر:

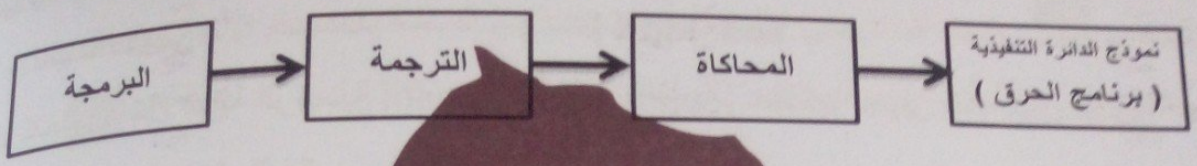


- **الدالة الخامسة** **Led_Cmd (_LCD_CURSOR_OFF)** ;

تستخدم هذه الدالة لنقل أوامر الشاشة لكي تقوم بتنفيذها ، وهذا الأمر لإلغاء ال CURSOR .

و لمسح الشاشة نكتب الأمر الآتي : **Led_Cmd (_LCD_CLEAR)** ;

مراحل تمثيل و محاكاة الدوائر :



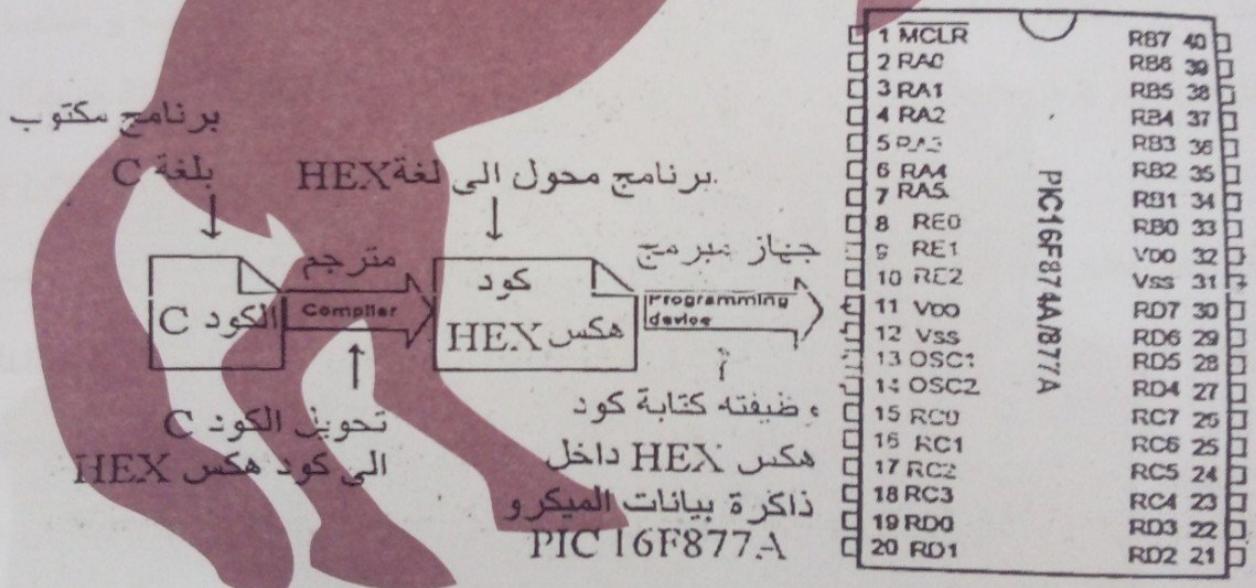
١- **البرمجة :** عبارة عن برنامج يكتب على الحاسب شخصي **PC** بلغة عالية المستوى كلغة C

٢- **الترجمة :** إذا لم يكن في البرنامج أخطاء ، سيتم إنشاء ملف سداسي عشر Hex File وهذا الملف يعرف بالكود أو شفرة الآلة .

٣- **المحاكاة :** باستخدام برنامج بروتس .

٤- **برنامج الحرق :** يستخدم لحفظ ال Hex File في ذاكرة الميكرو خلال جهاز (المبرمجة) ويتم اختبار عمل الدائرة على نموذج الدائرة التنفيذية سواء على لوحة التجارب أو المطبوعة .

المراحل المتبعة لتحويل لغة برمجة عالية C إلى لغة برمجة منخفضة HEX :



خطوات تمثيل و محاكاة دائرة إلكترونية باستخدام برنامج ISIS :

١- إضافة المكونات الإلكترونية المتاحة .

٢- وضع العناصر الإلكترونية في المكان المخصص لها في التصميم .

٣- توصيل المكونات باستخدام الماوس .

٤- تشغيل و ايقاف المحاكاة .

الأجهزة و الأدوات التي نحتاجها لبرمجة المتحكم الدقيق PIC 16F877A :

١-لوحة تجارب bread board لتجميع الدائرة و اختبارها .

٢-لوح لحام الدائرة المطبوعة لبناء و اختبار الدوائر التجريبية .

٣-رقاقة المتحكم الدقيق .

٤-مبرمج المتحكم الدقيق .

٥-جهاز كمبيوتر و مجموعة برامج لعمل المحاكاة و البرمجة و حرق البرنامج .

٦-مصدر جهد مستمر ومنظم للتغذية .

٧-ملتيميتر رقمي .

٨-مكونات أخرى مثل محرك الخطوة و المقاومات و LEDs و المكثفات و الأسلاك .

الخطوات اللازمة لإجراء التجارب باستخدام المتحكم الدقيق :

١-كتابة كود البرنامج بلغة الميكرو سي .

٢-تصميم الدائرة بواسطة برنامج البروتس .

٣-محاكاة الدائرة للتأكد من أنها تعمل بشكل جيد .

٤-تصميم الدائرة المطبوعة .

٥-حقن البرنامج في الميكرو كونترولر .

كيف يتم حقن البرنامج HEX في الميكرو كونترولر :

١-يتم كتابة البرنامج بالميكروسي واختباره و بنائه و تكوين ملف HEX .

٢-يحمل الملف عن طريق برنامج QL-PROG الذي يعرف المبرمجة و ينقل البرنامج الى المبرمجة .

٣-توصل المبرمجة بأحد منافذ الكمبيوتر .

٤-نقوم بتحميل ملف HEX بواسطة برنامج QL-PROG من الكمبيوتر الى ذاكرة المتحكم .

الباب الرابع : اساسيات برمجه المتحكم الدقيق بلغة السي

البرنامج المسئول عن ترجمة برنامج مكتوب بلغات المستوى العالي إلى لغة الآلة يسمى المترجم Compiler حيث يتم تمثيل الكود القابل للتنفيذ كتسلسل من الأعداد بنظام السداسي عشر ويسمى ملف الهكس HEX Code.

الهيكل الأساسي لبرنامج مكتوب بلغة السي

التعليقات Comments

⇩

المتغيرات Variables

⇩

تعريف الدوال
Function Prototypes

⇩

الدالة الأساسية
Main Function

⇩

الدوال Functions

أنواع البيانات في لغة الميكرو سي :

١- الأحرف Characters .
٢- الأعداد الصحيحة Integers .
٣- الأعداد العشرية Float .
٤- الكلمات المحجوزة من أجل تعريف الدوال .

أولا : التعليقات Comments :

وهي جزء من البرنامج وتستخدم لتوضيح عمل البرنامج ، ويتم تجاهل التعليقات ولا يتم ترجمتها من قبل المترجم ، ويوجد منها نوعان :

التعليق القصير ويبدأ بالعلامة //

التعليق الطويل ويبدأ بالعلامة */ وينتهي بالعلامة /*

ثانياً : المتغيرات Variables:

وهو أي عدد تتغير قيمته أثناء تنفيذ البرنامج. ويجب أن يتم تعريف المتغير قبل أن يتم استخدامه لأول مرة في البرنامج حيث يتعرف المترجم عليها كسجلات في ذاكرة RAM حيث يتم حجز حيز لها حسب النوع.

ويتم تعريف المتغير بـ **الاسم ، النوع**

اسم المتغير Variable Name :

يمكن أن يشمل الاسم أي من الحروف الأبجدية (A-Z) ، (a-z) والأرقام 0-9 ويتم تخصيص قيم أولية في بداية البرنامج. لكن غير مسموح بتسمية المتغير باسم أي مخزن في البرنامج مثل : do, for, case, float, break, signed, void, double, while, if.

نوع المتغير Variable Type:

وهو الجزء الذي يحدد نواع وقيمة المتغير وحجمه في الذاكرة RAM وينقسم إلى :-

نوع المتغير	نوع البيانات بالبادئة		التعبير عن الأعداد	الحجم في ذاكرة RAM	المـدى
Char	Unsigned		الصحيحة الموجبة	8 bits	0 : 255
	Signed		الصحيحة الموجبة والسالبة	8 bits	-128 : 128
Int	Short	Unsigned	الصحيحة الموجبة	8 bits	0 : 255
		Signed	الصحيحة الموجبة والسالبة	8 bits	-128 : 128
	Int	Unsigned	الصحيحة الموجبة	16 bits	0 : 65535
		Signed	الصحيحة الموجبة والسالبة	16 bits	-32768 : 32768
	Long	Unsigned	الصحيحة الموجبة	32 bits	
		Signed	الصحيحة الموجبة والسالبة	32 bits	
Float			الأعداد العشرية	32 bits	

مثال : ما المقصود بجمل البرنامج الآتية :

```
Char x                ;
Unsigned int gate1    ;
Signed int start, sum ;
gate1 =20             ;
```

Char x;: يعني أننا حجزنا مكان في الذاكرة اسمه x ، لتخزين الأعداد الصحيحة الموجبة سعته ٨ بت.

Unsigned int gate1; : يعني أننا حجزنا مكان في الذاكرة اسمه gate1 لتخزين الأعداد الصحيحة الموجبة سعته ١٦ بت.

Signed int start, sum; : يعني أننا حجزنا مكانين في الذاكرة اسمهما start و sum لتخزين الأعداد الصحيحة الموجبة والسالبة سعة كل منهما ١٦ بت .

gate1 = 20; : يعني تخزين العدد الصحيح الموجب ٢٠ كـقيمة أولية في المتغير المسمى gate1 .

ثالثاً : الثوابت Constants :

الثابت هو رقم أو حرف له قيمة ثابتة ولا يمكن تغيير قيمتها أثناء تنفيذ البرنامج ويتم حجز مكان لها في ذاكرة البرنامج Program Memory ، ويتعرف غليه المترجم باسمه مسبوقاً بكلمة const .

أنواع الثوابت :

١- ثوابت العدد الصحيح Integer Constants .

٢- ثوابت الأعداد العشرية Floating Point Constants .

٣- الثوابت الحرفية Character Constants .

٤- ثوابت السلسلة النصية String Constants .

ثوابت العدد الصحيح :

ثوابت العدد الصحصح يمكن أن تكون عشرية أو ثمانية أو ثنائية أو سداسي عشرية مثل :-

نوع العدد الصحيح	البادئة	مثال
عشري		Const MAX= 100
سداسي عشر	0X	Const MAX= 0XED
ثماني	0	Const MAX= 016
ثنائي	0B	Const MAX= 0B11001100

مثال : ما المقصود بالمثال البرمجي الآتي : **Const Minimum= -100**

يعني أننا حجزنا مكان بالذاكرة اسمه Minimum لتخزين العدد العشري 100- فيه.

ثوابت الأعداد الحقيقية :

وتتألف من جزء صحيح وعلامة عشرية ثم الجزء الكسر أو الحرف متبوع بـ e أو E والذي يمثل الأس.

مثال: ما المقصود بالمثال البرمجي الآتي :

1- Const T_Max= 32.6 ،،،،،،،،، 2- Const T_Max= 0.326E2

١- يعني أننا قمنا بتخزين العدد الحقيقي 32.6 بالصورة العشرية في مكان اسمه T_Max .

٢- يعني أننا قمنا بتخزين العدد الحقيقي 32.6 بالصورة الأسية في مكان اسمه T_Max .

الثوابت الحرفية :

الثابت الحرفي هو حرف يوضع بين علامتي اقتباس مفردة ' '

ثوابت السلسلة النصية :

ويتكون من سلسلة من الحروف تسمى سلسلة نصية ويتم وضعها بين علامتي اقتباس مزدوجة " "

مثال : ما المقصود بالمثال البرمجي الآتي :

1- Const I_Class= 'A' ;

2- Const Message="Press The Left Button" ;

١- تم الإعلان عن ثابت حرفي اسمه I_Class كحرف A .

٢- تم إعلان الشاشة بالرسالة (النص) Press The Left Button .

رابعاً : الدوال Functions

١- أوامر الإدخال والإخراج .

٢- دالة التأخير .

٣- عبارات الشرط .

٤- العبارات الحلقية .

الدالة الخالية (الفارغة) Void

هي دالة تستخدم لتغيير حالة أطراف مخارج المتحكم خلال نقل البيانات ، وعند كتابة البيانات على وحدة العرض LCD. ويتعرف عليها المترجم على مثل هذه الدوال بواسطة نوع نتيجتها والذي يحدد بأنه فارغ أو خالي من العائد Void. كما في الكود التالي :

```
void main ( )
{
    Commands
}
```

أوامر الإدخال والإخراج :

لكل منفذ في المتحكم مسجلين هما :

PORT : ويقوم بتحديد قيمة الإشارة المنطقية على أطراف المنفذ إما عالية (+5 v)

وتأخذ القيمة المنطقية 1 أو منخفضة (0 v) وتأخذ القيمة المنطقية 0 .

TRIS : يتحكم في اتجاه انتقال البيانات من أو إلى المتحكم ، أي يحدد اما أن تكون أطراف المنفذ مدخلات

أو مخرجات. لتحديد المنفذ كـ خرج يوضع 0 ،، ولتحديد المنفذ كـ دخل يوضع 1 .

مثال : اشرح الكود التالي :

```
void main () //
  {
TrisD=0        ; //
PortD= oxFF ;  // .+5v
  }
```

الدالة الرئيسية وتسمى الدالة الفارغة .

القوس المتعرج المفتوح يبين بداية البرنامج الرئيسي .

تهيئة جميع أطراف المنفذ D كمخارج .

جعل جميع أطراف المنفذ D في الحالة المرتفعة أي عند جهد +5v.

القوس المتعرج المغلق يبين نهاية البرنامج .

الوصول إلى بت فردية (إرسال أمر إلى طرف من الأطراف دون مخاطبة باقي أطراف المنفذ).

جعل الطرف رقم ٥ في منفذ A في الحالة المرتفعة . **PortA.A5=1; //**

جعل الطرف رقم ١ في منفذ b في الحالة المنخفضة . **PortB.f1=0; //**

تهيئة الطرف رقم 2 في منفذ C كـ مخرج . **TrisC.C2=0; //**

تهيئة الطرف رقم 5 في منفذ D كـ مدخل . **TrisD.f5=1; //**

دالة التأخير :

هي دالة داخلية تستخدم في توليد تأخير زمني بالمللي ثانية أو الميكرو ثانية. وصيغتها كالتالي :

عمل تأخير زمني قدره ١ ثانية. **Delay_ms(1000) ;**

عمل تأخير زمني قدره ٢ ثانية . **Delay_us(2000000) ;**

عبارات الشرط : Conditional Operators

هي عنصر شائع الاستخدام في البرنامج ، وتمكننا من تنفيذ عملية من بين عدة عمليات ، بمعنى آخر إذا تحقق الشرط ___ ، افعل ___ وإذا لم يتحقق الشرط ، افعل ___.

أ- عبارة الشرط IF

```
If ( expression ) Operation
```

إذا تحقق شرط العبارة (expression) يتم تنفيذ العملية (operation) ثم يتقدم البرنامج في التنفيذ .
و إذا لم يتحقق الشرط لا يتم تنفيذ العملية .

ب- عبارة الشرط If-Else

```
If ( expression ) Operation 1
Else              Operation 2
```

إذا تحقق شرط العبارة (expression) يتم تنفيذ العملية (operation 1) ثم يتقدم البرنامج في التنفيذ .
وإذا لم يتحقق الشرط يتم تنفيذ العملية الأخرى (operation 2) و يتقدم البرنامج في التنفيذ.

جـ - عبارة Switch

عبارة سويتش تمكنك من الاختيار بين عمليات متعددة .

```
Switch ( expression )
  {
Case Const 1 :
Operation 1
    Break ;
Case Const 2 :
Operation 2
    Break;
Default :
Expected_Operation
    Break ;
  }
```

يتم تنفيذ الشرط expression

يتم مقارنة الثابت الأول Cons1 بالشرط expression
وإذا وجد تطابق سيتم تنفيذ العملية ١
وحتى العثور على كلمة break
يتم مقارنة الثابت الثاني Cons2 بالشرط expression
وإذا وجد تطابق سيتم تنفيذ العملية ٢
وحتى العثور على كلمة break
وإذا لم يعثر على أي تطابق مع أي ثابت يتم تنفيذ عملية افتراضية

العبارات الحلقيه LOOP Operation

أ- التكرار اللانهائي باستخدام الأمر go to Lable :

يمكننا هذا الأمر من التكرار اللانهائي للبرنامج أو لمجموعة من الأوامر.

```
void main ( )
   {
TrisC=0   ;            PortC=0;
  Loop:
portC=0b00000001 ;  delay_us(100000) ;
portC=0b00000010 ;  delay_us(100000) ;
portC=0b00000100 ;  delay_us(100000) ;
portC=0b00001000 ;  delay_us(100000) ;
 goto Loop;
  }
```

برنامج لإضاءة ٤ ليدات متصلة بالأطراف الأول والثاني و الثالث والرابع لمنفذ C والتكرار باستخدام أمر العبارة الحلقية go to Lable

ب- التكرار باستخدام أمر While :

```
While (expression)
   {
Commands
      .
   }
```

يتم تنفيذ الأوامر Commands بشكل متكرر حتى يصبح التعبير أو الشرط expression غير متحقق

Example :

```
Int A=0 ;
While (A<10)
   {
   A++
   }
```

تم الإعلان عن متغير اسمه A وتم تحديد قيمة ابتدائية له ، وسيتم تنفيذ الحلقة لعدد ١٠ مرات.
حيث يقوم البرنامج باختبار صحة الشرط بين القوسين وإذا وجده صحيحاً يقوم بتنفيذ التعبير أو العملية وهو زيادة قيمة المتغير A بمقدار ١ فتتكرر الحلقة إلى أن يصبح الشرط غير متحقق فيخرج البرنامج من الحلقة .

حلقة While (1) اللانهائية :

يتم تشكيل هذه الحلقة بعدم وضع متغير بين القوسين ونضع مكانه ١ أي يكون الشرط غير متغير ويتم تكرار العملية بصورة لانهائية .

```
While (1)
   {
Commands
      .
   }
```

جـ حلقة do-while :

في هذه الحالة يتم تنفيذ العملية operation مرة واحدة على الأقل بغض النظر عما إذا كان الشرط متحقق أم لا ، لأن تعبير الشرط يتم التحقق منه في نهاية الحلقة. وإذا لم تكن النتيجة صفر أي الشرط متحقق تكرر هذه الخطوات .

```
Int a = 0 ;
do
{ a = a + 1 ; }
While (a<=1E06);
```

تم الإعلان عن متغير اسمه a وتم تحديد قيمة ابتدائية له ، وبمجرد أن يجد البرنامج كلمة do سيقوم بتنفيذ الأوامر التي تتبعها داخل الأقواس المتعرجة والتحقق من قيم متغير الحلقة والذي يحدد تكرار الحلقة أو الخروج منها ، وهو هنا عبارة عن إضافة واحد للمتغير a ليصبح a=1 .

وبعد التنفيذ ينتبه البرنامج لكلمة while فيقوم بتقييم أو اختبار الشرط بين القوسين وهو أن تكون قيمة المتغير a أقل من أو تساوي ١× ١٠ أس ٦

د- حلقة FOR :

تستخدم حلقة for لتكرار عملية أو تعبير معين لعدد من المرات .

```
For ( initial_expression ; Condition_expression ; Change_expression )
    { Operation }
```

يتم تحديد قيمة أولية لمتغير الحلقة condition_expression ثم يتم تنفيذ العملية operation بشكل متكرر و بعد كل تكرار تتغير قيمة التعبير change_expression و يستمر التكرار إلى أن يصبح condition_expression الشرط غير متحقق .

مثال :

```
For ( K = 0 ; K < 5 ; K++ )
  { Operation }...
```

- تم الإعلان عن متغير اسمه K وتم تحديد قيمة أولية له .
- يتم تقييم القيمة الأولية للمتغير الموجودة بالتعبير الأول K= 0 بالشرط الموجود في التعبير الثاني K< 5 ثم يقوم بتنفيذ التعبير الذي يليه K++ (تزايد قيمة المتغير K بمقدار ١) عند كل تكرار .
- يقيم البرنامج هذا التعبير الشرطي فإن تحقق الشرط يتم تنفيذ العملية وتكرر الحلقة من جديد حتى يصبح الشرط غير متحقق وهو أن تصبح قيمة المتغير K=5 وعندها يخرج البرنامج من الحلقة .

١- أكتب برنامج يضئ الليدات الموصلة بالمنفذ B ، ثم انتظار لمدة نصف ثانية ، ثم إطفاء جميع الليدات أكتب هذا البرنامج مرتين ، مرة ينفذ فيها البك مرة واحدة دون تكرار ، ومرة مع التكرار الدائم .

```
void main ( )
{
    TrisB = 0          ;
    PortB = 0          ;
    PortB = 0X FF      ;
    Delay_ms ( 500 )   ;
    PortB = 0          ;
}
```

- - - - - - - - -

```
void main ( )
{
    TrisB = 0          ;
    PortB = 0          ;
Loop :
    PortB = 0X FF      ;
    Delay_ms ( 500 )   ;
    PortB = 0          ;
go to Loop             ;
}
```

٢- أكتب برنامج يجعل أول طرفين في المنفذ A مخارج ، و باقي الأطراف مداخل .

```
void main ( )
{
    TrisA = 0B 111100 ;
}
```

٣-أكتب برنامج يجعل البك يخرج 5V عن طريق الرجول b0 , b3 , b6 ، ثم ينتظر لمدة 200 ms ، ثم يقوم بإطفاء هذه الليدات ، ثم يخرج 5V عن طريق الرجول b1 , b2 , b4 ، ثم ينتظر لمدة 800 ms يتم تكرار تنفيذ البرنامج باستمرار .

```
void main ( )
{
    TrisB = 0              ;
    PortB = 0              ;
Loop :
    PortB = 0B 01001001    ;
    Delay_ms ( 200 )       ;
    PortB = 0              ;
    PortB = 0B 00010110    ;
    Delay_ms ( 800 )       ;
go to Loop                 ;
}
```

٤-ترتبط 8 Leds بأطراف المنفذ B من المتحكم Pic 16F84A عبر مقاومة مناسبة لتحد من التيار .

أكتب برنامج للعد بالنظام الثنائي من 0 إلى 255 و عرض الخرج على Leds ،

أضف زمن تأخير 500 ms بين كل خرج .

```
void main ( )
{   int X              ;
    TrisB = 0          ;
    PortB = 0          ;
    Delay_ms ( 500 ) ;
    For ( X = 0  ;  X < 255  ;  X++ )
    {  PortB++         ;   Delay_ms ( 500 ) ;  }
}
```

٥- أكتب برنامج لتكرار إضاءة الليدات الزوجية الموصلة بالمخرج C بالتتابع باستمرار بفاصل زمني 750 ms .

```
void main ( )
{
    TrisC = 0              ;
    PortC = 0              ;
While (1) {
    PortC = 0B 00000001    ;
    Delay_ms ( 750 )       ;
    PortC = 0B 00000101    ;
    Delay_ms ( 750 )       ;
    PortC = 0B 00010101    ;
    Delay_ms ( 750 )       ;
    PortC = 0B 01010101    ;
    Delay_ms ( 750 )       ;
} }
```

٦- في البرنامج التالي اشرح ما سينفذه المتحكم (أشرح الأوامر .. أمر أمر) :

void main ()	الدالة الرئيسية وتسمى الدالة الفارغة .
{	القوس المتعرج المفتوح يبين بداية البرنامج الرئيسي .
TrisB=0 ;	تهيئة جميع أطراف المنفذ B كمخارج .
PortB=0 ;	جعل جميع أطراف المنفذ B في الحالة المنخفضة أي عند جهد 0v .
PortB= oXFF ;	جعل جميع أطراف المنفذ B في الحالة المنخفضة أي عند جهد 5v .
Delay_ms(250) ;	عمل تأخير زمني قدره 250 ملي ثانية.
}	القوس المتعرج المغلق يبين نهاية البرنامج .

١٠ اكتب برنامج لإضاءة الليد المشع على الطرف 2 في المخرج D ، ثم إطفاءه ، ثم إضاءة الليد على الطرف 4 ، ثم إضاءة الليدان معا 20 مرة .

```
void main ( )
{
    int X                    ;
    TrisD = 0                ;
    PortD = 0                ;
    PortD = 0B 00000100      ;
    Delay_ms ( 500 )         ;
    PortD = 0                ;
    PortD = 0B 00010000      ;
   For ( X = 0 ; X < 20 ; X++ )
    {
    PortD = 0B 00010100      ;
    Delay_ms ( 500 )         ;
    PortD = 0                ;
    Delay_ms ( 500 )         ;
    }
}
```

٨-ترتبط 8 Leds بأطراف المنفذ B من المتحكم Pic 16F84A عبر مقاومة مناسبة لتحد من التيار .
و تم ترتيب Leds بطريقة دائرية ، أكتب الجمل التي تصف كيف يمكن تشغيل Leds بنظام إضاءة متتابع لليدات الثمانية ، ثم عمل وميض سبع مرات لجميع الليدات ثم تكرار كل ما سبق باستمرار .
استخدم فاصل زمني مناسب .

```
void main ( )
{
    TrisB = 0          ;          PortB = 0          ;
Loop :
    PortB = 0B 00000001   ;       Delay_ms ( 500 )   ;
    PortB = 0B 00000011   ;       Delay_ms ( 500 )   ;
    PortB = 0B 00000111   ;       Delay_ms ( 500 )   ;
    PortB = 0B 00001111   ;       Delay_ms ( 500 )   ;
    PortB = 0B 00011111   ;       Delay_ms ( 500 )   ;
    PortB = 0B 00111111   ;       Delay_ms ( 500 )   ;
    PortB = 0B 01111111   ;       Delay_ms ( 500 )   ;
    PortB = 0B 11111111   ;       Delay_ms ( 500 )   ;
For ( X = 0   ;   X < 20   ;  X++ )
  {
    PortB = 0B 11111111   ;       Delay_ms ( 500 )   ;
    PortB =  0            ;       Delay_ms ( 500 )   ;
  }
go to Loop ;
}
```