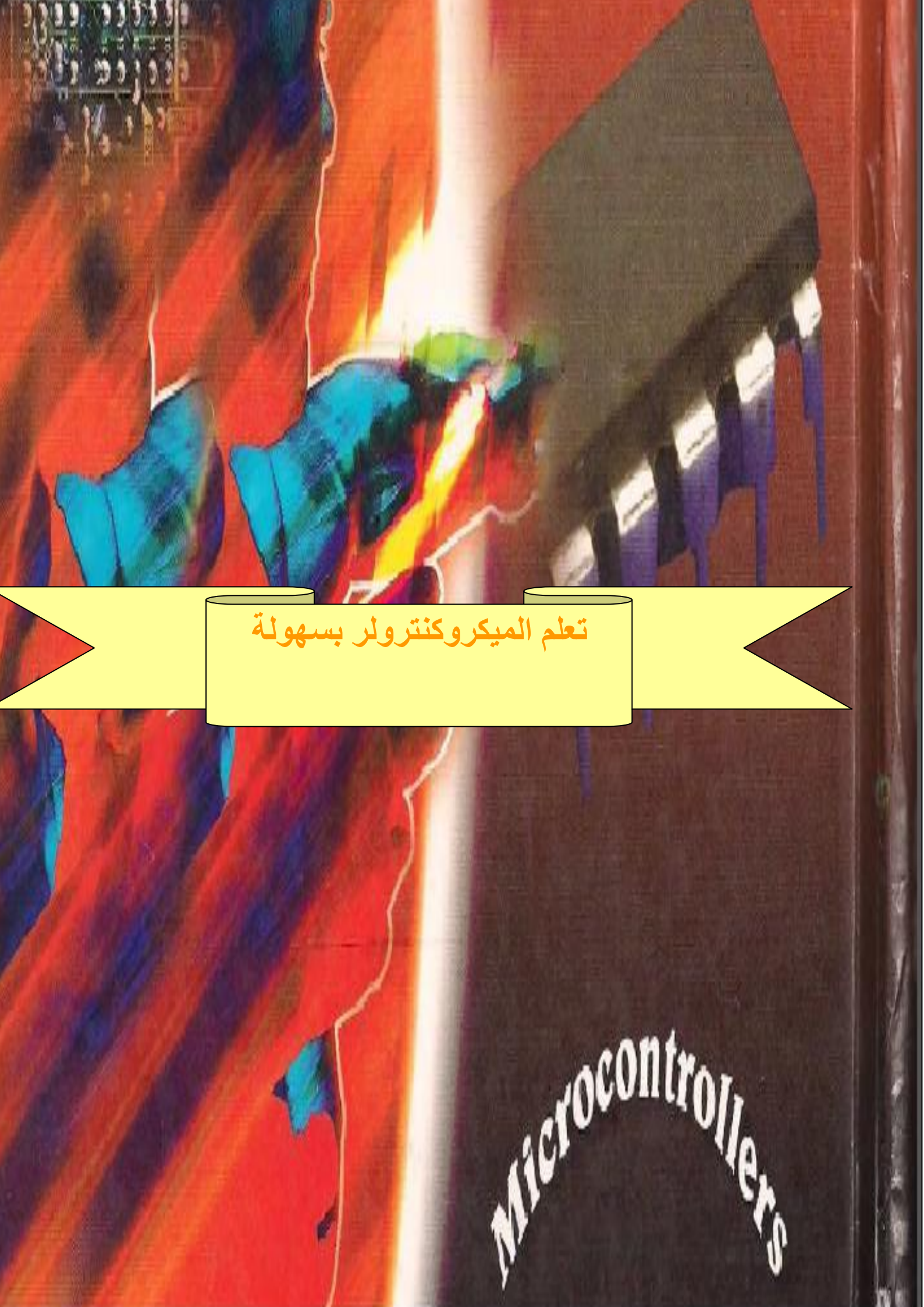

تعلم الميكروكنترولر بسهولة
Microcontrollers

**بسم الله الرحمن الرحيم**

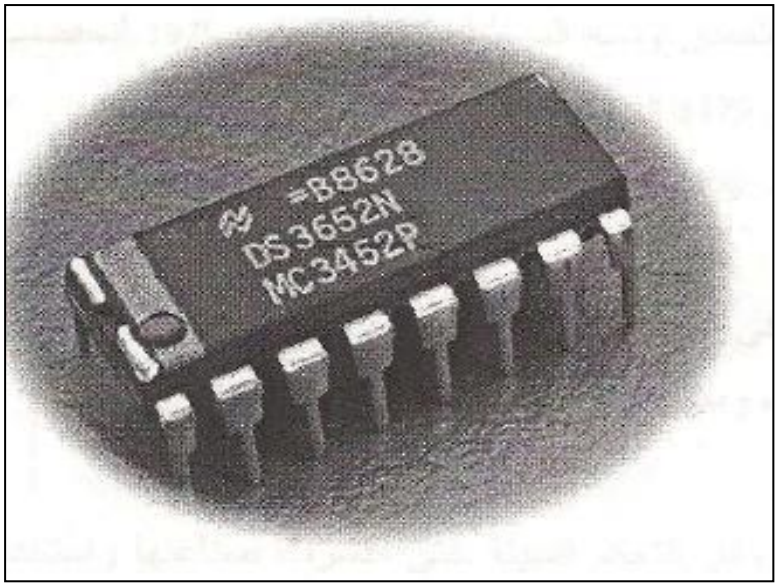
# تعلم الميكروكنترولر بسهولة

مقدمة :

مما لا شك فيه أن الميكرو كنترولر أصبح من اهم سمات عصر الاتصالات وتتنافس الشركات الكبرى في تصنيعة ومن اهم هذه الشركات ميكروشيب وتوشيبا وانتيل واتميل والعديد من الشركات ولكن ما هو الميكروكنترولر سوف تفاجأ أن الميكروكنترولر يشبه لحد كبير الحاسب الالي ولكنه حاسب صغير حيث له مدخلات ومخرجات ووحدة معالجة ويستخدم كثيرا في تكيفات الهواء حيث يتحكم في درجات الحرارة ويستخدم ايضا في الربوت الانسان الالي ويتحكم في المحركات الكهربية ويدخل في إلكترونيات السيارات الحديثة بشكل اساسي وحتى لا اطيل عليكم في المقدمة اكتفي بهذا القدر ولندخل سويا الى عالم الميكروكنترولر لنكتشف اسراره وخفاياه

مؤلف الكتاب :::: استاذ عيد فتحي

# الفصل الاول
# تركيب الميكروكنترولر




كما في الصورة يشبه الميكرو إلى حد كبير IC أو الدوائر المتكاملة ولكن يشبه في تركيبه جهاز الحاسوب حيث يحتوي على

1- معالج صغير يطلق عليه microprocessor يقوم بمعالجة البيانات وعمل العمليات الحسابية كالجمع والطرح والضرب والقسمة والعلميات المنطقية مثل OR و AND و NOT

2- رامات RAM وهي ذاكرة الوصول العشوائي وتشبه لحد كبير رامات الكمبيوتر إلا انها صغيرة الحجم وصغيرة المساحة أيضا وتستخدم لتخزين بيانات البرامج الجاري تنفيذها حاليا وتفقد محتواها عند انقطاع التيار الكهربي

3- EEPROM :: وهي الذاكرة الدائمة في الميكروكنترولر وتشبه الذاكرة ROM في الحاسب إلا انها يمكن ان تمحو محتوياتها من خلال المبرمجة " بروجرمر " لعدة مرات ولا تفقد محتواها عند انقطاع التيار الكهربي

ملحوظة
اذا انقطع التيار الكهربي أثناء شحن الميكروكنترولر فان هذا يؤدي الى تلف الميكروكنترولر

4- الفلاش ميموري flash memory : وهو يشبه القرص الصلب او الهارد ديسك حيث يتم تخزين البرامج عليه

5- وحدات الادخال والاخراج Input/Output Units
وحدة الادخال وهي PIN او رجول الميكرو التي تتصل بمفاتيح التحكم أو الكاي باد key pad وهي لوحة مفاتيح تشبة الموجوده في الالة الحاسبة أو الحساسات "سنسور" وهو لقياس درجة الحرارة ويوجد انواع كثيرة من الحساسات فهناك من تقيس شدة الاضاءة مثل المقاومات الضوئية وهناك ما تقيس ضغط الهواء او نسبة ثاني اكسيد الكربون في الهواء وهناك ما تستقبل الموجات

وحدات الاخراج وهي رجول pin تتصل بلمبات led ليد أو شاشة lcd او ريلاي او ترانزستور او محركات تيار مستمر

ملحوظة
بنات الادخال والاخراج في الميكروكنترولر مشتركة بمعني انه يمكن أن تستخدم رجل للإدخال او الاخراج وسوف تعرف ذلك في تصميم الليد ماتركس

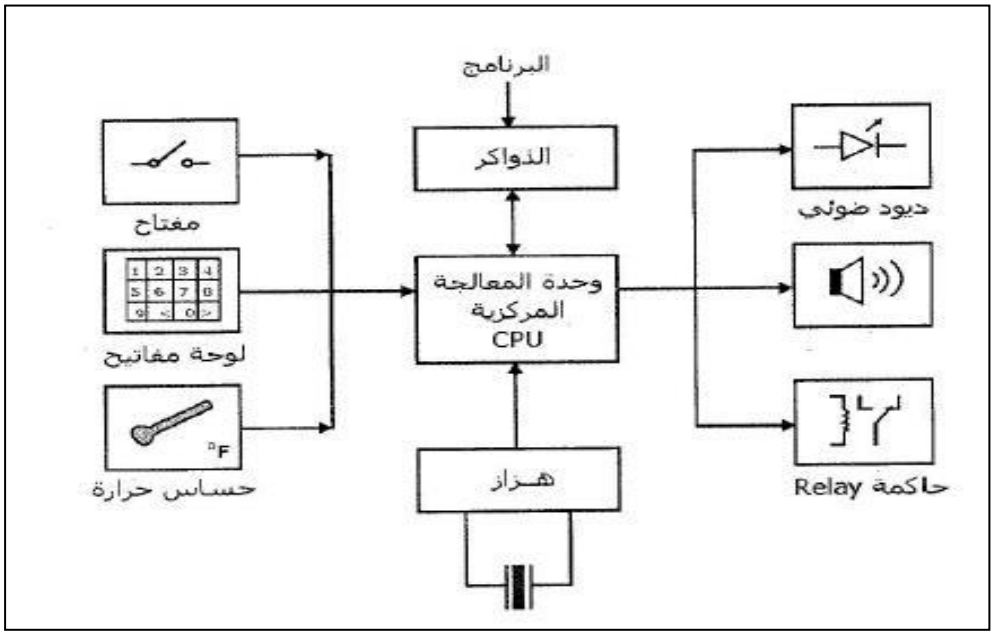
6- التايمر Timer يستخدم في تنفيذ التعليمات في زمن محدد

7- وحدة الباور سبلاي وهي وحدة التغذية الكهربية ويتم تغذية الميكرو بجهد مستمر 5 فولت وتيار 25 ملي امبير وسوف نشرح في الفصل القادم الدائرة الكهربية للميكرو

8- محول ADC وهو يحول الاشارات التماثلية الانالوج إلى اشارات رقمية ديجيتال والاشارة الرقمية تأخذ قيمتان هي اما عالية 5 فولت او منخفضة صفر فولت
وترجع اهمية محول الاشارات ADC الى ان الميكرو لا يستطيع التعامل مباشرة مع الاشارات التماثلية مثل اشارات الصوت او موجات الاستقبال

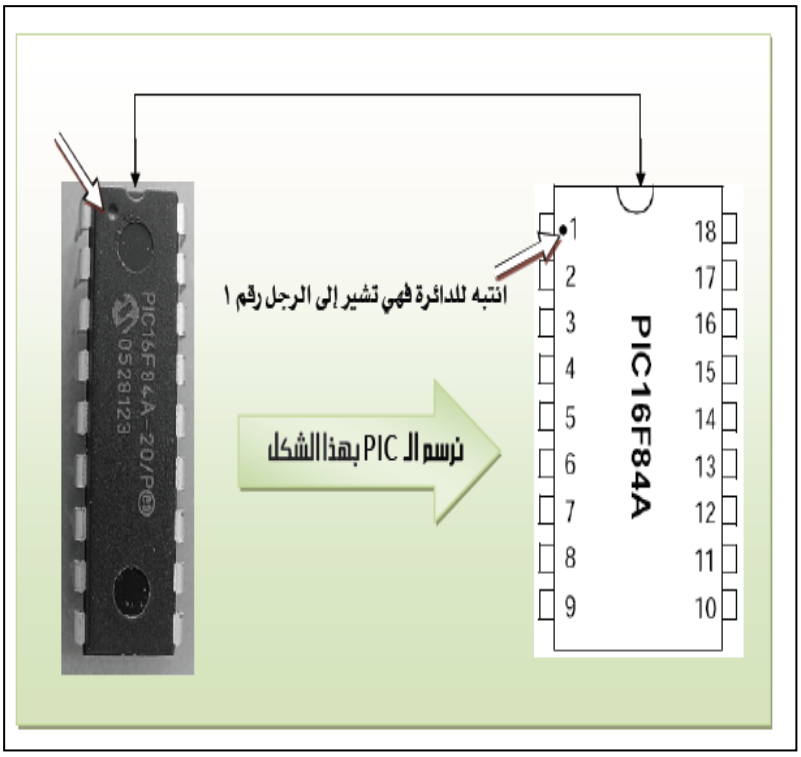
9 – مذبذب او ساعة نظام وهي تحدد قيمة اقصى زمن يمكن تنفيذ اوامر الميكروكنترولر
ويوجد بعض انواع الميكرو كنترولر بها مذبذب داخلي مثل عائلات PIC18F

وهذا شكل تركيب الميكروكنترولر



# الفصل الثاني
# قرائة الداتا شيت
# للميكروكنترولر

**أولا تحديد رجول الميكروكنترولر**
لاحظ مع الصور الاتية



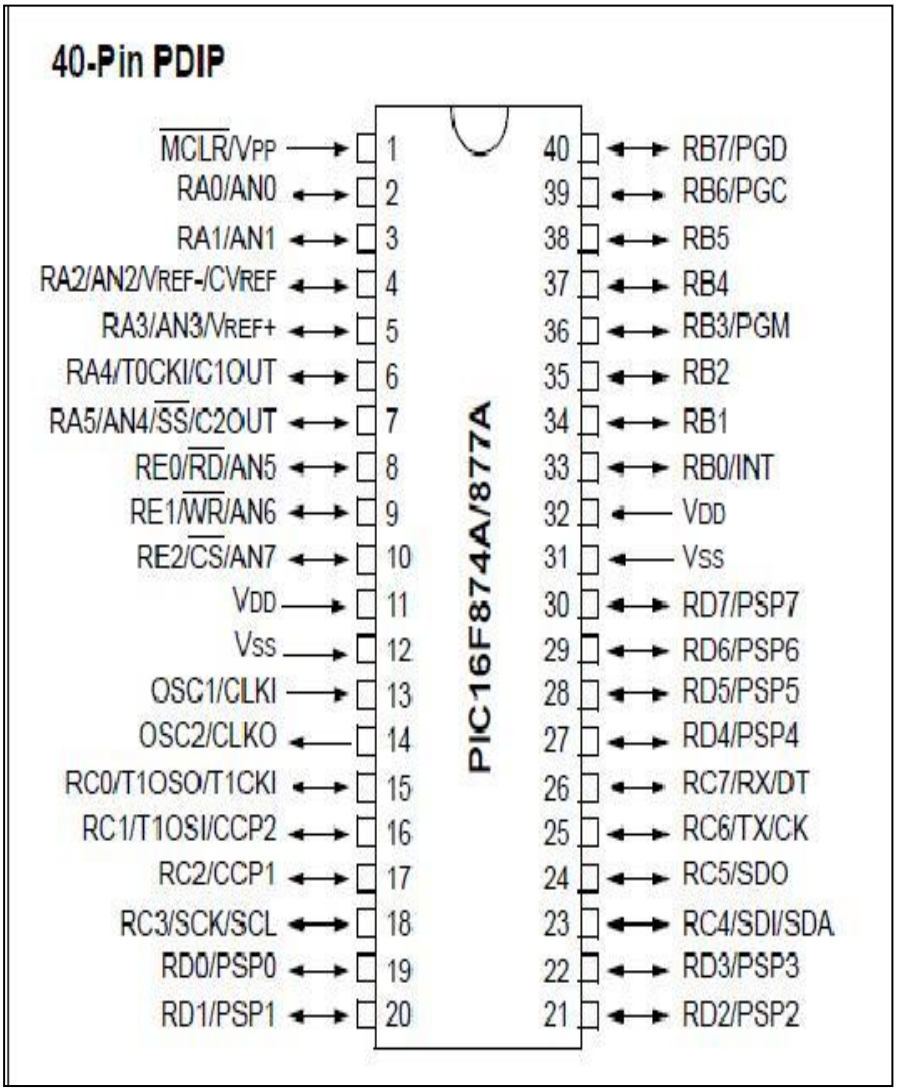

يتم تحديد الرجل او البن رقم "1" في الميكرو كالتالي ستجد نصف دائرة وتوجد على يسارها نقطة صغيرة بجوار هذه النقطة البن رقم واحد
ويتم التسلسل إلى نهاية الرجول بحيث الرجل رقم "1" يقابلها اخر رجل واو بن في الميكروكنترولر

كما يوجد ميكروكنترولر يحتوي على 18 بن ويوجد اخر يحتوي على 40 بن واخر 44 بن

ويجب أن تعلم أننا في هذه الكتاب سوف نستخدم ميكروكنترولر من النوع PIC المنتشر في مصر ويوجد في محلات رام والنخيلي في القاهرة وسوف نكون اكثر تحديدا حيث نستخدم PIC16F877A

**وظائف البنات في الميكروكنترولر**

سوف أقوم بشرح معظم وظائف البنات في الميكرو فأنتبه جيدا عزيزي القارئ فهذا الجزء من الاهمية حيث أن باقي اجزاء الكتاب مبنية على هذا الجزء



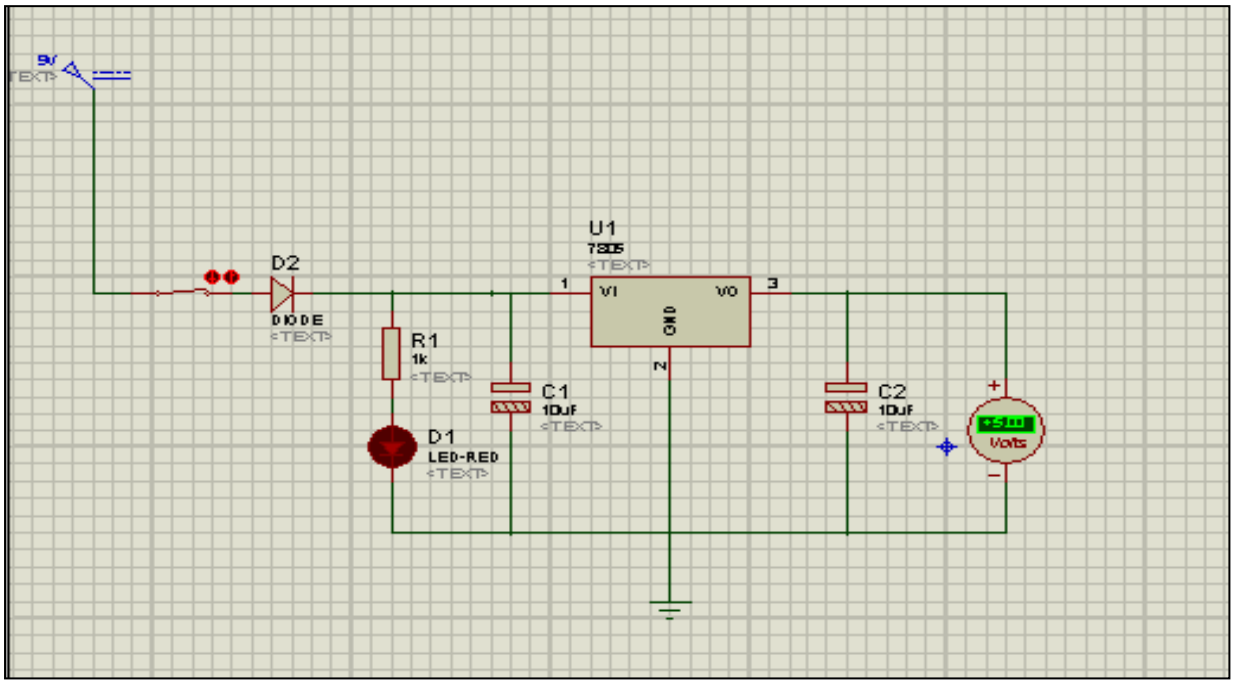

في الشكل السابق للميكرو PIC16F877A يوجد رجول لا يمكن الاستغناء عنها فمثلا

الرجول 11و 12 لتغذية الميكرو بالجهد الضروري له

ستجد بجوار الرجل 11 VDD وهي جهد موجب 5 فولت

اما الرجل 12 VSS فهي توصل بالارضي

ولكي تحصل على جهد 5 فولت و25 ملي امبير يجب أن تستخدم
الدائرة الاتية

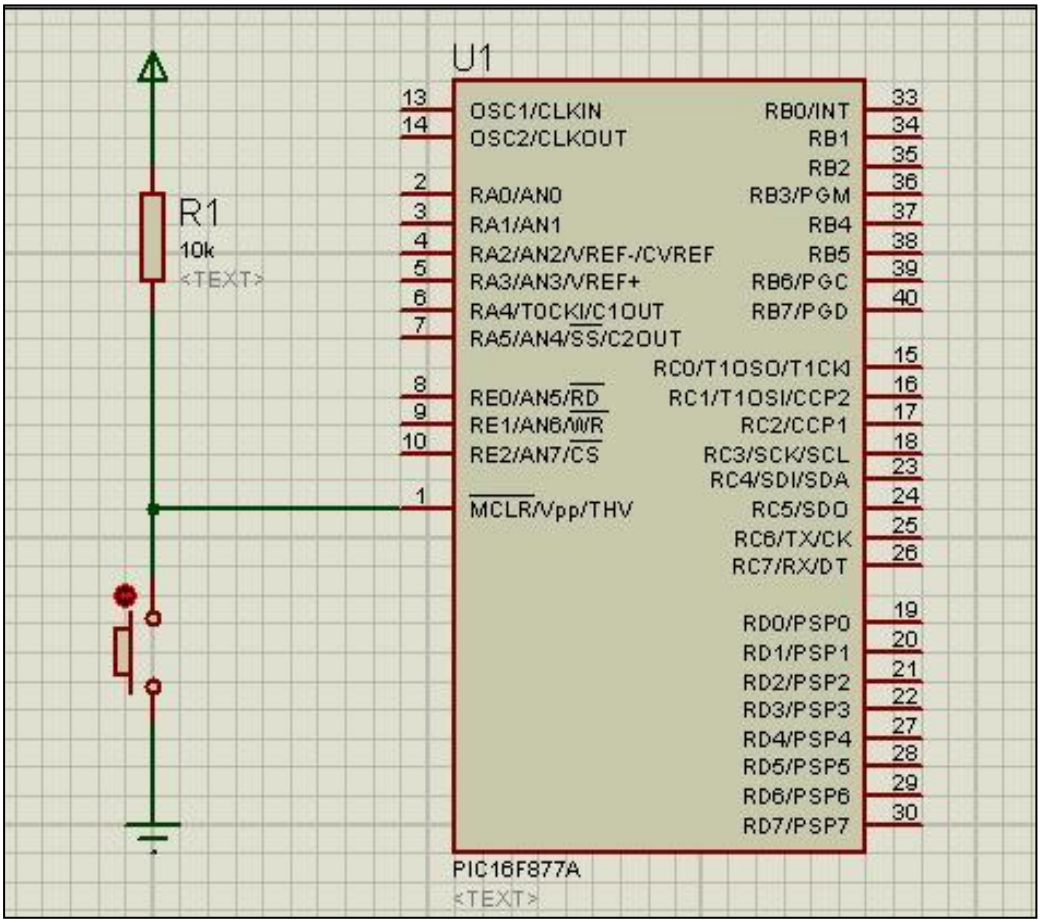



اذا كنت ستستخدم بطارية 9 فولت 300 ملي امبير
وتحتاج منظم جهد 7805 (اي سي ريجيرتور )
وتحتاج ايضا 2 مكثف كيميائي 10 ميكروفاراد
ودايود
مفتاح لغلق وفتح الدائرة
لمبة اشارة( ليد) لتعرف هل هناك جهد كهربي ام لا
ومقاومة 1 كيلو اوم تتصل مع LED

وهذا هو شكل منظم الجهد 7805

حيث أن
طرف للدخل 9 فولت 300 ملى امبير
وطرف ارضي GND
وطرف خرج يخرج 5 فولت

**الرجل رقم (1)**
ستجد مكتوب بجواراها MCLR وتعني master clear أي اعادة التشغيل اي اعادة تشغيل الميكروكنترولر مثل زر ريسترت في الكمبيوتر ولكن يسمى هنا ريست
وسوف تلاحظ وجود شرطة فوقه ـــــــــ وهي تعني ان اعادة التشغيل تعمل عند الجهد المنخفض طالما أن الجهد عالي لن يقوم الميكرو بعمل اعادة تشغيل وتوصيل الدائرة كما في الشكل التالي

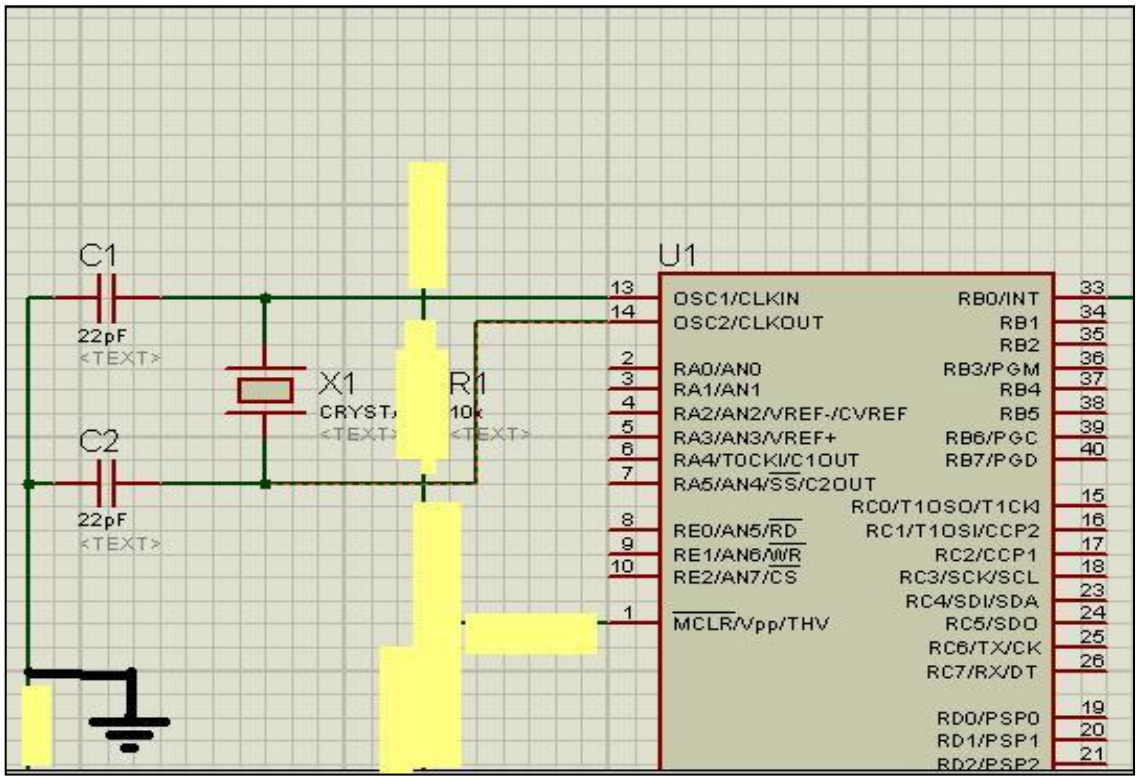


**الرجول 13 و 14**

وهي اطراف مولد النبضات "المذبذب " Oscillator وتحتاج كريستالة 8 ميجا هيرتر ومكتوب بجوارهما OSC1 و OSC2

وستجد في الداتا شيت المكثفات الضرورية لعمل الكريستالة

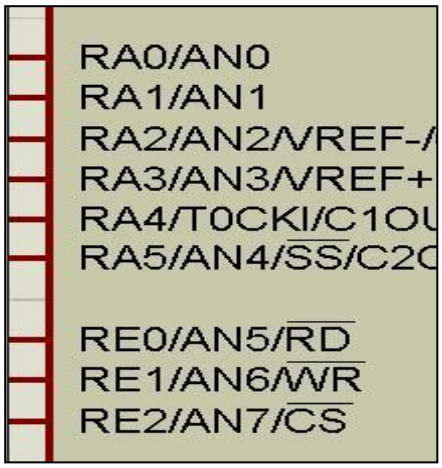
وتوصيل الدائرة كما في الشكل التالي




سوف تحتاج مكثفين 22 بيكوفاراد متصلين بالتوازي مع الكرستالة

**الرجول AN**

وهي الرجول الخاصة بإدخال الاشارات الانالوج او التماثلية وهي 8 رجول تبدأ من AN0 حتى AN7




ملحوظة

الرجول الانالوج يمكن أن تستعمل كا ديجيتال وسوف يتضح ذلك

**باقي الرجول**

وهي مجموعة منافذ Ports تستخدم كمدخلات أو مخرجات

Port A أو RA 6 رجول من RA0 : RA5

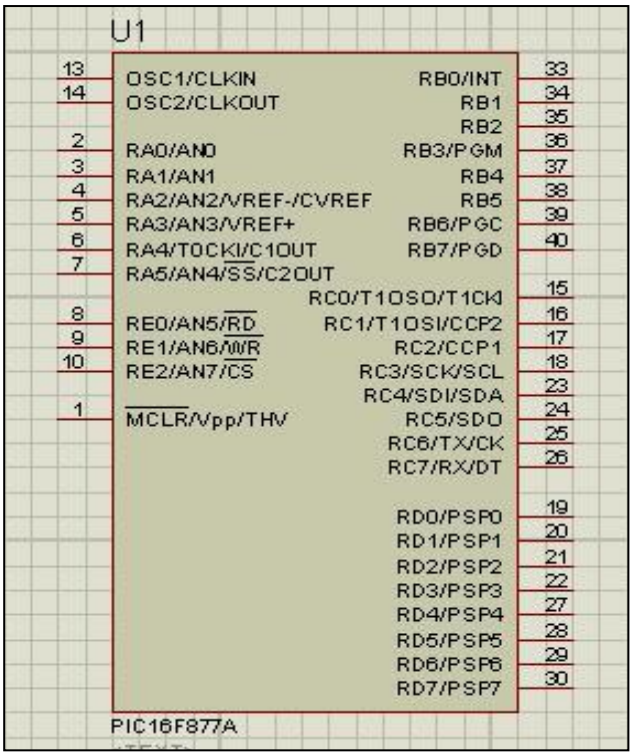
Port B أو RB 8 رجول من RB0: RB7

Port C أو RC 8 رجول من RC0:RC7

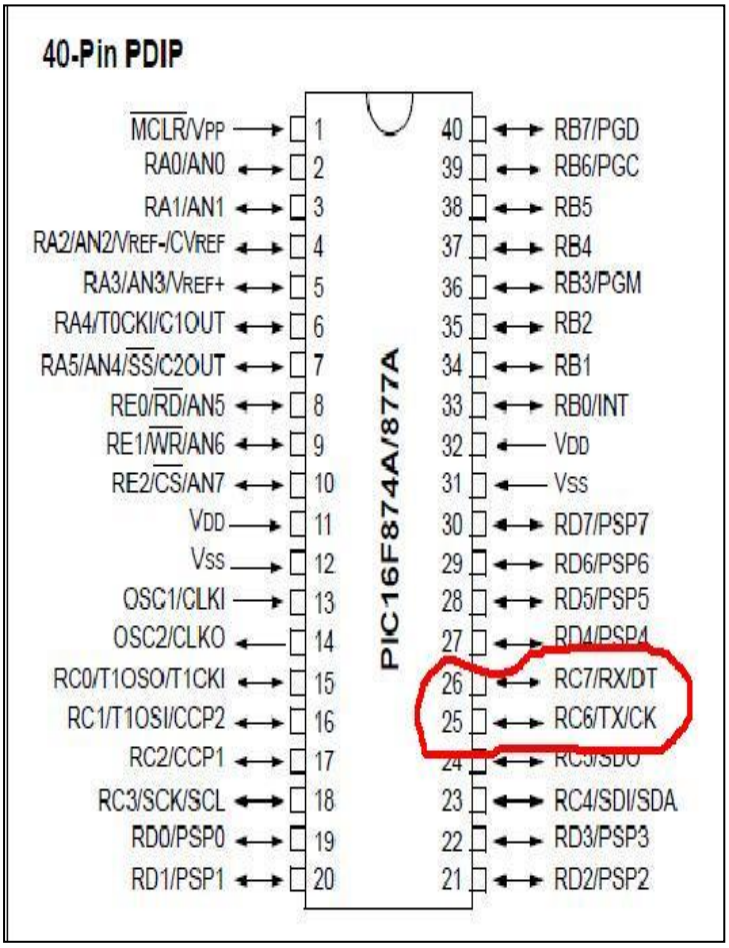
Port D أو RD 8 رجول من RD0 :RD7

Port E أو RE 3 رجول من RE0 : RE2

والشكل يبين باقي رجول الميكروكنترولر

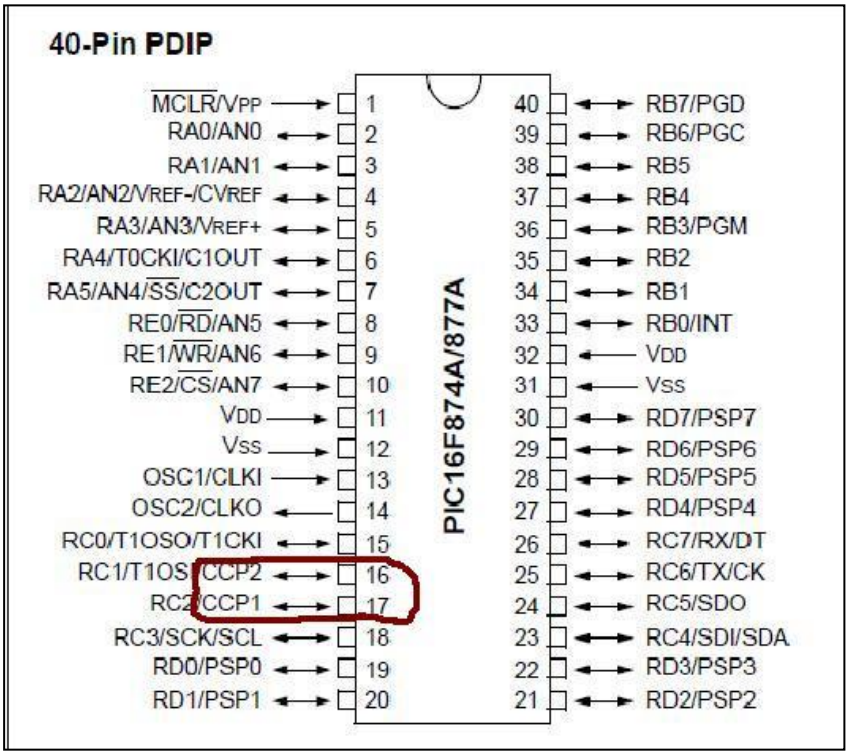



## وظائف أخرى للرجول



معظم رجول او بنات الميكروكنترولر تحتوي على اكثر من وظيفة
و
ستجد مكتوب بجوار الرجل 26 و 25 RX و TX وهذه الرجول وظيفتها ربط الميكروكنترولر بالكمبيوتر أو ميكروكنترولر أخر
RX تستقبل البيانات
TX ترسل البيانات

الرجول 17 و 16 ستجد مكتوب بجوارها CCP1 و CCP2 وهي مسئولة عن انتاج الترددات مثل انتاج موجة بتردد 37000 هيرتز وسوف تستخدم عند مشروع ريموت كنترول بسيط




وهكذا نكون قد انتهاينا من معظم وظائف الرجول Pins وسوف نتحدث عن خصائص الميكروكنترولر الكهربية

وكما ذكرنا انفا أن الميكرو يحتاج
جهد 5 فولت
تيار 25 ملي امبير
اقصى تردد يمكن أن ينتجة 20 ميجا هرتز

وبذلك نكون انتهاينا من الداتا شيت الخاصة بالميكرو
PIC16F877A

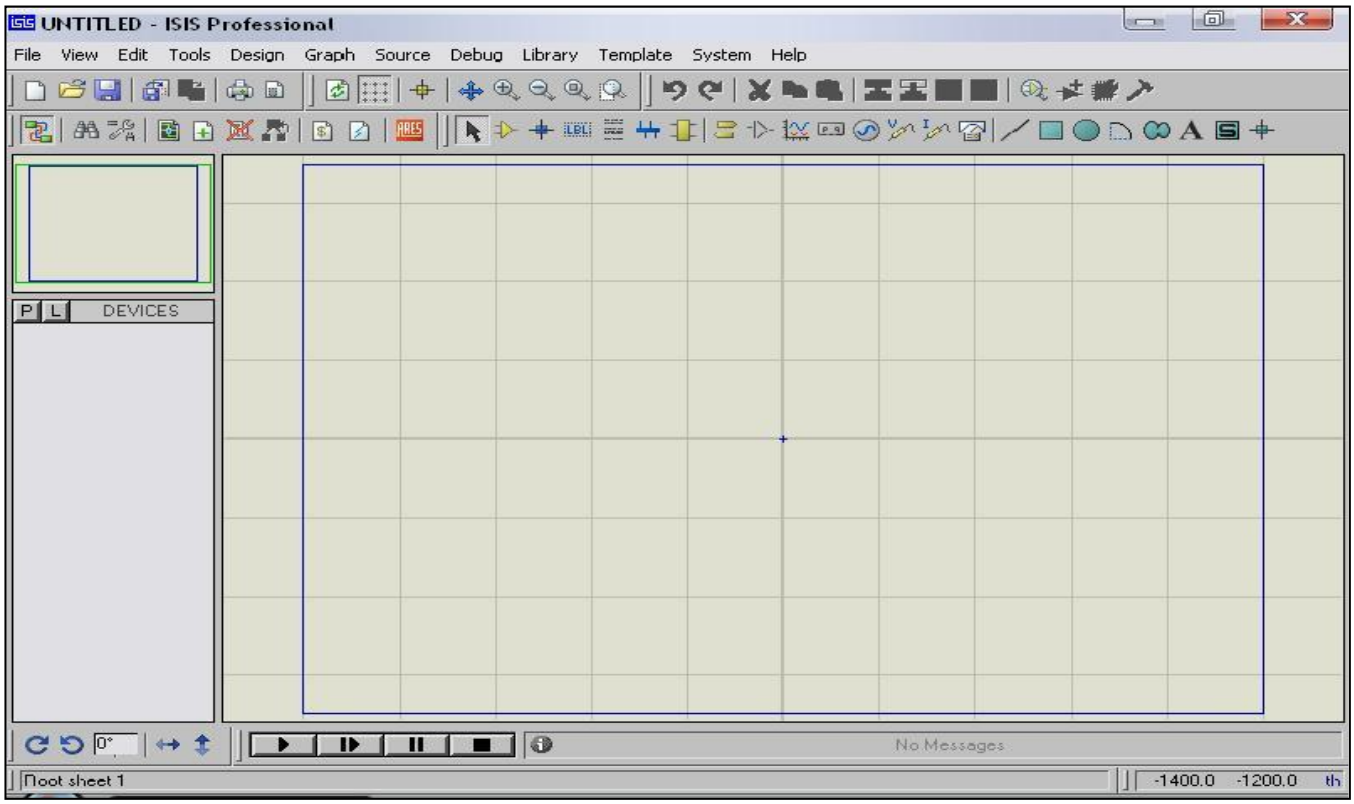
# الفصل الثالث
# البرامج المستخدمة مع الميكروكنترولر

# برنامج *Proteus*

يتميز برنامج بروتس عن غيره من برامج الرسم الالكتروني أنه يحتوي على عدد كبير من العناصر الالكترونية والميكروكنترولر كما أن فيه خاصية المحاكاة حيث بعد تصميم الدائرة يمكن تجربتها على البرنامج
كما توجد في برنامج بروتس خاصية غاية في الاهمية وهي أنه يتقبل ملفات هيكس على الميكروكنترولر ومن هنا نجد أنه قريب جدا من الواقع العملي وأي خطأ في الدائرة سوف ينبهك أثناء تشغيل الدائرة

بعد تحميل البرنامج سوف نقوم بفتحه كالتالي
من قائمة start ثم All Programs ثم بروتس 7 برفشنال اختر الايقونة الزقاء ومكتوب بجوارها ISIS 7 كما في الشكل التالي

بعد فتح برنامج بروتس سوف تظهر واجهة المستخدم كالتالي

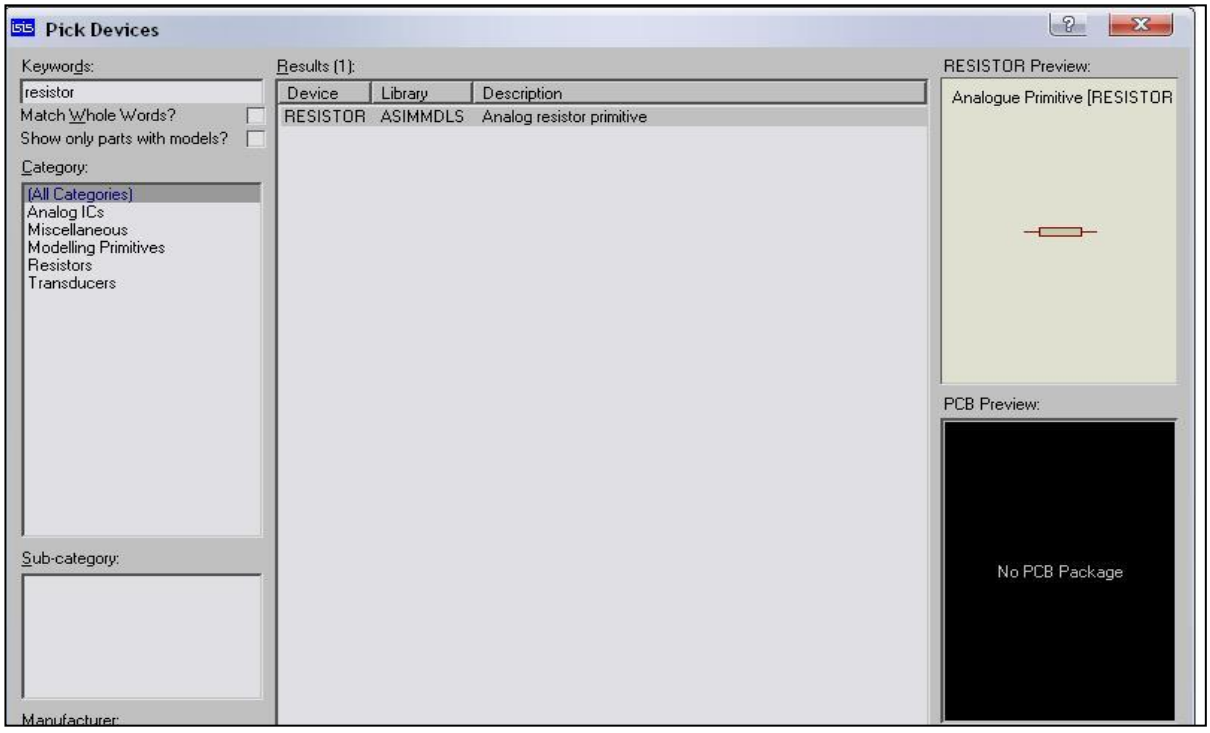


يمكن استخدام Save لحفظ المشروع

ولكي تستطيع الوصول إلى العناصر الالكترونية أضغط على هذه الاداة

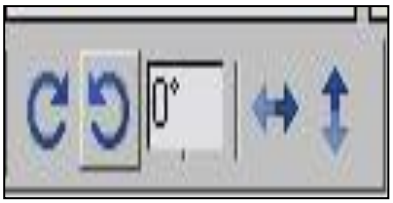
وبعد تحديدها ستجد يسار الشاشة قائمة DEVICE ستجد حرف P كما في الشكل التالي

اضغط علية لتظهر قائمة العناصر الالكترونية كما في الشكل التالي



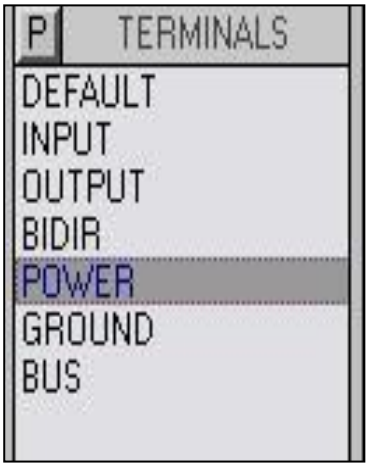

قم بكتابة العناصر التي تريد أضافتها مثل مقاومة "Resistor"
سوف تظهر اضغط عليها دبل كيلك لتضاف إلى قائمة العناصر ثم
اكتب LED RED اي لمبة ليد حمراء
ثم قم بإغلاق القائمة
لتظهر الادوات في قائمة DEVICE

كما يمكن عكس اتجاه العناصر من خلال اسهم تبديل الاتجاه كما في الشكل التالي




واذا اردت اضافة DC Power اختر الاداة الاتية

لتظهر قائمة بها دي سي بور والارضي كما في الشكل التالي

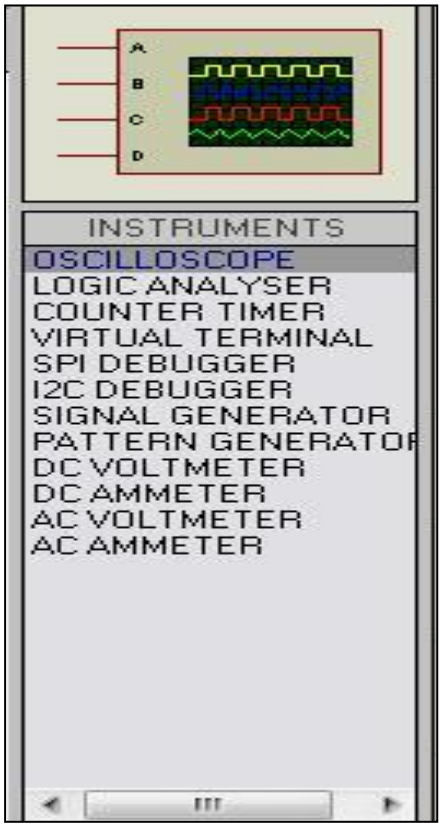



كما يمكن أضافة عناصر البور الذي تريدة مستمر أو متردد من هذه القائمة

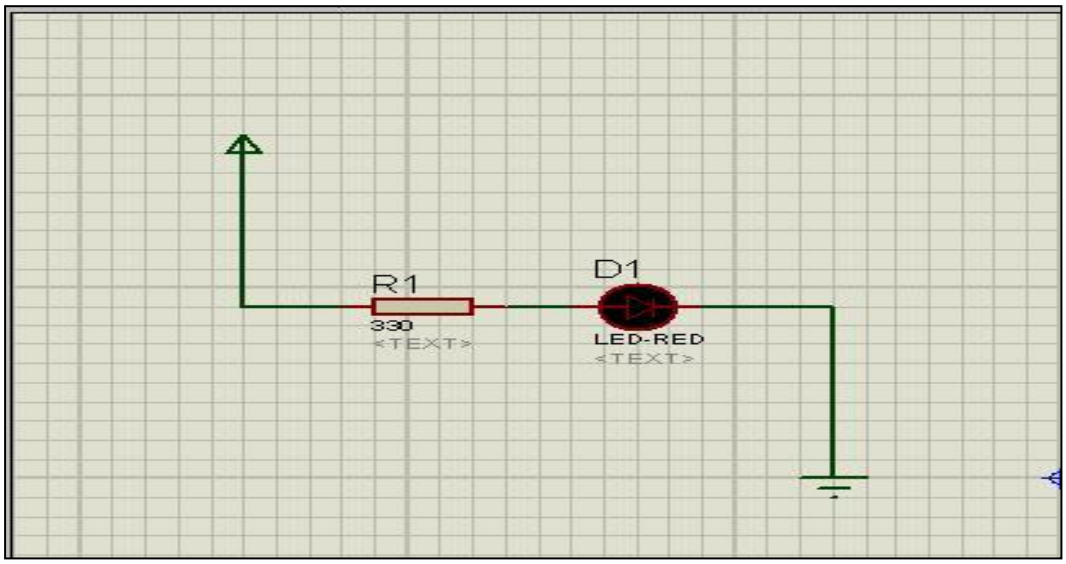
لتظهر القائمة التالية

كما يمكن اضافة اجهزة القياس بالضغط على الشكل التالي

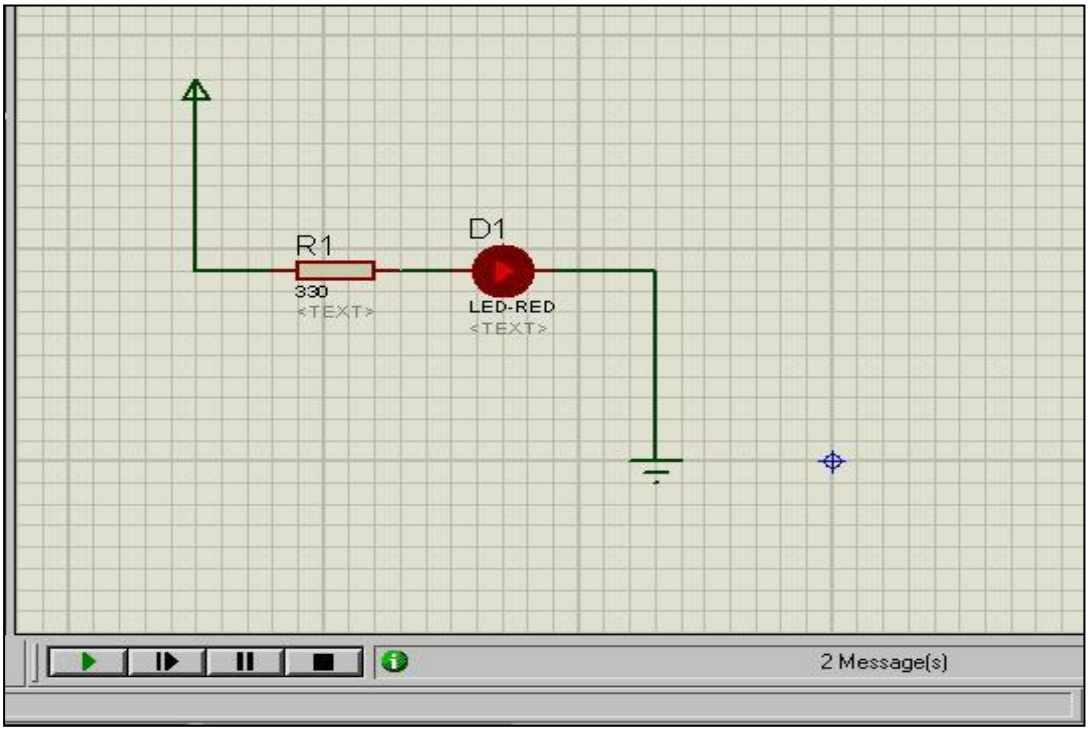
لتظهر القائمة كالتالي



لنقوم الان برسم دائرة بسيطة قم بتحديد قائمة العناصر الانف ذكرها وقم بالضغط عليها مرة واحدة داخل صندوق الرسم واضف باقي العناصر
كما في الشكل التالي




بعد ذلك اضغط على قائمة تشغيل RUN اسفل مربع الرسم مع ملاحظة تعديل قيمة المقاومة إلى 330 بدل 10 كيلو بالضغط عليها نقرة مزدوجة



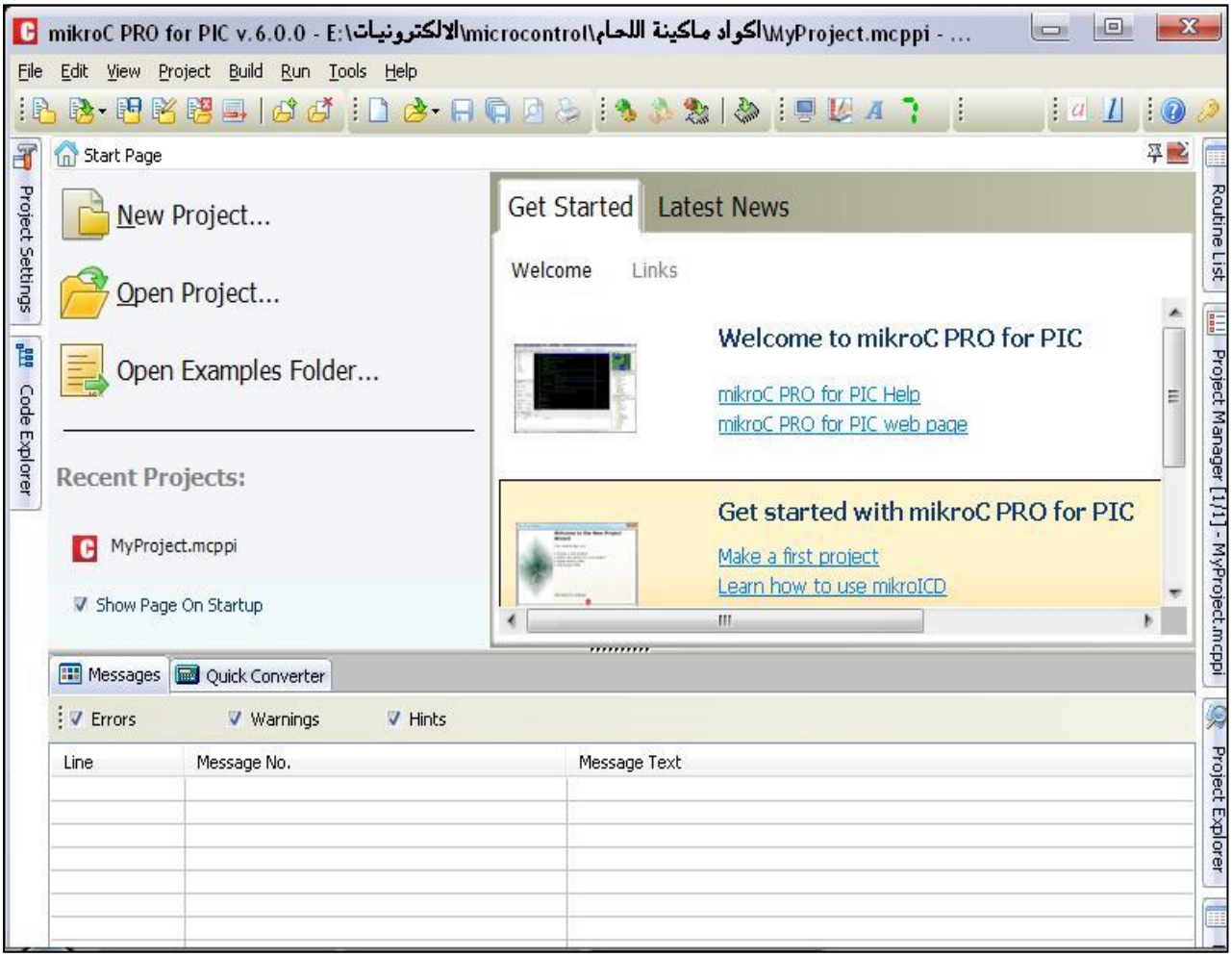
وهكذا نكون انتهينا من شرح برنامج بروتس وسوف نقوم بشرح برنامج كتابة الشفرة للميكروكنترولر وهو ميكروسي بك

## برنامج ميكروسى بك

## mikroC PRO for PIC

**كما يمكنك تحميل البرنامج مجاني من موقع الشركة وهي تعطي مساحة اقل لكنها تكفي بالغرض**

بعد تحميل البرنامج قم بفتحه لتكون واجهته كالتالي

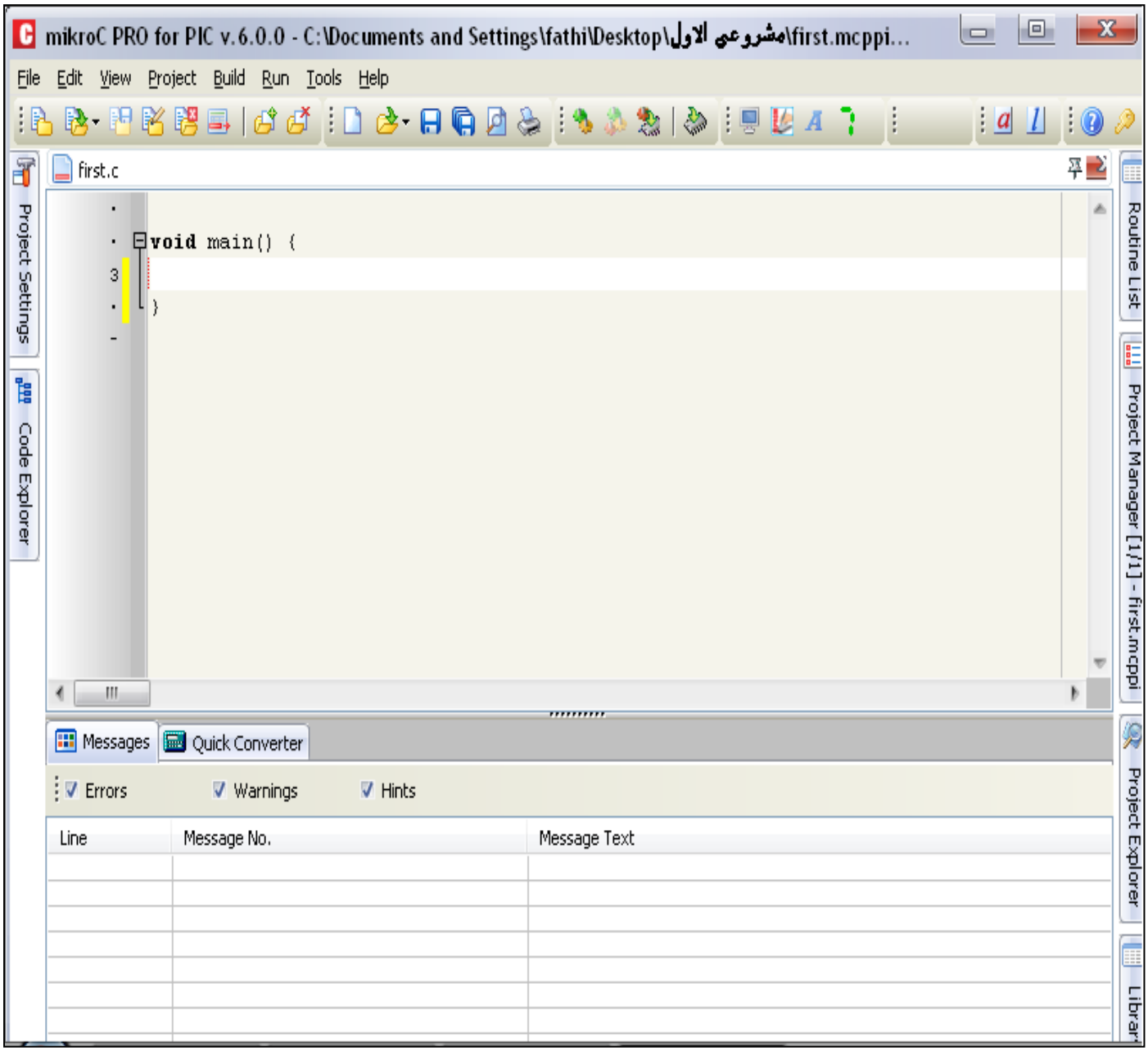



أضغط على New Project لانشاء مشروع جديد

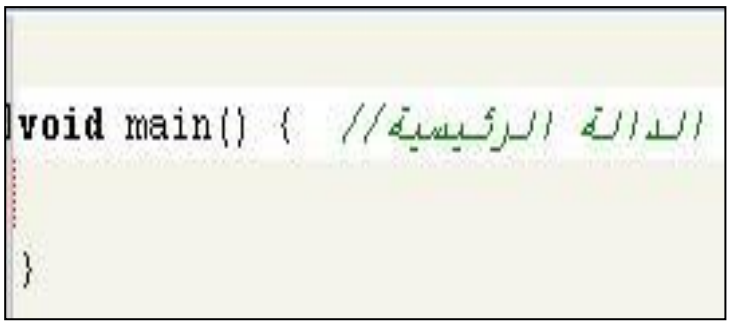
لتظهر لك النافذة التالية

اضغط على Next لتظهر النافذة التالية

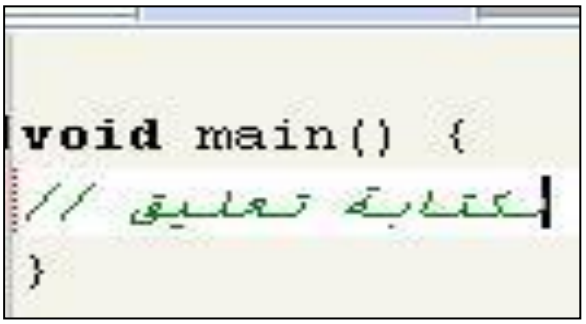
ستجد مكتوب فيها Project Name اي اسم المشروع
كما ستجد Project Folder أى مكان حفظ المشروع
كما ستجد Device Name أى اسم الميكروكنترولر
حدد PIC16F877A
كما ستجد Device Clock وهي مقدار الكرستالة أو المذبذب
وسوف تكون 8 ميجاهيرتز
بعد ذلك أضغط Next حتى تظهر نافذة الكود كالتالي



ستجد مكان كتابة الكود داخل الدالة الرئيسية void main()




كما يمكن أن تكتب تعليق بإضافة "//" ثم أضافة التعليق ويكمن أهمية التعليق إذا كان البرنامج متشعب فتكتب وظيفة كل كود داخل البرنامج




***** القوائم***

يحتوي برنامج ميكرو سي على قوائم سوف نذكر أهم هذه القوائم

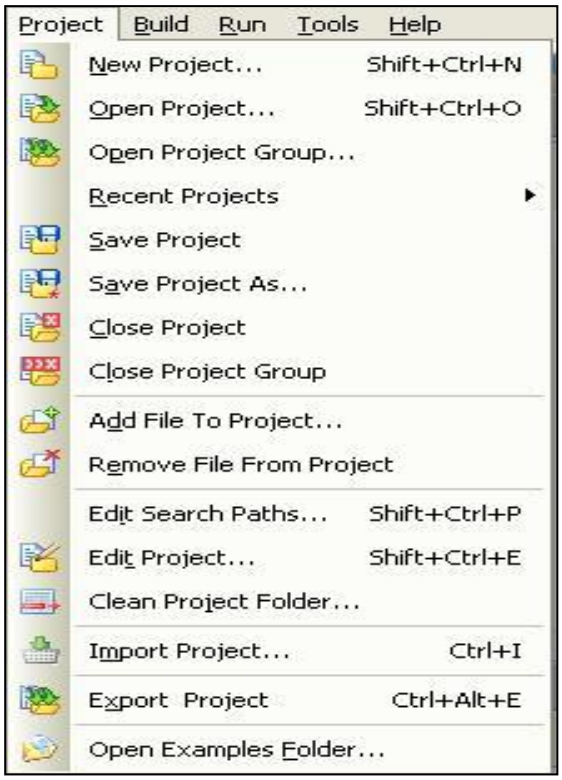
قائمة File

تحتوي على
New أي ملف جديد
Open فتح ملف موجود
Save حفظ التعديلات الحالية
Save As حفظ المشروع بإسم جديد
Exit اغلاق المشروع

قائمة Edit

تحتوي على
Undo تراجع في الكود
Redo عكس التراجع
Cut قص كود
Copy نسخ كود
Paste لصق الكود
Delete حذف كود بعد تحديده
Find البحث عن نص داخل الاكواد
GOTO Line الذهاب إلى سطر معين

# قائمة Project




وتحتوي على

New Project

لانشاء مشروع جديد

Open Project

لفتح مشروع موجود

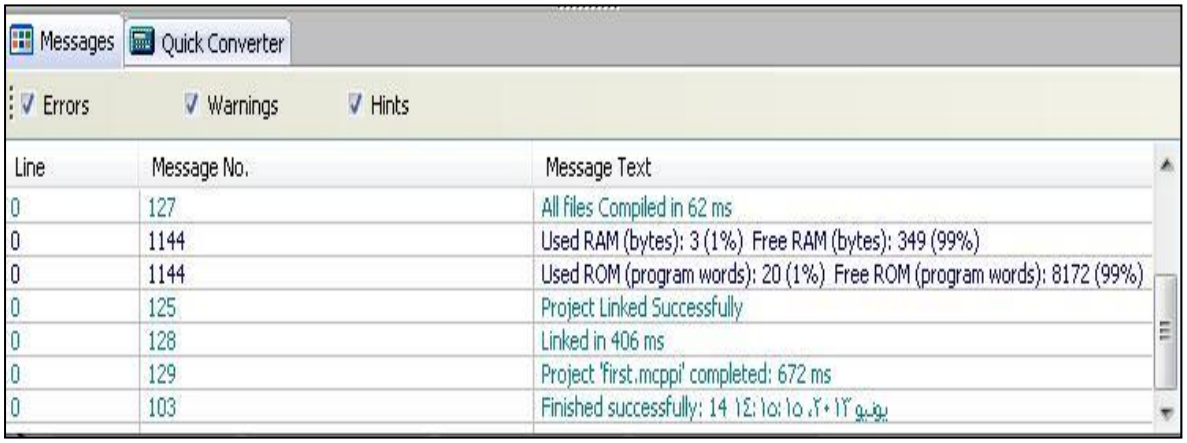
Edit Project

لتعديل خصائص مشروع مثل تغيير نوع الميكروكنترولر

## قائمة Build

وتعتبر قائمة في غاية الاهمية حيث أنها تنشأ ملف بامتداد HEX الذي يتم شحنه أو حرقه أو تحميله داخل الميكرو كنترولر

وعندما يكون الكود صحيح تظهر اسفل هذه النافذة




ليدل على أن الكود صحيح وغير ذلك تظهر أخطاء باللون الاحمر

## برنامج PICPgm Programmer

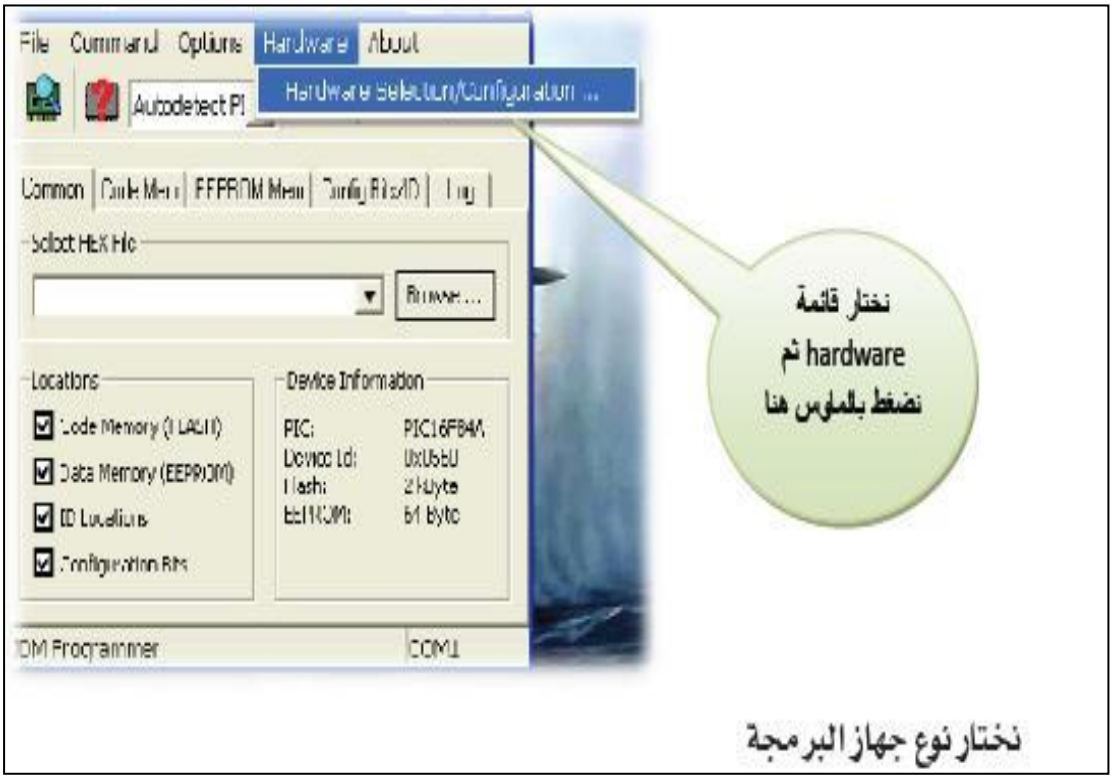
يقوم هذا البرنامج بشحن أو حرق ملف Hex داخل الميكروكنترولر

أولا يتم شراء مبرمجة كما في الشكل التالي

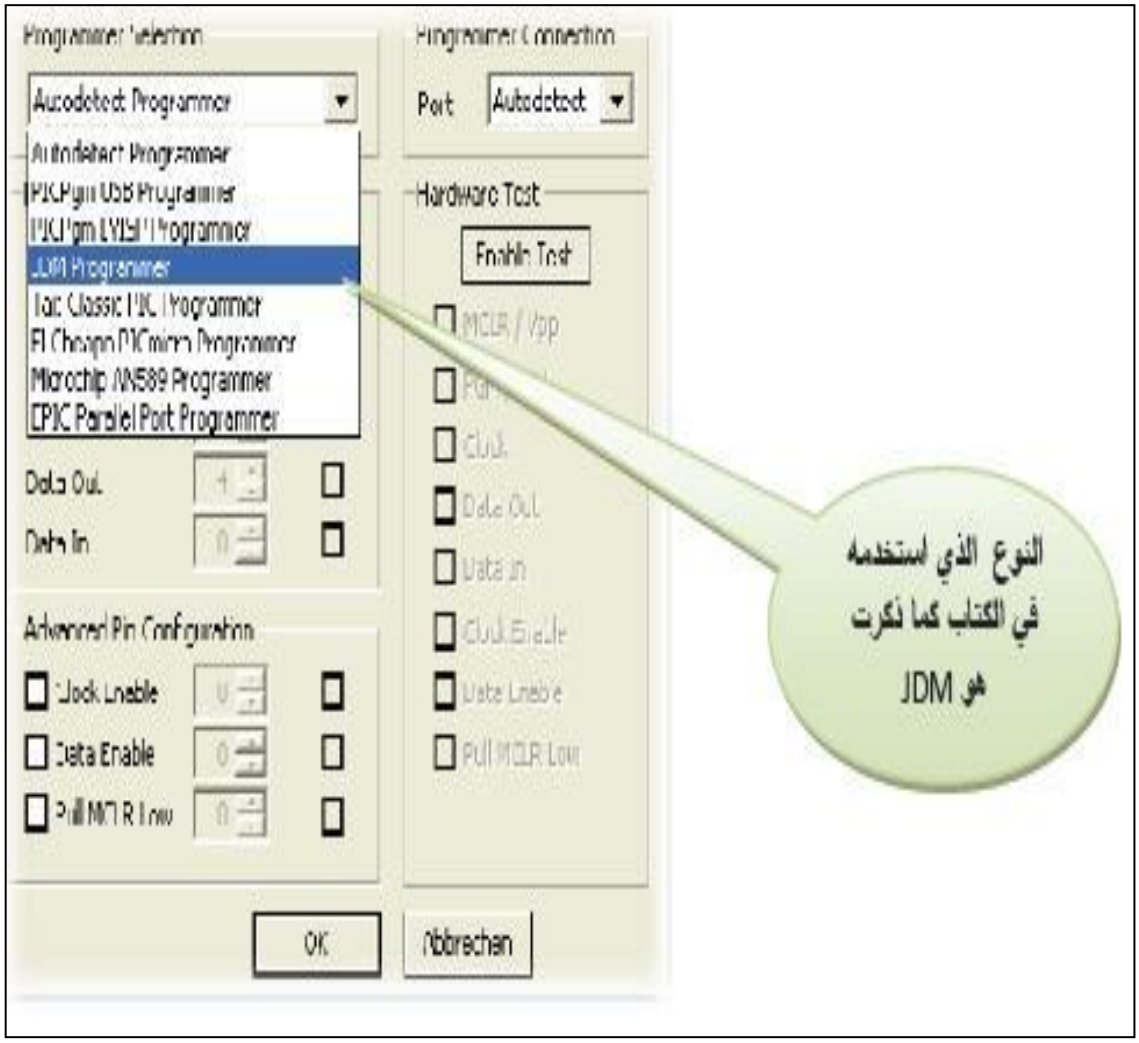
وتحتاج كابل سيريال بورت
وشكله كالتالي

ثم يتم توصيل كابل السيريال بالمبرمجة  والكمبيوتر كالتالي

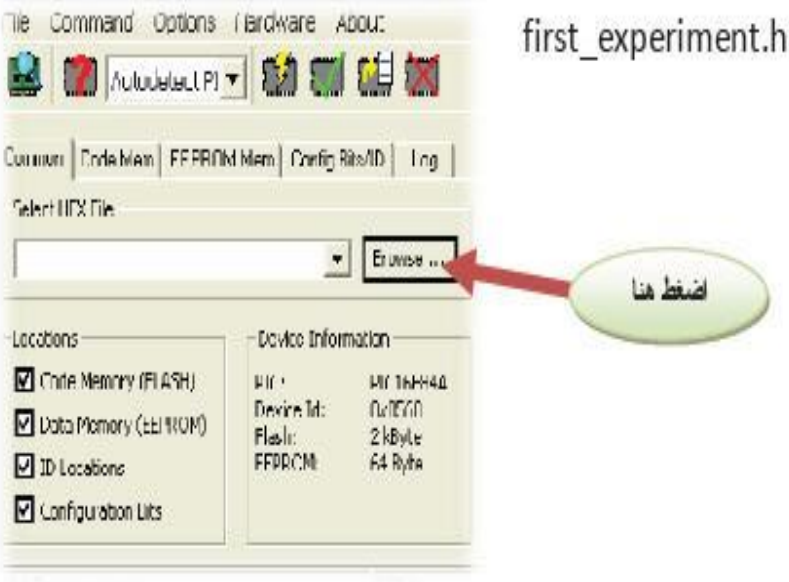
بعد ذلك شغل برنامج PICPgm Programmer



ثم نختار نوع المبرمجة وهي JDM




ثم اضغط على زر Browse وحدد المكان الذي يوجد فيه المشروع ومن ثم حدد ملف الهيكس حيث سيكون في تجربتنا هذه اسمه first_experiment.hex



نختار الملف ثم نضغط على زر Open
Look in:
book
first_experiment.hex
Date Modified: 31/01/2009 09:22 ص
Size: 120 bytes
اضغط هنا
بعد تحديد الملف
My Recent Documents
My Documents
My Computer
My Network Places
File name:
first_experiment.hex
Files of type:
HEX Files (*.hex)
Open
Cancel

ونختار نوع البك المستخدم وهو في هذه
File Command Options Hardware About
Autodetect
PIC16F747
PIC16F777
PIC16F818
PIC16F819
PIC16F84A
PIC16F87
PIC16F870
PIC16F871
PIC16F872
PIC16F873
PIC16F8734
PIC16F874
Common
Config Bits/ID
Log
Select HEX
Browse ...
Device Information
PIC:
PIC16F84A
Device Id:
0x0560
Flash:
2 kByte
EEPROM:
64 Byte

وبعد اتمام الشحن أفصل كابل السيريال ثم انزع الميكروكنترولر برفق

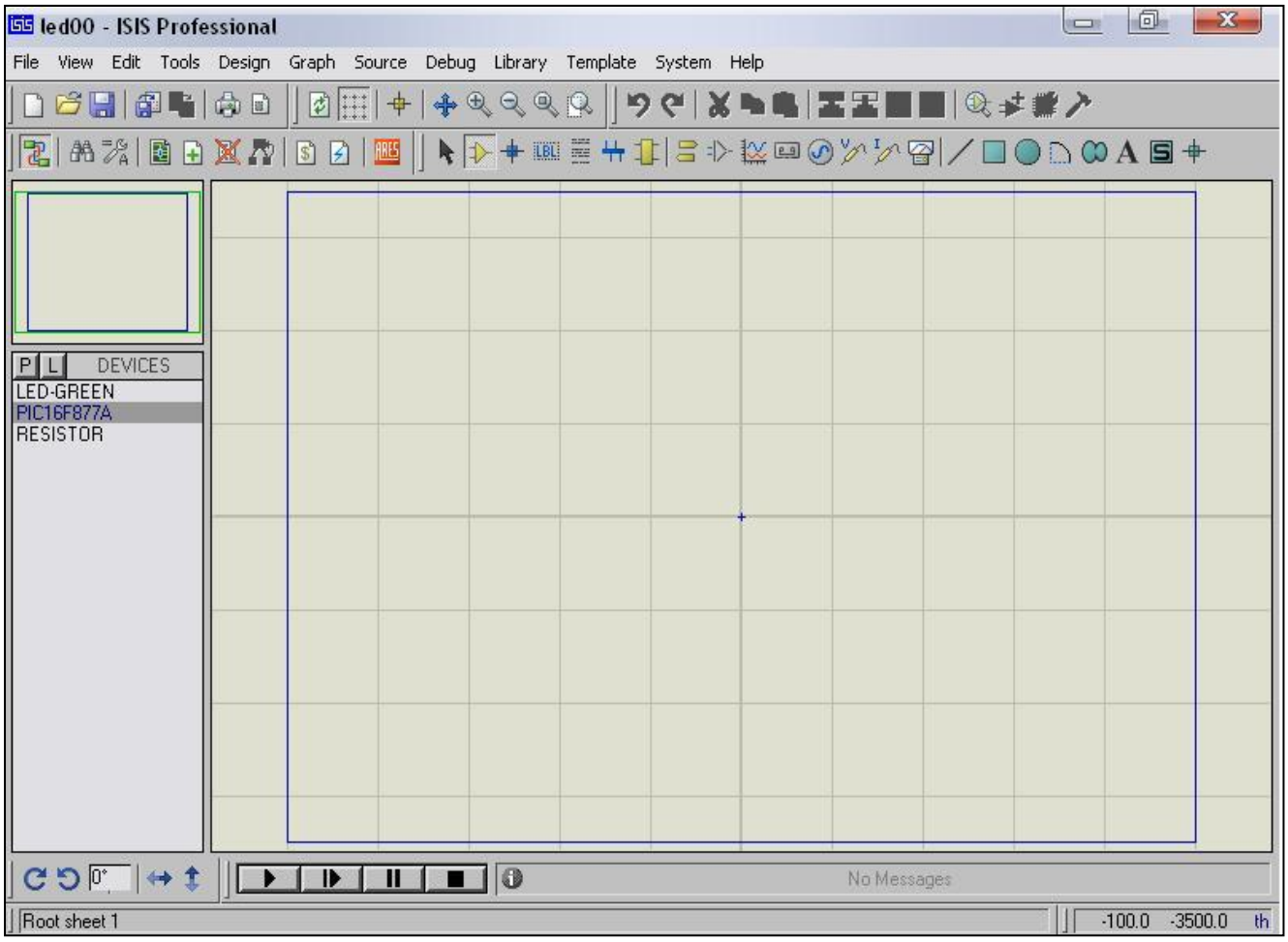
وبذلك نكون قد اتماننا هذا الفصل ويتبقى لنا التعامل مع الميكروسي ودواله ومكتباته المختلفة
هذا والله الموفق

# الفصل الرابع
# الليد ماتركس

## المشروع الاول

سوف نقوم في هذا المشروع بتصميم لمبة ليد تضيئ لعدد مرات محدد ثم تغلق

قوم بفتح برنامج بروتس كما تعلمت من قبل واضف العناصر الالكترونية التالية

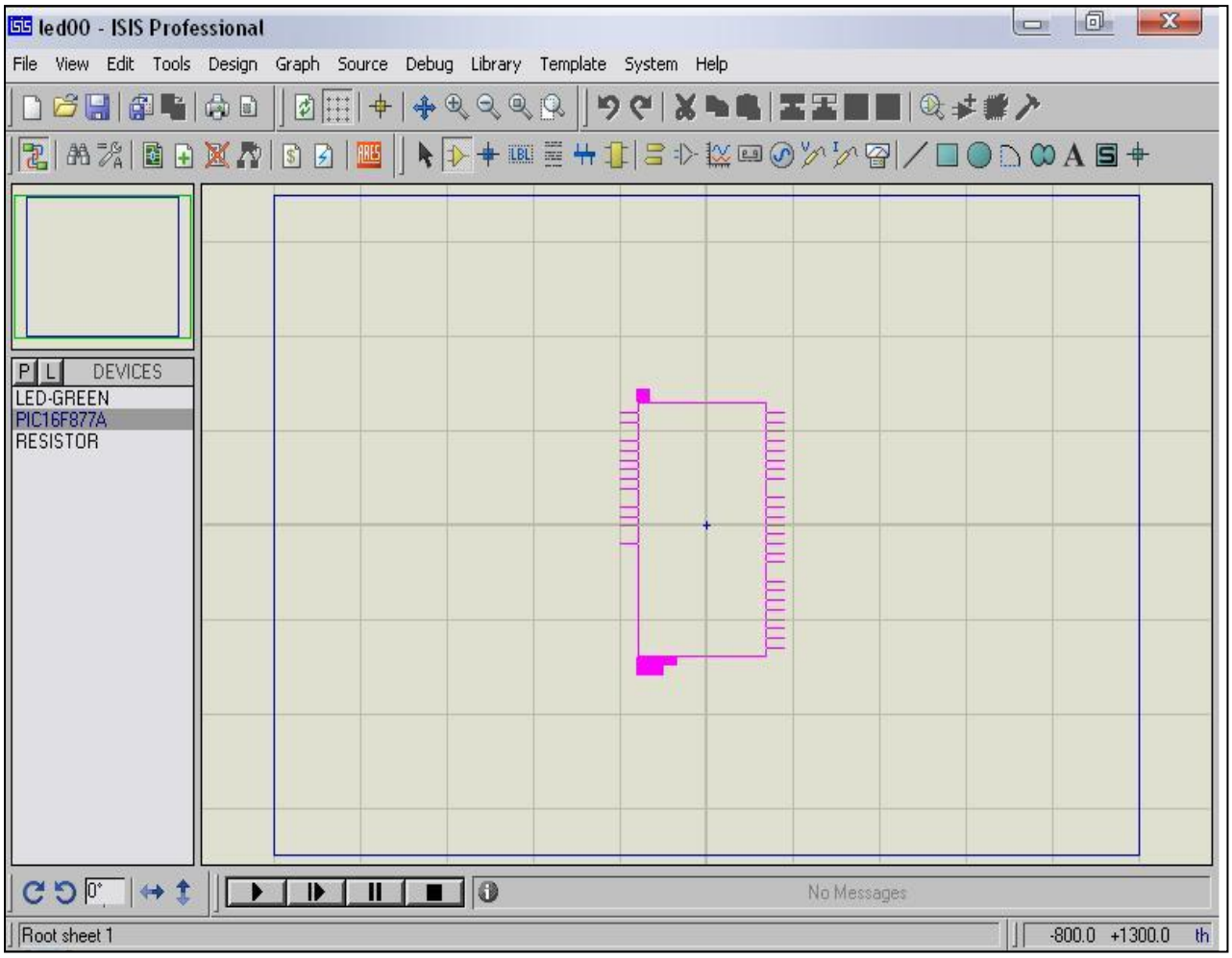



| | |
|---|---|
| led green | -1 |
| PIC16F877A | -2 |
| Resistor | -3 |

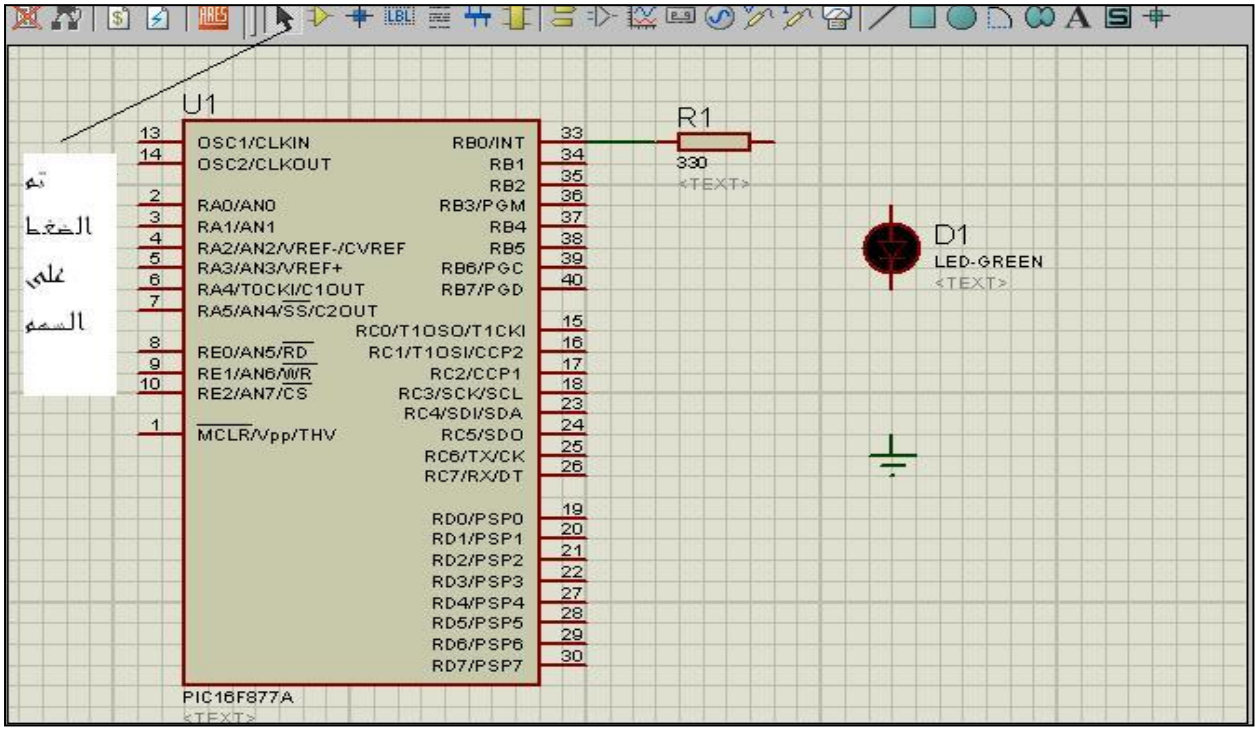
وتعني لمبة خضراء و الميكروكنترولر و مقاومة

قم بالضغط على الميكروكنترولر لتحديدة من صندوق العناصر الالكترونية

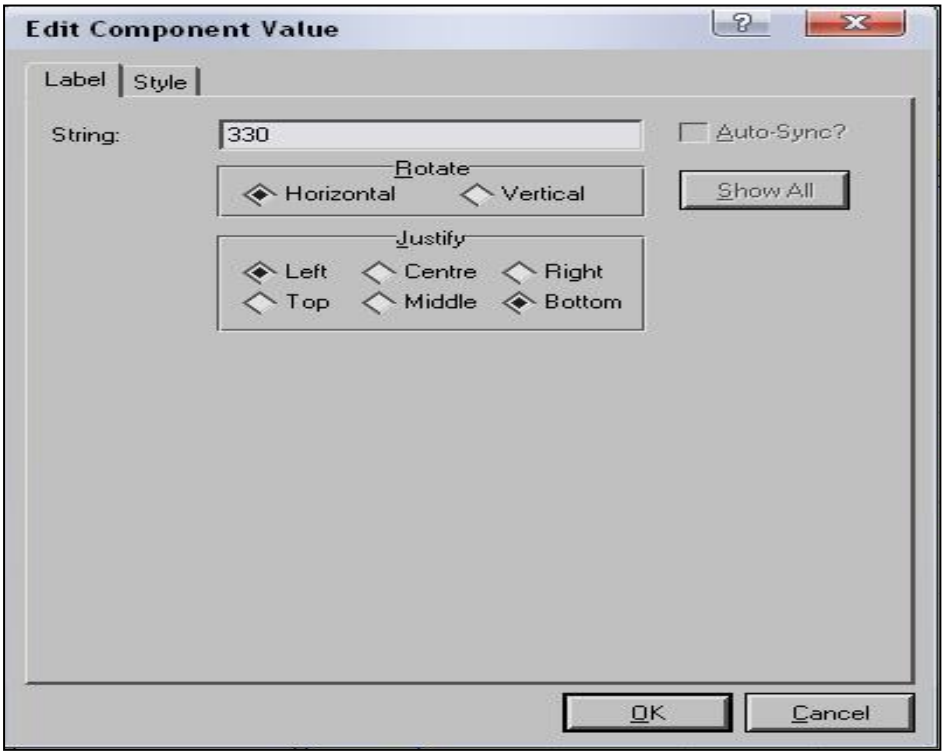
ثم بالضغط مره داخل صندوق الرسم سيظهر الميكركنترولر
ارفع اصبعك من على الماوس وضعه في منتصف صندوق الرسم



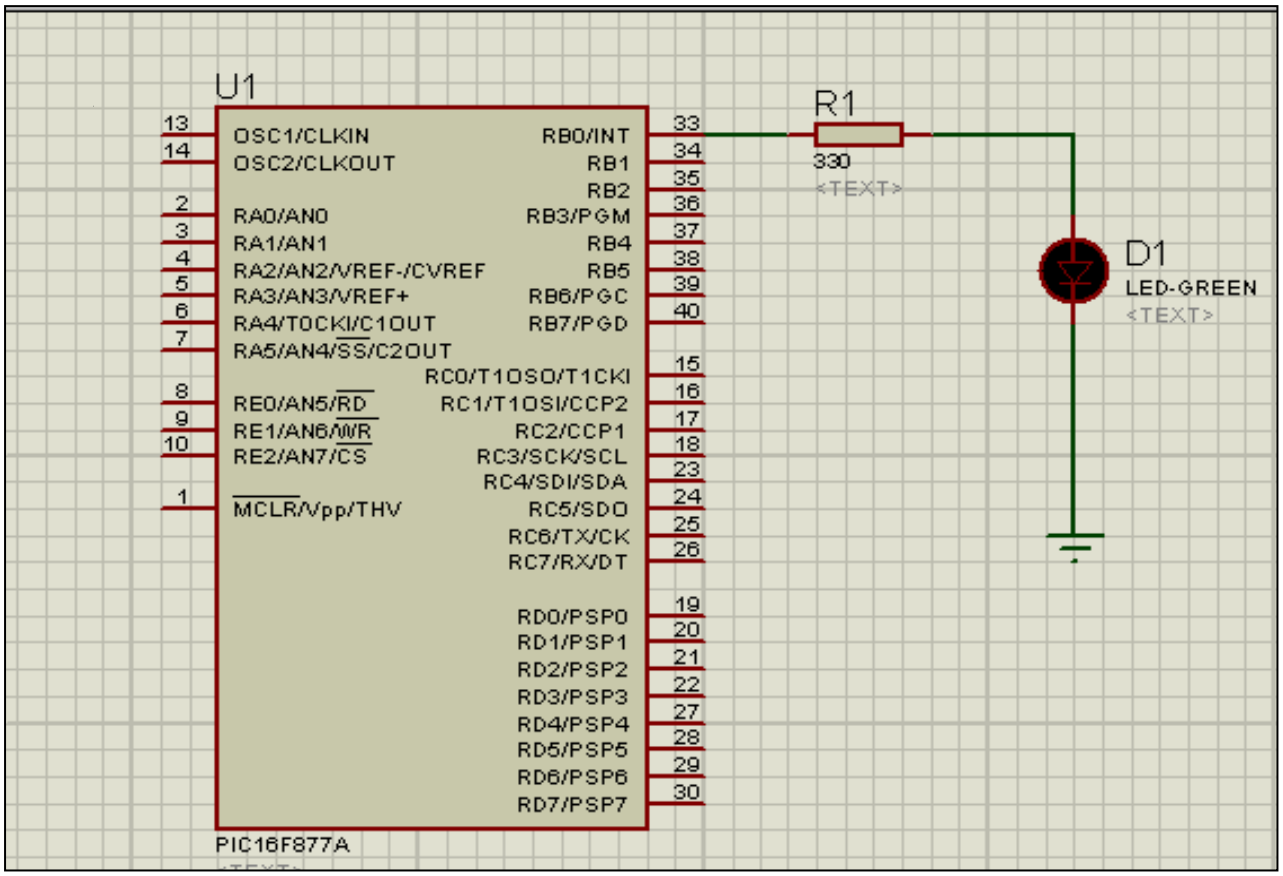

ثم ضع المقاومة بجوار RB0 وضع الليد ثم ضع الارضي GND كالتالي



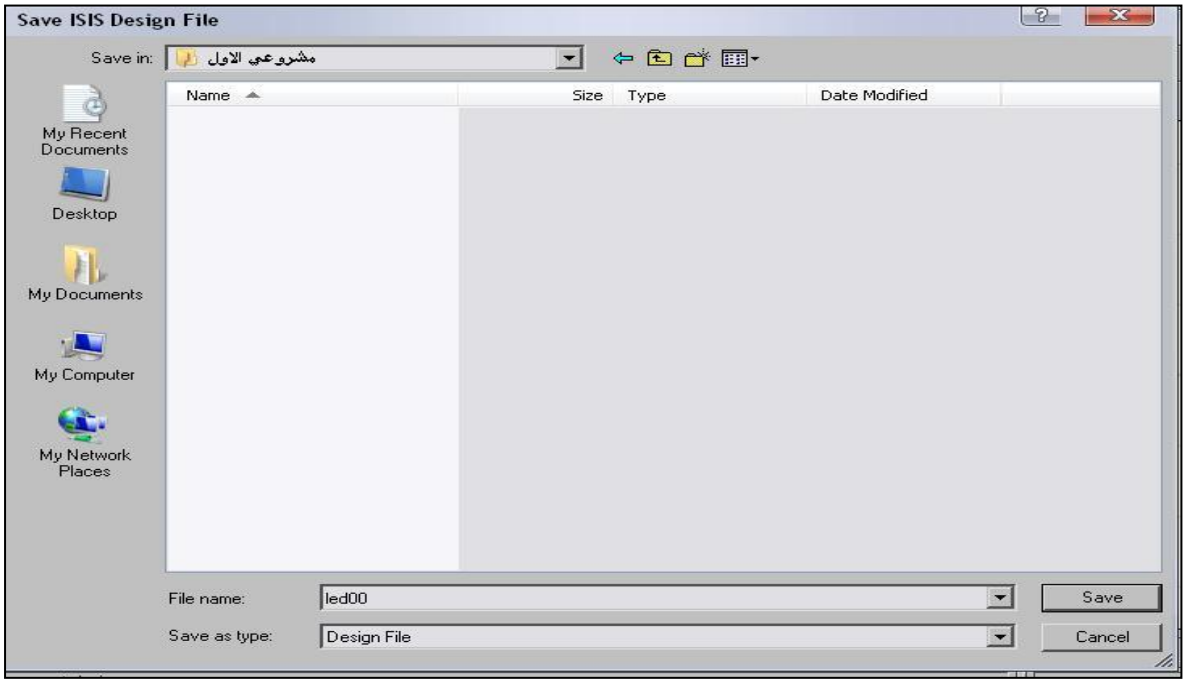

قم بالضغط على رمز السهم بعد وضع عناصر المشروع
قم بالضغط على نهاية المقاومة ثم ترك الزر الايسر للسهم
وتمرير السهم إلى النقطة RB0 ثم اضغط مرة بالماوس
ثم عدل قيمة المقاومة الى 330 بالضغط عليها مرتين بالماوس
كالتالي



ليصبح المشروع كهذا




بعد ذلك قم بحفظ المشروع من رمز save
وضعه في مجلد وتذكر اين وضعت المجلد لكي نحفظ مشروع ميكروسي معه



يأتي الدور المهم الان وهو مشروع ميكروسي ويجب عليك تذكر الاتي

يجب أن تكون نهاية الكود بالفاصلة المنقوطة ;
التي تسمى سمى كول
كي نهيأ البورت كخرج نستعمل الامر
TRISB.B0=0;
وسوف نشرح الامر
اولا امر TRIS ثم البورت واخترنا هنا بورت B ثم دوت "."
ثم B وهي تعني ثنائي ثم رقم الرجل او البن وهي 0
ثم نساويها بالصفر ليعرف ميكروسي أن البورت سوف تعمل كخرج ولا تنسى الفاصلة المنقوطة في نهاية السطر

الامر

PORTB.B0=1;
PORTB.B0=0;
الامر PORT نخبر ميكروسي هل سيتم توصيل الجهد 5 فولت على الرجل ام لا اذا كان يساوي واحد اذن يوصل اما اذا كان يساوي صفر اذن مغلق

ولان سرعة المعالجة للأمر 8 مليون في الثانية فسوف يتم ذلك بسرعة كبيرة ويجب وضع مسافة زمنية من خلال الامر
Delay_ms(1000);
والرقم 1000 هو المدة الزمنية وتقدر بالملي ثانية ويمكن تغيرها

الحلقة for
هذه الحلقة تقوم بتكرار الكود عدد محدد من المرات وشكلها كالتالي

For(1;2;3){

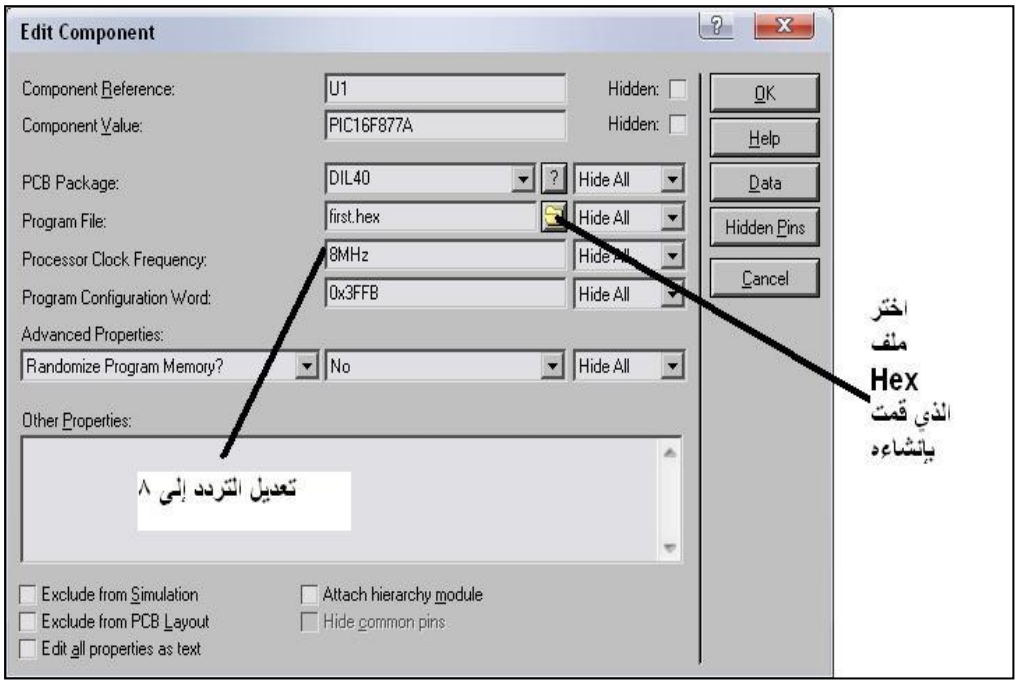
الكود المراد تكراره
}
المعامل 1 وهو بداية العد
المعامل 2 عدد مرات التكرار
المعامل 3 قيمة الزيادة
مثال
For(x=0;x<5;x++){

}
ولكن كي يتعرف ميكروسي على المعامل X يجب تعريفه
وذلك في قسم التصريحات العامة أي قبل الدالة الرئيسية
Void main()
نقوم بوضع متغير رقمي ثم وضع اسم المتغير
Int x;
والمتغيرات هي معاملات تسهل عملية كتابة الكود وهي ضرورية
ولا يمكن الاستغناء عنها لذلك لا تقلق
حيث أن قيمتها تتغير فقمنا بوضع قيمة ابتدائية للمتغير x وهي
صفر و قيمة x تتغير داخل الحلقة for حيث تزيد بمقدار واحد

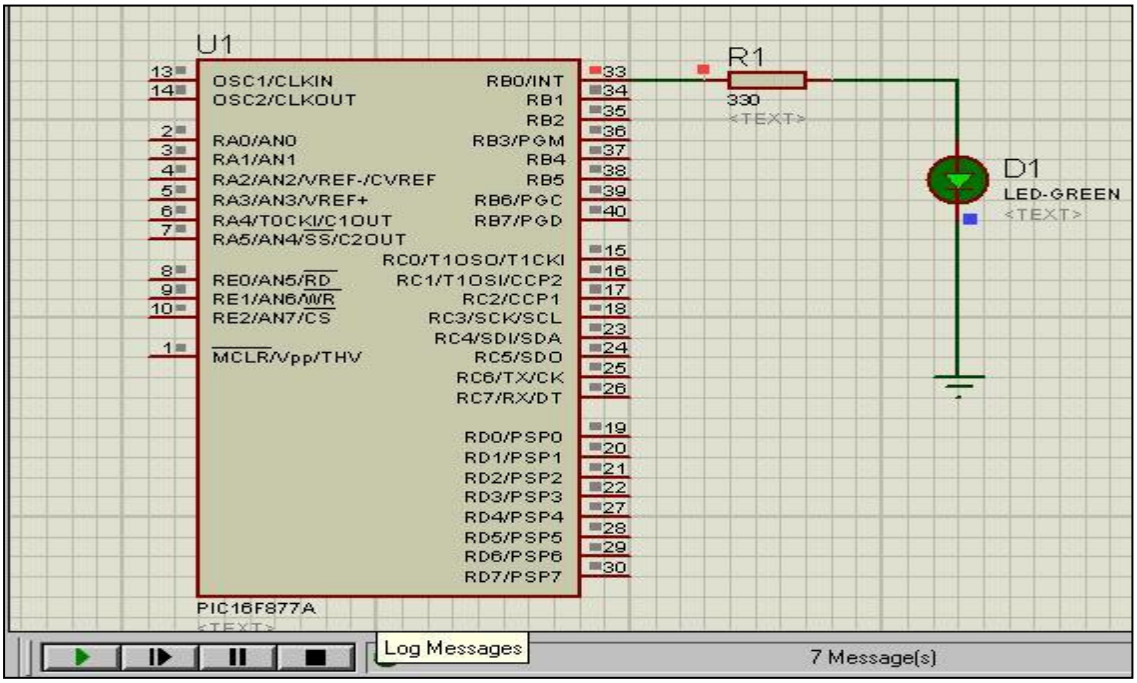
والكود النهائي كالتالي

```
 int x; //الاعلان عن متغير
void main() {
 trisb.b0=0; // وضع البورت كخرج
 for(x=0;x<8;x++){
  portb.b0=1;       //البورت مخرج جهد
  delay_ms(1000);   //زمن التوقف
  portb.b0=0;       //البورت مغلق
  delay_ms(1000);   // زمن التوقف
  }
}
```

بعد ذلك نقوم من قائمة Build بأختيار Build ليتم انشاء الملف Hex
ولكي نضيف ملف الهكس إلى مشروع بروتس قوم بالضغط على الميكروكنترولر داخل بروتس مرتين لتظهر النافذة التالية




قوم بتعديل التردد الى 8 ميجاهيرتز
ثم قم بأختيار ملف الهيكس الذي انشأته ثم اضغط OK
ثم نقوم بالضغط على زر التشغيل لتجربة المشروع



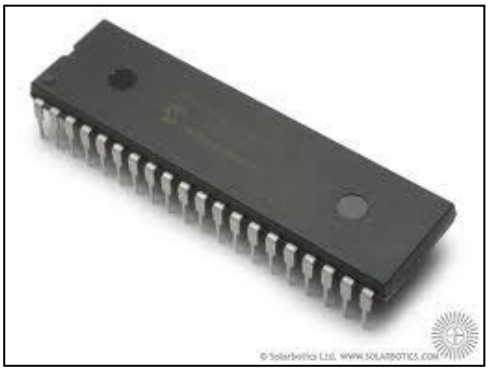
ملحوظة
لم نستخدم دائرة المذبذب ولا دائرة المستركلير ولم نوصل البور وذلك لانه يمكن الاستغناء عنهم في برنامج بروتس اما في الدائرة العملية فلا تعمل بدون الثلاث دوائر السابق ذكرهم

شكل العناصر الالكترونية في التجارب العملية

1- بطارية 9 فولت

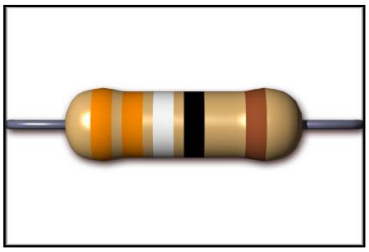
2- جاك بطارية

3- ميكروكنترولر PIC16F877A

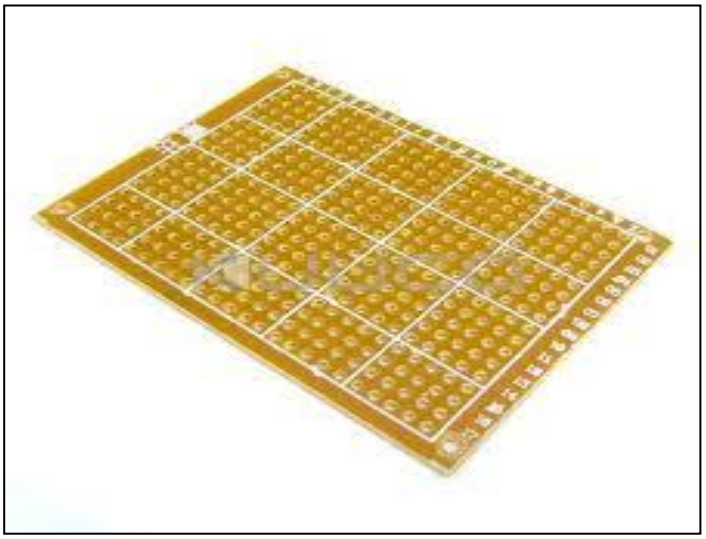


4- لمبه ليد

5- مقاومة



6- بورد تست او لوحة تجارب



**المشروع الثاني**

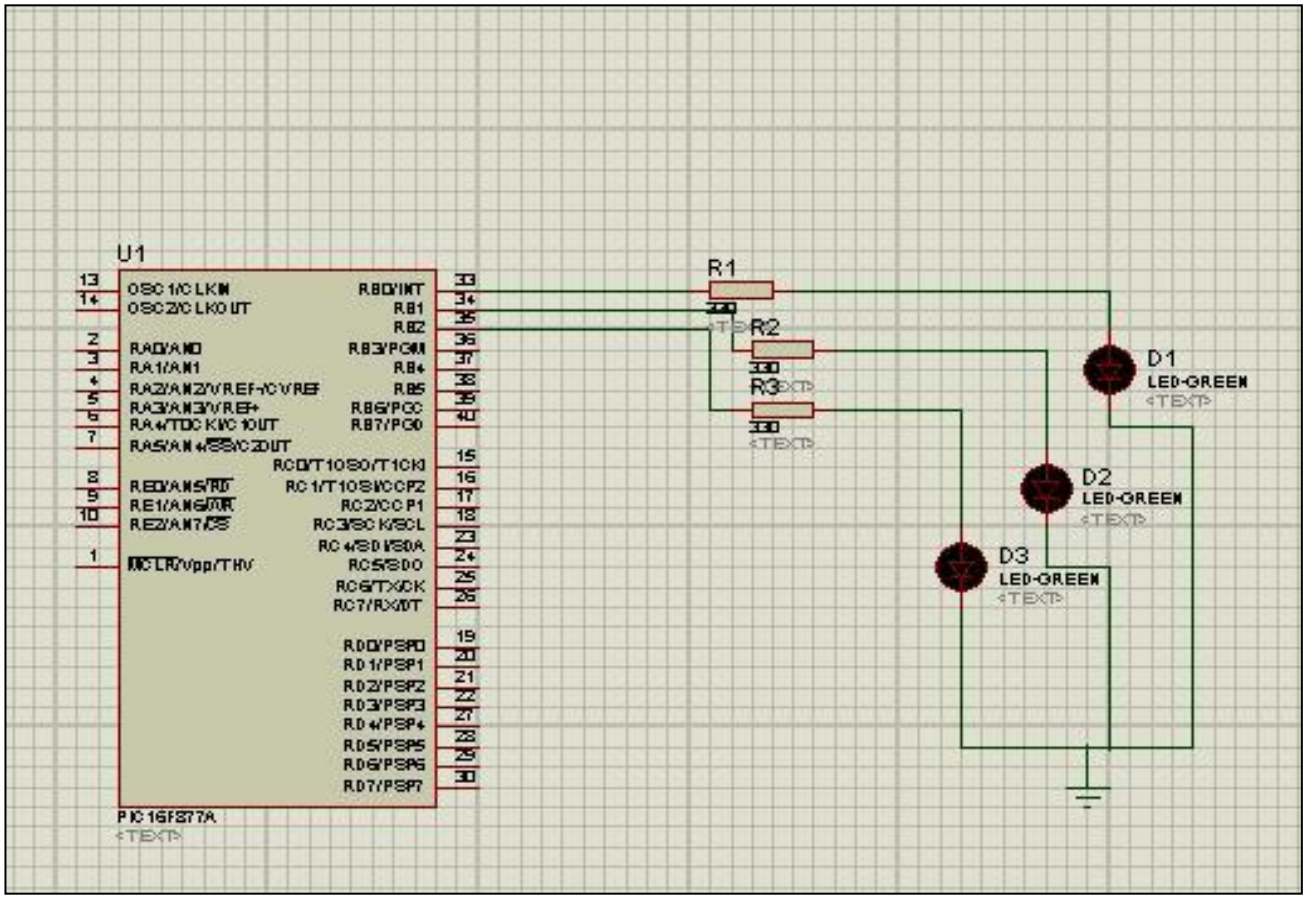
اضائة واطفاء أكثر من لمبة باستمرار

قم بفتح برنامج بروتس واحفظه في مجلد

وقم بإضافة العناصر الاتية

| | |
|---|---|
| led red | -1 |
| Resistor330 | -2 |
| PIC16F877A | -3 |

وارسم المشروع كالشكل التالي

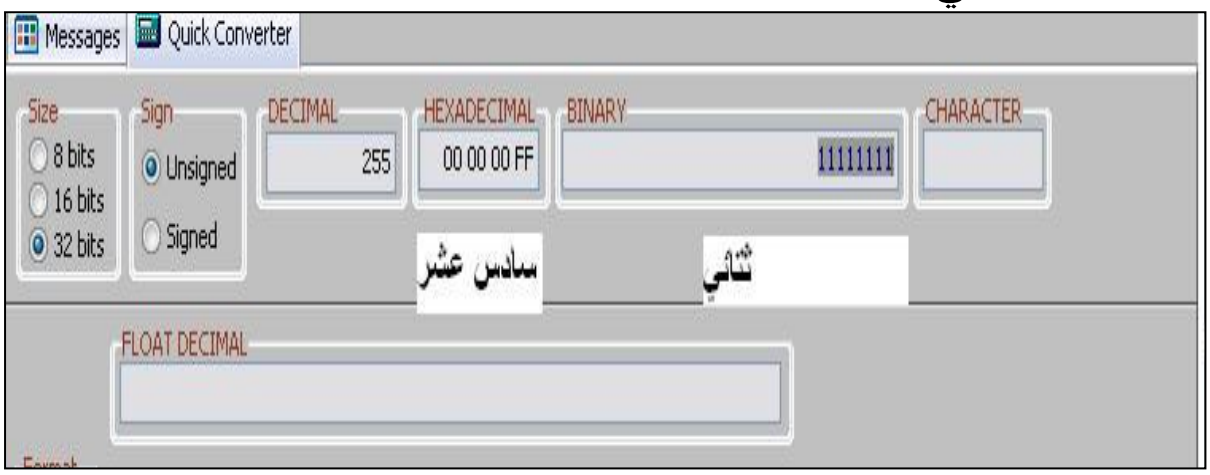


قم بفتح مشروع جديد ميكروسي
واحفظه كما تعلمت سابقا
والان هدف المشروع أن تضيأ الليدات وتغلق إلى مالا نهاية لذلك سوف نستخدم الامر
While(1){

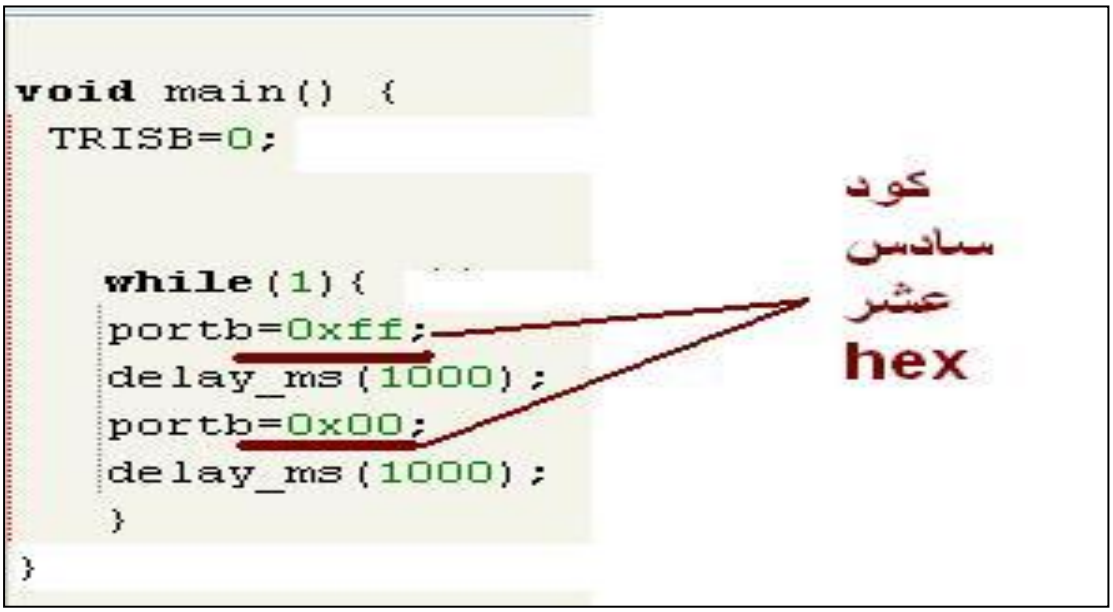
}
وهذا الامر عندما نضع الشرط 1 يعمل إلى ما لانهاية
ثانيا لاحظت في رسم التصميم أكثر من ليد لذلك سوف نستخدم الامر
TRISB =0B00000000;
وهذا يعني أن كل البورت من B0 الى B7 سوف تستخدم كمخرج
وهذا ينطبق مع الامر
PORTB ليصبح ;PORTB=0B11111111 وهذا يعني أن كل منافذ B تعمل
واليك الكود كاملا

```
void main() {
 TRISB=0; // وضع البورت كخرج


   while(1){ // حلقة تنفذ إلى مالا نهاية
   portb=0b11111111;      //البورت كله يضيئ
   delay_ms(1000);  //زمن التوقف
   portb=0b00000000;      // البورت كله مغلق
   delay_ms(1000);  // زمن التوقف
    }
}
```

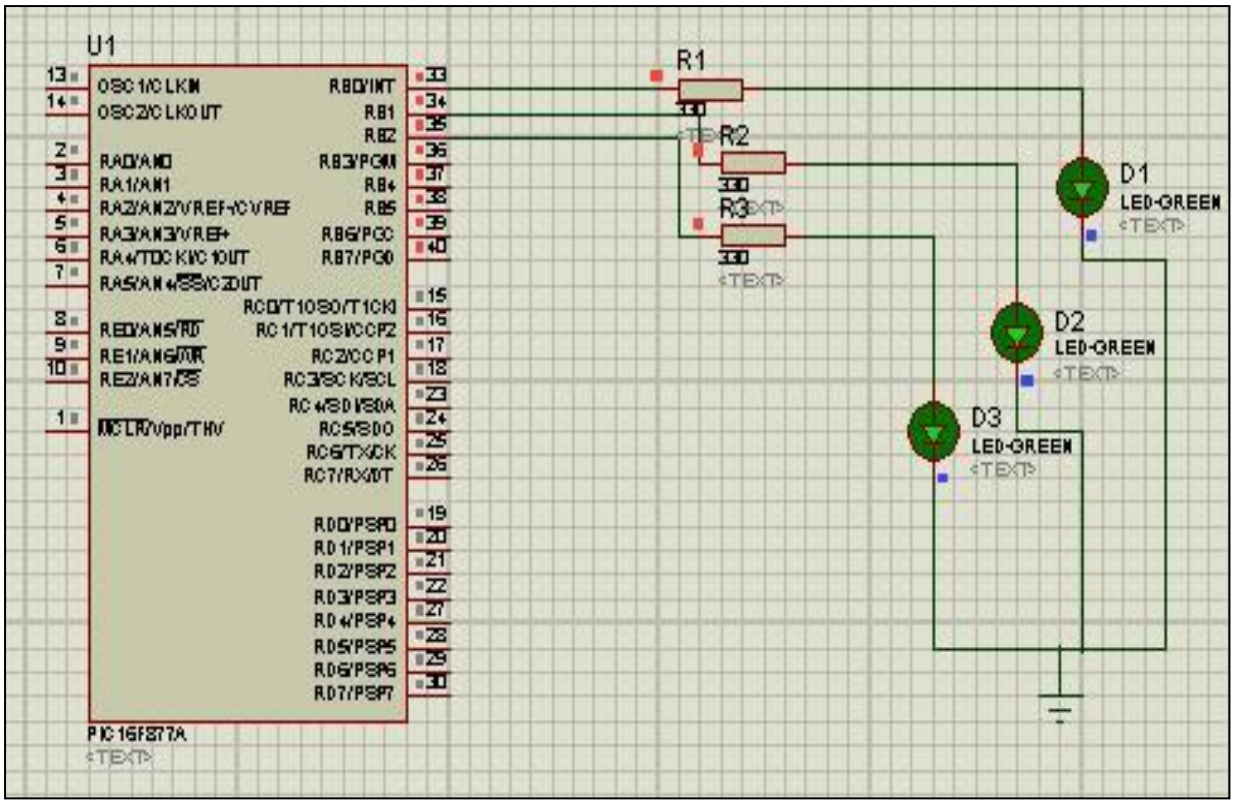
لاحظ مع الاتي
0b تعني ارقام ثنائية تأخذ قيمة اما 0 او 1
بما ان هناك 8 ارجل سوف نستخدم فسوف يصبح 0b00000000
يمكن التحويل إلى ارقام سادس عشر كالتالي
نضيف 0x ونحولها من خلال التبويب Quick Converter
حيث نكتب الرقم ثنائي ويتم تحويله إلى ارقام سادس عشر كما في الشكل التالي




ليصبح الكود كالتالي




وتصبح التجربة كهذا



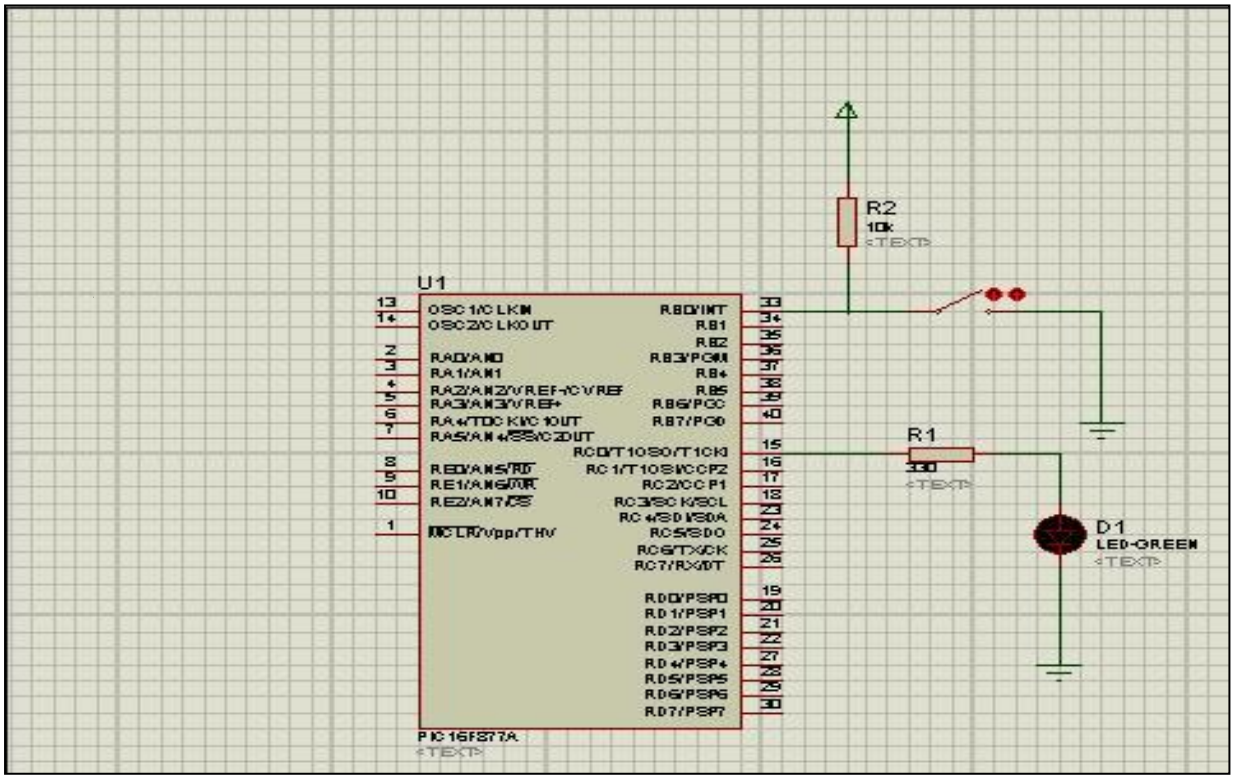

**المشروع الثالث**

وهو اضافة مفتاح غلق وفتح يقوم بأضائة الليد وغلقه

اولا نقوم بإنشاء مشروع بروتس جديد وحفظه
ثم اضافة العناصر الالكترونية التالية

Led green
Switch
PIC16F877A
Resistor

وارسم المشروع بالشكل التالي




لاحظ معي الاتي أولا أن لم يتم توصيل المفتاح مباشرة بل تم توصيله بجهد 5 فولت ومقاومة 10 كيلو وهكذا يكون الدخل عالي او 1 وعند الضغط عليه يوصل الجهد بالارضي ويصبح دخل الرجل 0 فولت أو منخفض

افتح مشروع ميكروسي

ولاحظ الاتي

أولا كي نجعل الرجل B0 كدخل نكتب الامر التالي

```
TRISB.B0=1;
```

وكذلك سوف نستخدم اداة الشرط IF

و if تعني "اذا" وصيغتها كالتالي

```
if ( 1 ) {
2
}
```

1 الشرط

2 جواب الشرط

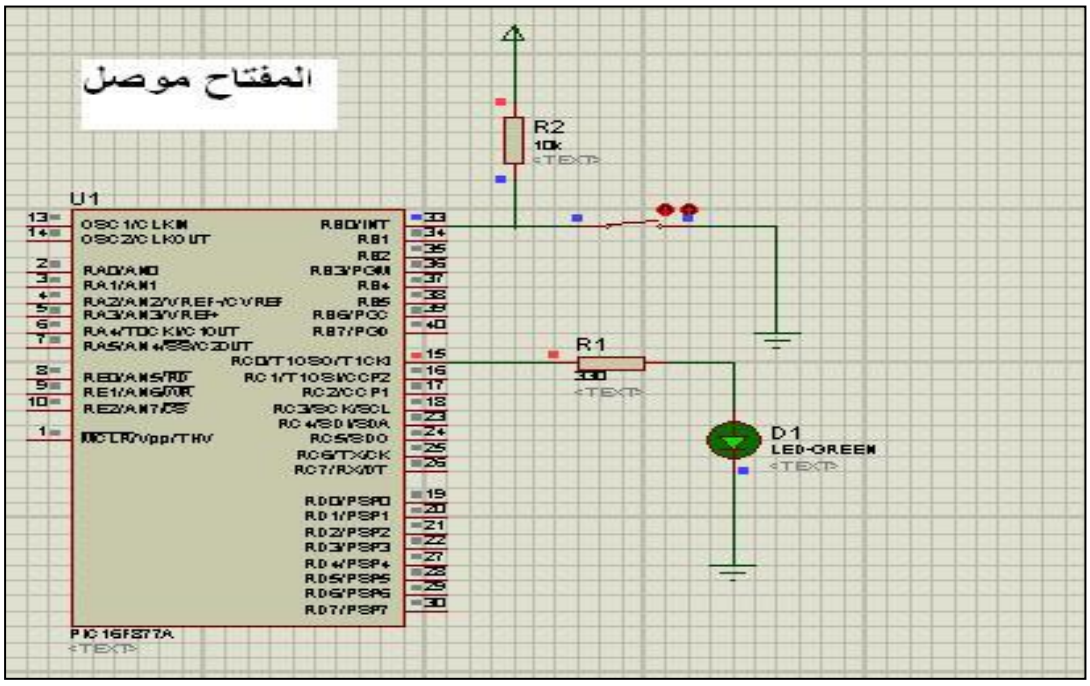
فنحن نريد عند الضغط على المفتاح تضيأ اللمبة فتكتب كالتالي

```
If(portb.b0==0){
```

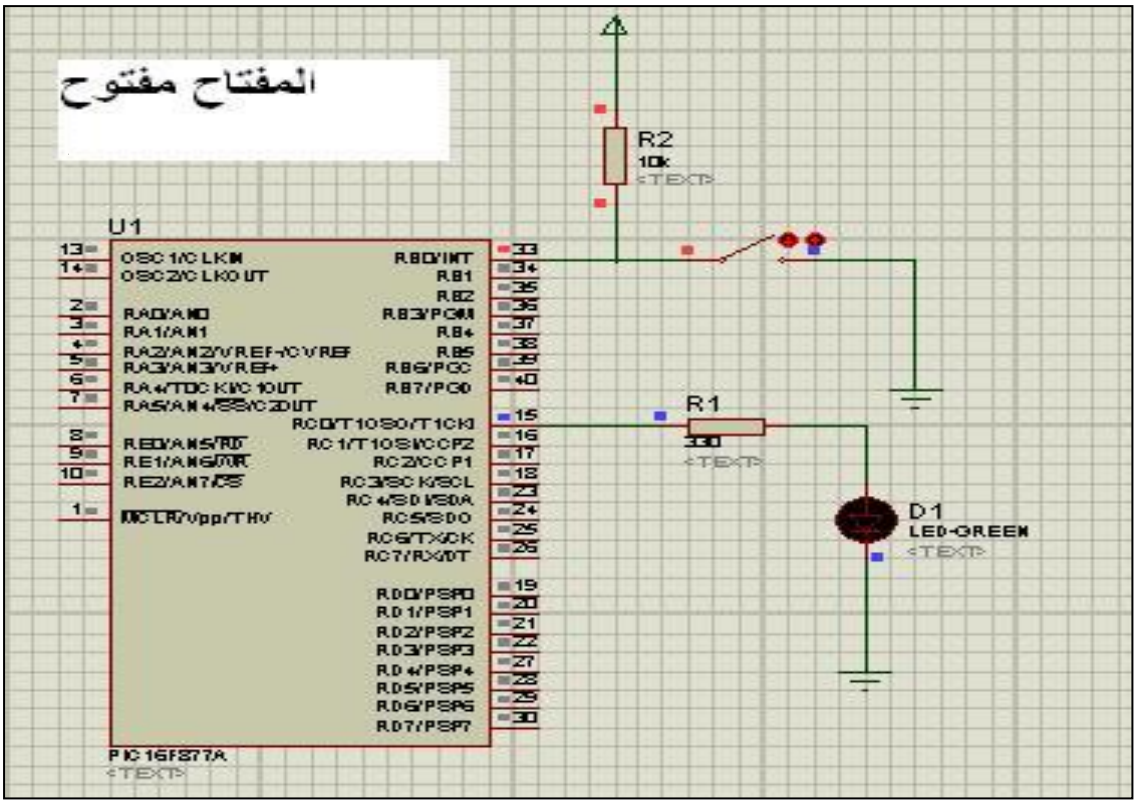
Portc.b0=0;
}
وكي يضيئ ويطفأ نستخدم العبارة else
وتعني وإلا
وفي الشكل التالي الكود

```
void main() {
 trisb.b0=1; // نستخدم البورت كدخل
 trisc.b0 =0; // نخرج

   while(1){ // حلقة تنفذ إلى مالا نهاية
   if (portb.b0==0){  // اذا كان المفتاح مغلق
    portc.b0=1;  // اذن شغل الليد

   } else{ // وإلا
    portc.b0=0; // اغلق الليد
   }
}
}
```

أضف ملف hex إلى الميكروكنترولر في بروتس وعدل التردد إلى 8 ميجاهيرتز
لتصبح النتيجة كالتالي
عند توصيل المفتاح هكذا



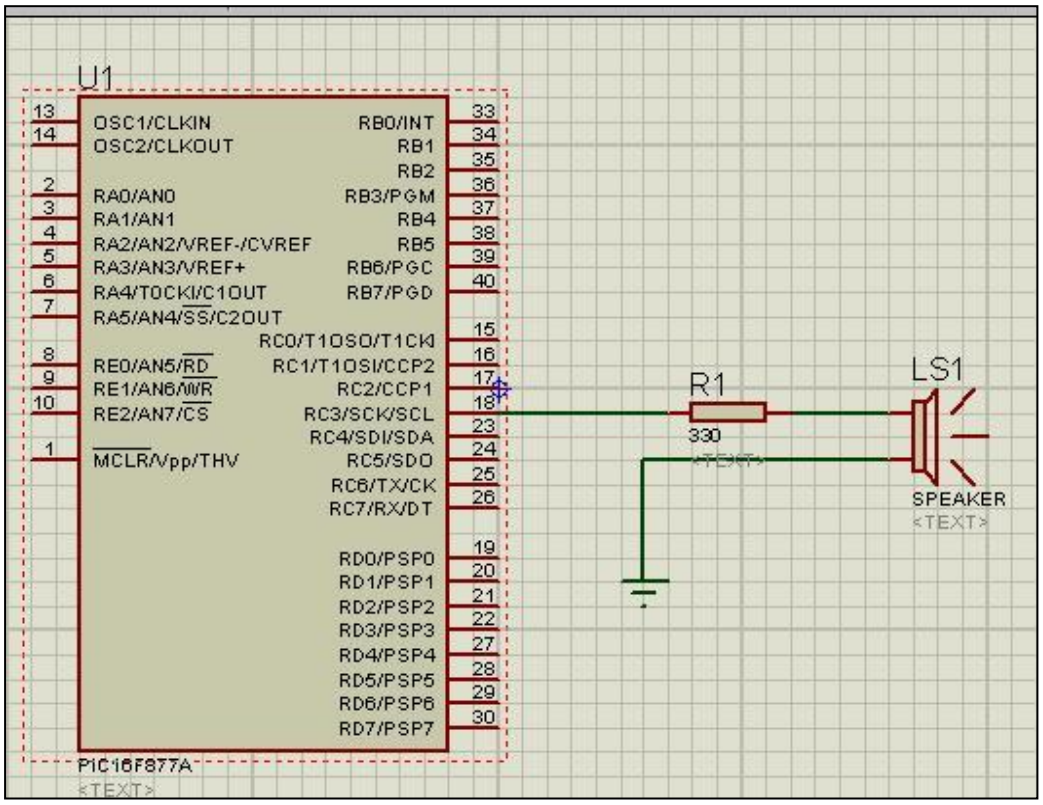
وعند فصل المفتاح هكذا




ونكتفي بهذا القدر من المشاريع في الليد ماتركس ونتجه إلى مشاريع إخراج الصوت في الفصل القادم

# الفصل الخامس
# النغمات في الميكروكنترولر

أخراج الصوت يكون عن طريق
المكتبة sound
افتح مشروع بروتس جديد وأضف العناصر الالكترونية التالية




| | |
|---|---|
| speaker | -1 |
| resistor | -2 |
| PIC16F877A | -3 |

ثم افتح مشروع ميكروسي جديد وسميه sound

واكتب الكود التالي

```
int i;

void main() {
trisc=0;
sound_init(&portc,3);
//المعامل الاول اسم البورت والثاني رقم الرجل
while(1){
 for(i=0;i<1000;i=i+50){
   sound_play(i,1000);
  // المعامل الاول التردد والثاني الزمن بالملي ثانية
 }
}
}
```

والمكتبة واضحة
فمعامل sound_init للتعريف بالبورت ورقم الرجل الذي سيتصل بالسماعة
واما sound_play لعمل السماعة ولها معاملان الاول التردد والثاني زمن التردد بالملي ثانية ويمكنك البحث عن نغمات في الانترنت

وبهذا نكون قد انهينا الفصل الخامس الخاص بالنغمات واخراج الصوت

هذا والله الموفق

**بسم الله الرحمن الرحيم**

**تأليف عيد فتحي**

# مكتبة Button

**تقوم المكتبة بالتأكد من ضغط الزر**

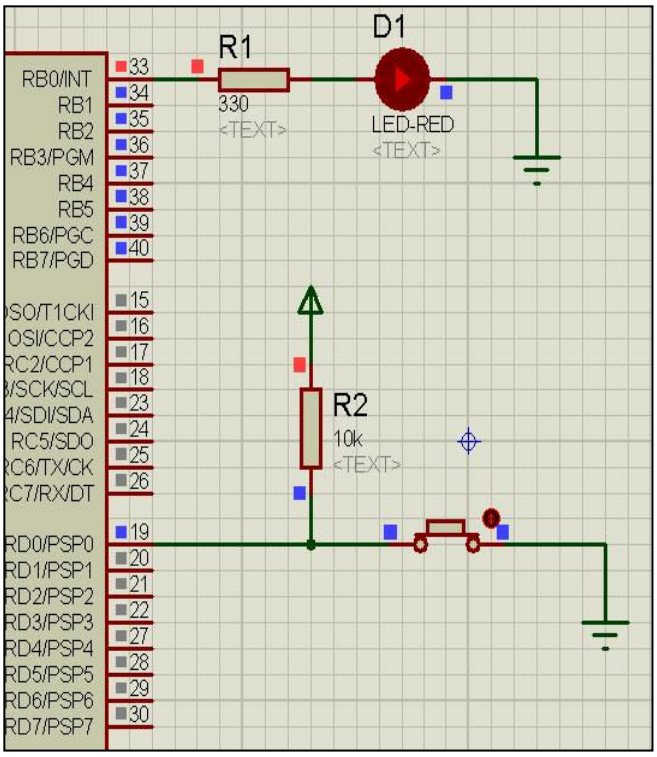
**له اربع معاملات**

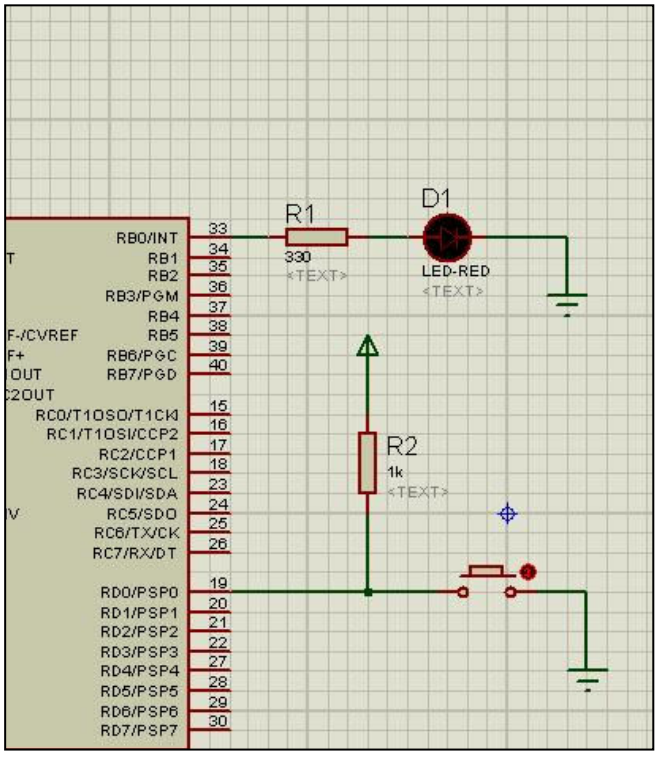
**١- اسم البورت الموصل بالزر ويسبقه علامة &**
**٢- رقم الدبوس او البن في البورت**
**٣- زمن ضغط الزر ويقدر بالمللي ثانية**
**٤- الزر مفعل 1 او غير مفعل 0**

**بمعنى هل يعمل على المنطق العالي او المنخفض ودائما ما نعمل بالمنطق المنخفض لذلك يأخذ القيمة صفر**

**مثال**

```
void main()  {
 trisb=0;
 portb=0;
 trisd.b0=1;
while(1){
 if(Button(&PORTD,0,10,0)){

  portb.b0=1;

 }else{
  portb.b0=0;


 }

}
}
```




**هذا والله الموفق (عيد فتحي)**

بسم الله الرحمن الرحيم

التعامل مع الجهود العالية
تأليف عيد فتحي

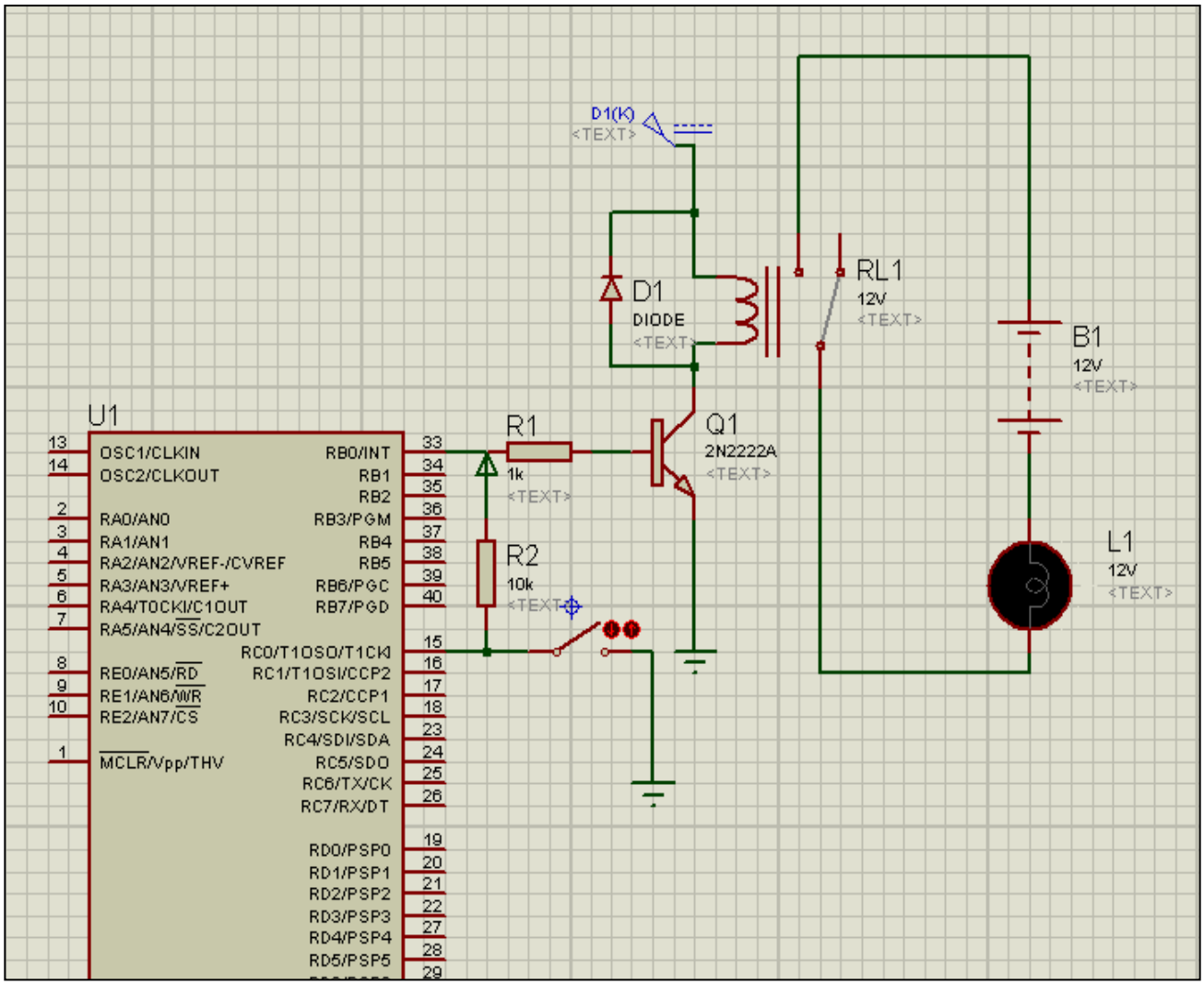
حتى الان تعاملنا مع الجهود المنخفضة التي لا يتجاوز تيارها ٢٠ ملي امبير ولكن ماذا لو اردنا التحكم في محرك سواء DC or AC ماذا سنفعل وكيف سيتحكم الميكروكنترولر الصغير في الجهود العالية ابقى معي ياصديقي حتى تعرف

اولا سوف نقوم بعمل دائرة صغيرة يتم فصل وتوصيل لمبة على جهد بطارية ١٢ فولت وسوف يكون الكود كالتالي

```
void main()  {
 TRISB.b0=0;
 TRISC.b0=1;
 portb=0;
 while(1){
  if(PORTC.B0==0){
   portb.b0=1;
  }else{
   portb.b0=0;

  }
 }
}
```

كما يتضح من الكود اننا سوف نجعل بورت B0 كخرج وبورت C0 كدخل أي بورت B0 سوف يتصل بالجهد العالي وبورت C0 يتصل بمفتاح كي يتم التحكم من خلالة

لاحظ الدائرة الاتية




اولا سوف نستعمل الترانزستور كمفتاح حيث أن البورت B0 عندما يصله الجهد سوف تكتمل الدائرة وتعمل وعند فصل البورت B0 لان تكتمل الدائرة ولن تعمل سوف تكون دائرة مفتوحة
وسوف نقوم بحساب المقاومة RB للترانزستور كالتالي
RB=2.2*RL*HFE
حيث RB مقاومة القاعدة و RL مقاومة الحمل و HFE معامل التضخيم ويتم الرجوع الى الداتا شيت للترانزستور المستعمل ولكن نحن في الغالب نستعمل ترانزستور 2N2222 وهو معلوم سلفا أنه يستخدم مقاومة ١ كيلو اوم

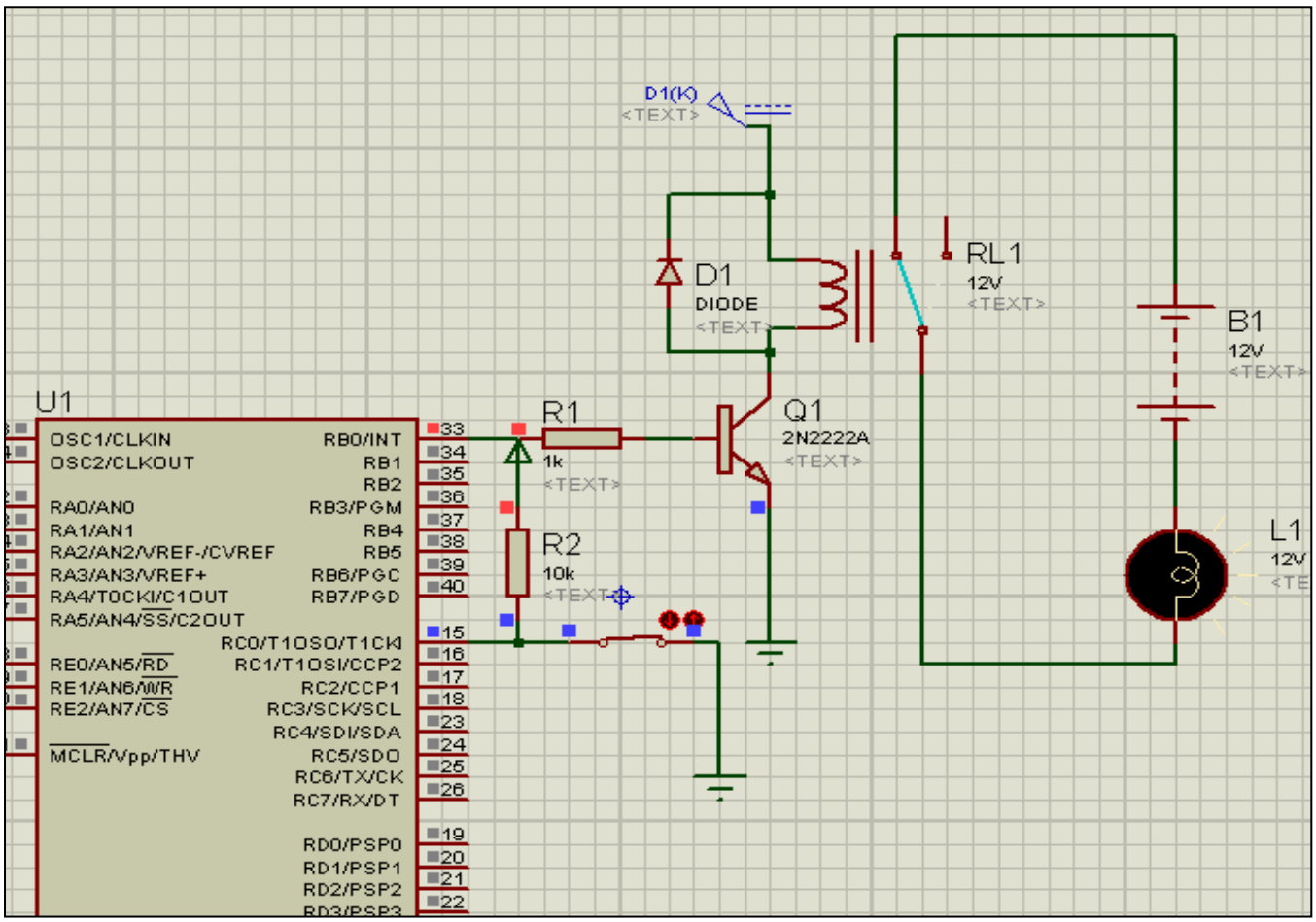
ثانيا نستخدم دايود او موحد حتى لا يرتد أى جهد كهربي الى الميكروكنترولر فيتسبب في تلفه ومن المعروف أن الدايود يمرر

الجهد الكهربي في اتجاه واحد لذلك سوف نقوم بعكسه حتى لا يرتد الجهد الكهربي ومن الجدير بالذكر أن الدايو يتم شرائه حسب قيمة التيار الكهربي لذلك سوف نستخدم دايو ٣٠٠ ملي امبير

كما أن توصيل الدايود بالعكس يستخدم في دائرة شحن البطاريات

ثالثا نستخدم ريلاي للتوصيل مع الجهود ٢٢٠ فولت تيار متردد والريلاي له خمس اطراف طرفين يوصلان مع دائرة الميكروكنترولر وثلاثة مع الدائرة الخارجية ذات ٢٢٠ فولت

لتصبح النتيجة كالتالي



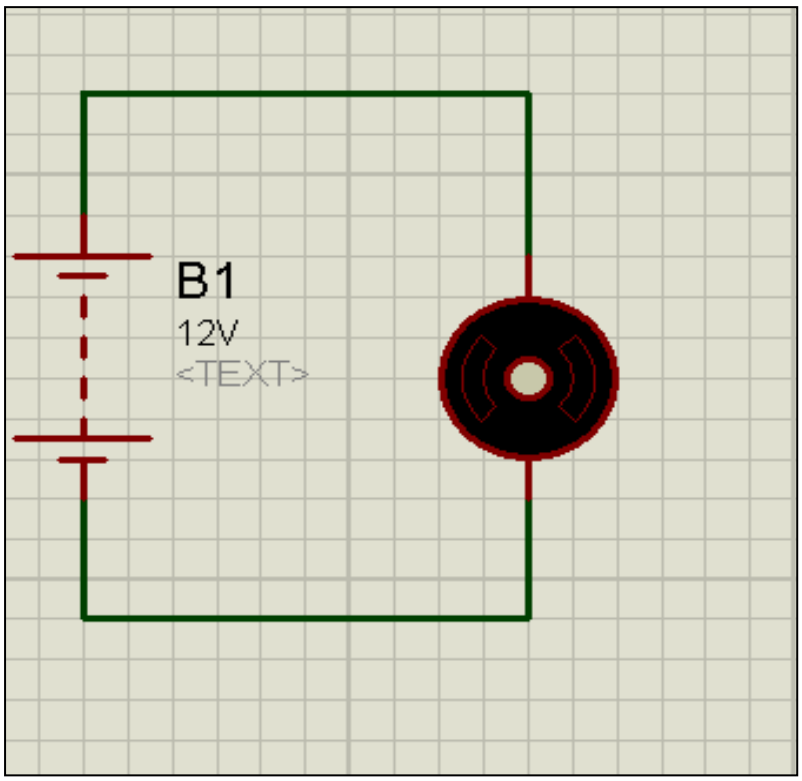

والله الموفق واليه المصير

عيد فتحي

بسم الله الرحمن الرحيم

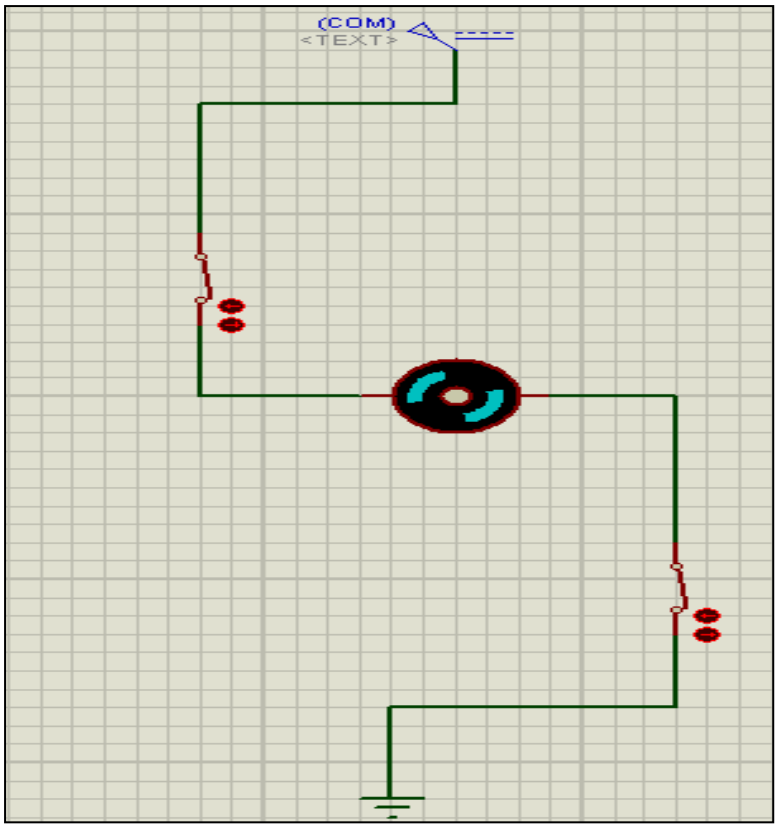
التحكم في اتجاه محرك

يمكنك عكس اتجاه محرك DC اذا قمت بعكس اتجاه التيار الكهربي وذلك بعكس اقطاب التوصيل للمحرك DC

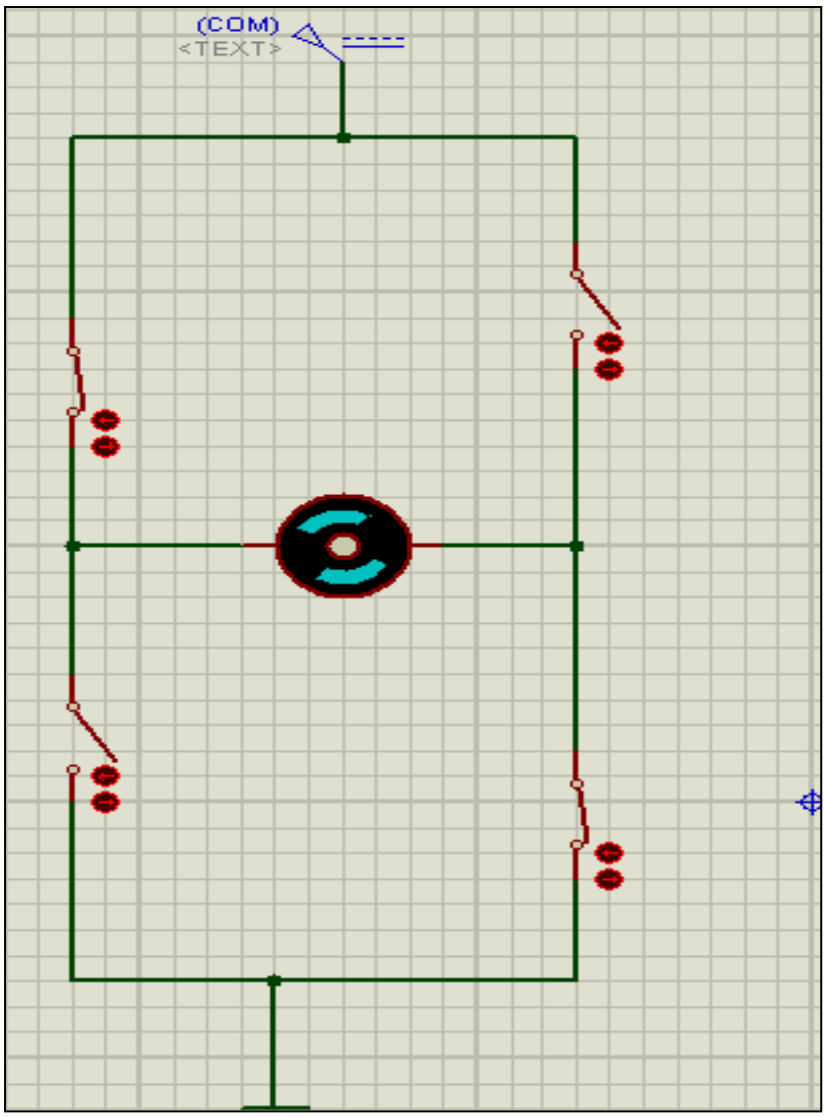



ويمكن فعل ذلك من خلال دائرة القنطرة H حيث انها تشبه حرف H

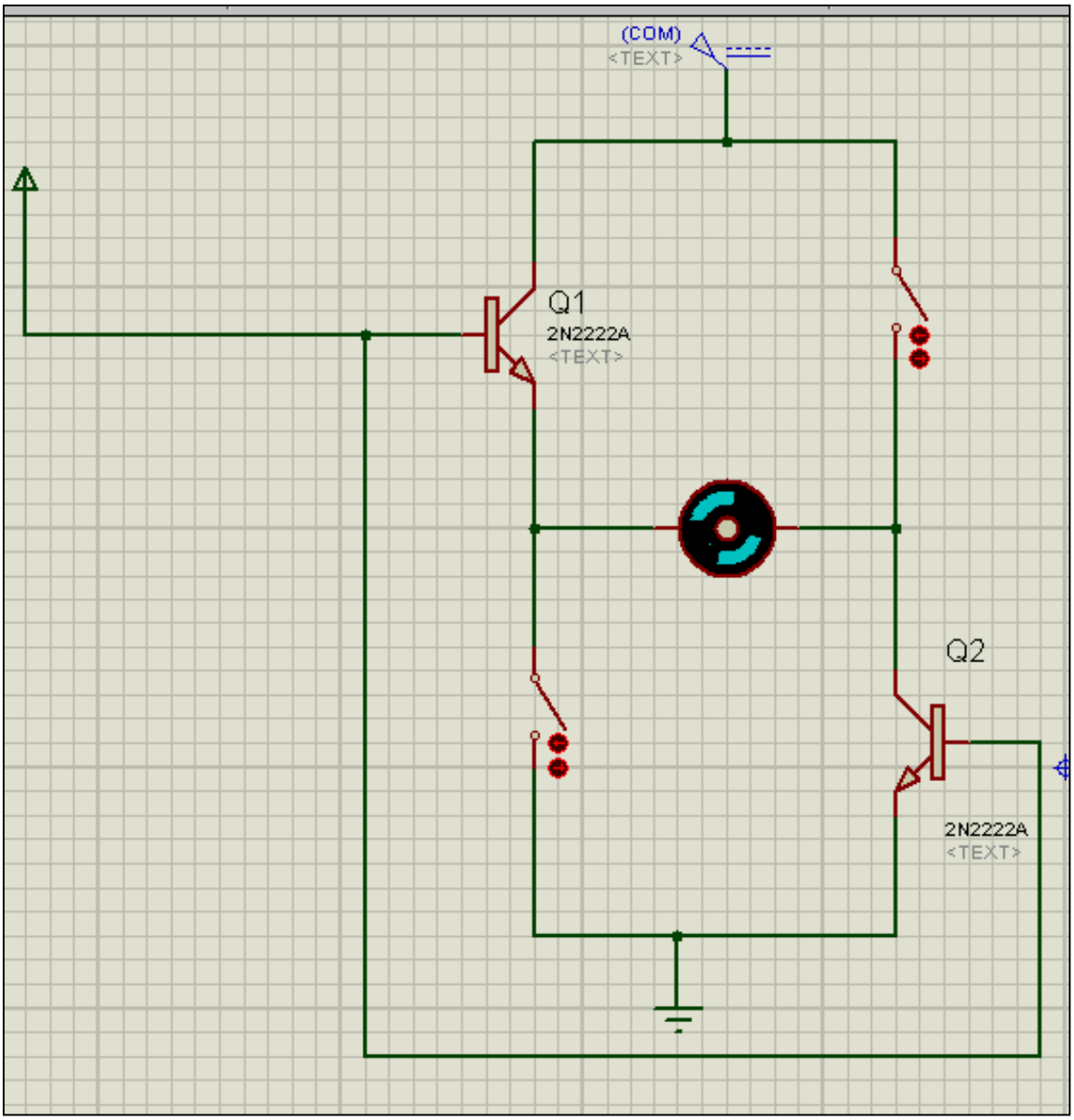
لاحظ معي اذا تم توصيل المحرك من احد الاطراف بالطرف الكهربي والاخر بالارضي هكذا




ولو اردنا عكس اتجاه المحرك سوف نستخدم طرفين اخرين
بالاطراف المعاكسة للطرفين الاولين كما في الرسم

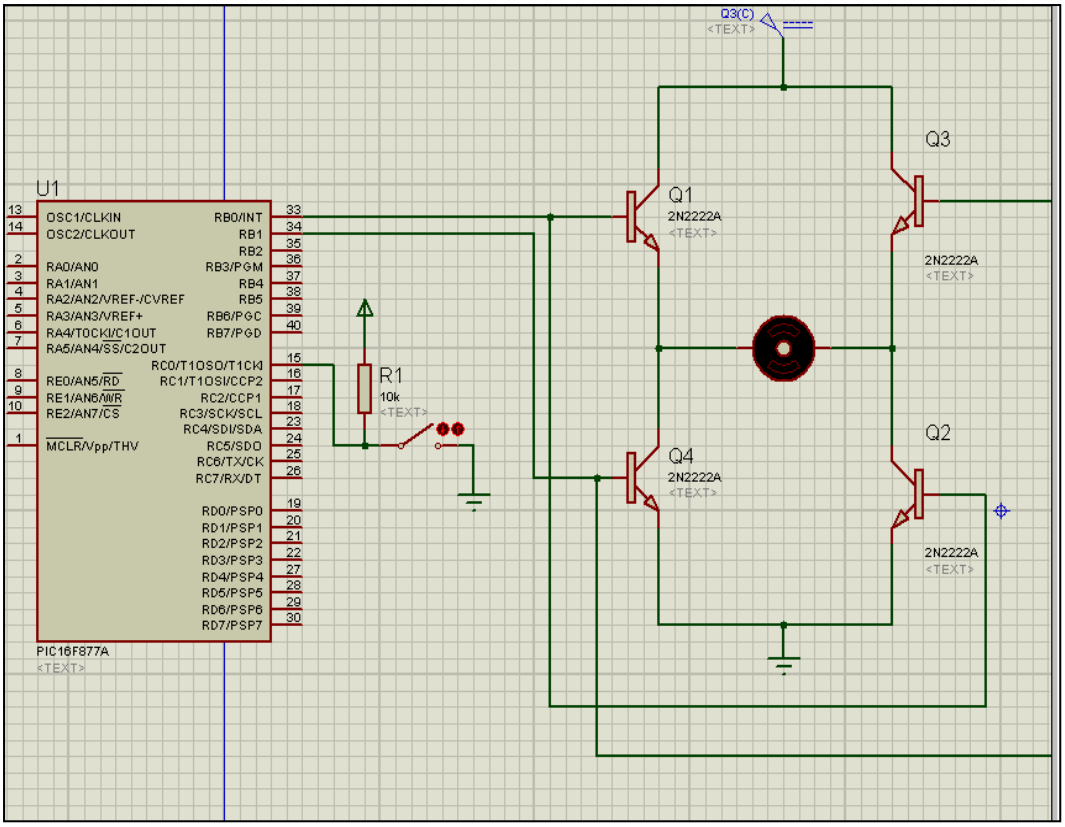



ولكن كيف يمكن استبدال المفاتيح بالترانزستور
اولا سوف نستخدم دائرة  الترانزستور كمفتاح ويتم التحكم من خلال طرف القاعدة B للترانزستور
ويتم التوصيل كالتالي



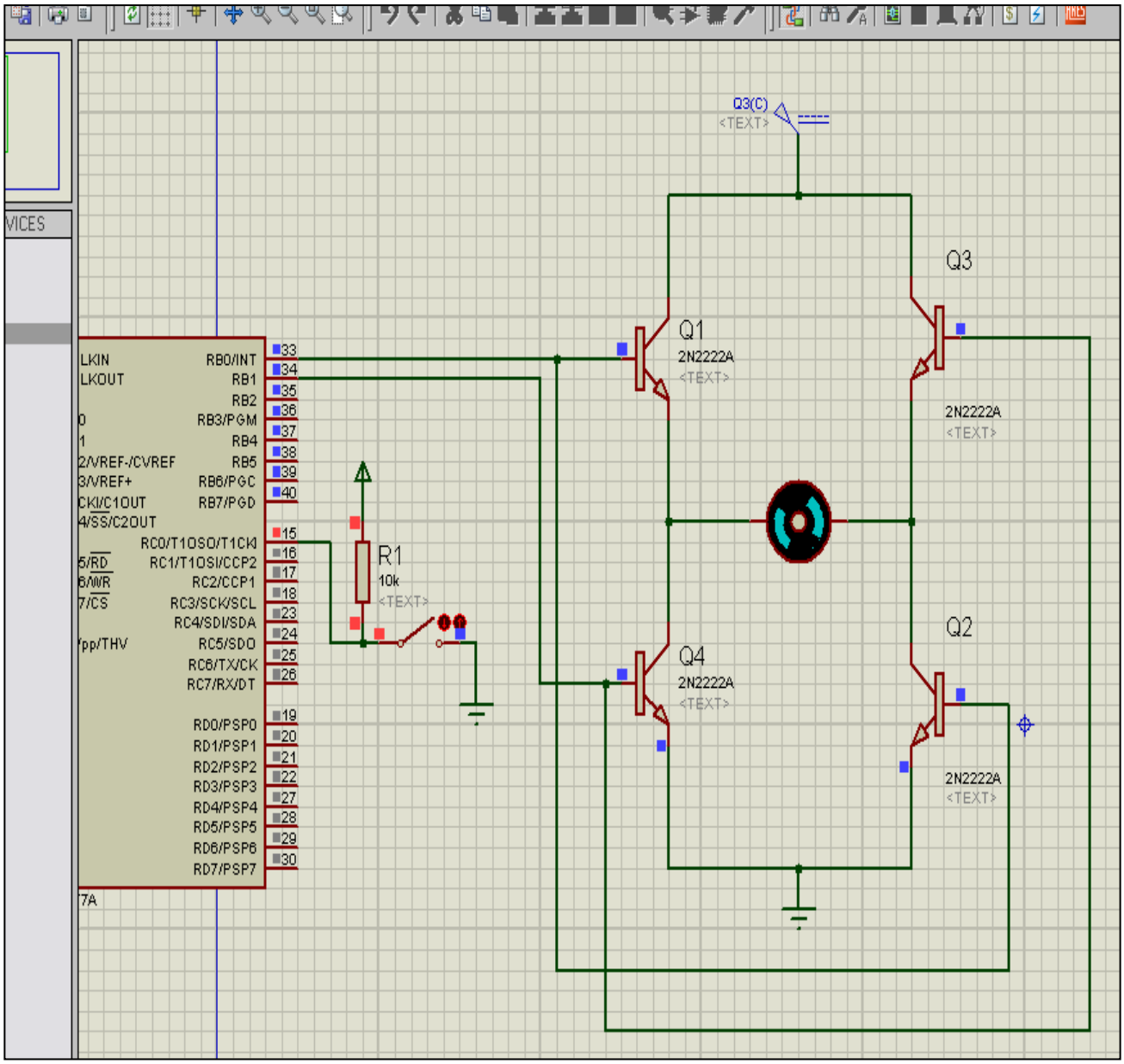

تم توصيل اطراف القاعدة للترانزستور ١ مع ٣ ثم يتم توصيل اطراف القاعدة ٢ مع ٤

ويتم التحكم بعد ذلك من خلال الميكروكنترولر




والكود كالتالي

```
void main()  {
TRISB=0;
TRISC.B0=1;
PORTB=0b00000001;
while(1){
 if(PORTC.B0==0){
 portb=0b00000010;

 }else{
  portb=0b00000001;

 }
}
}
```




والله الموفق واليه المصير

عيد فتحي

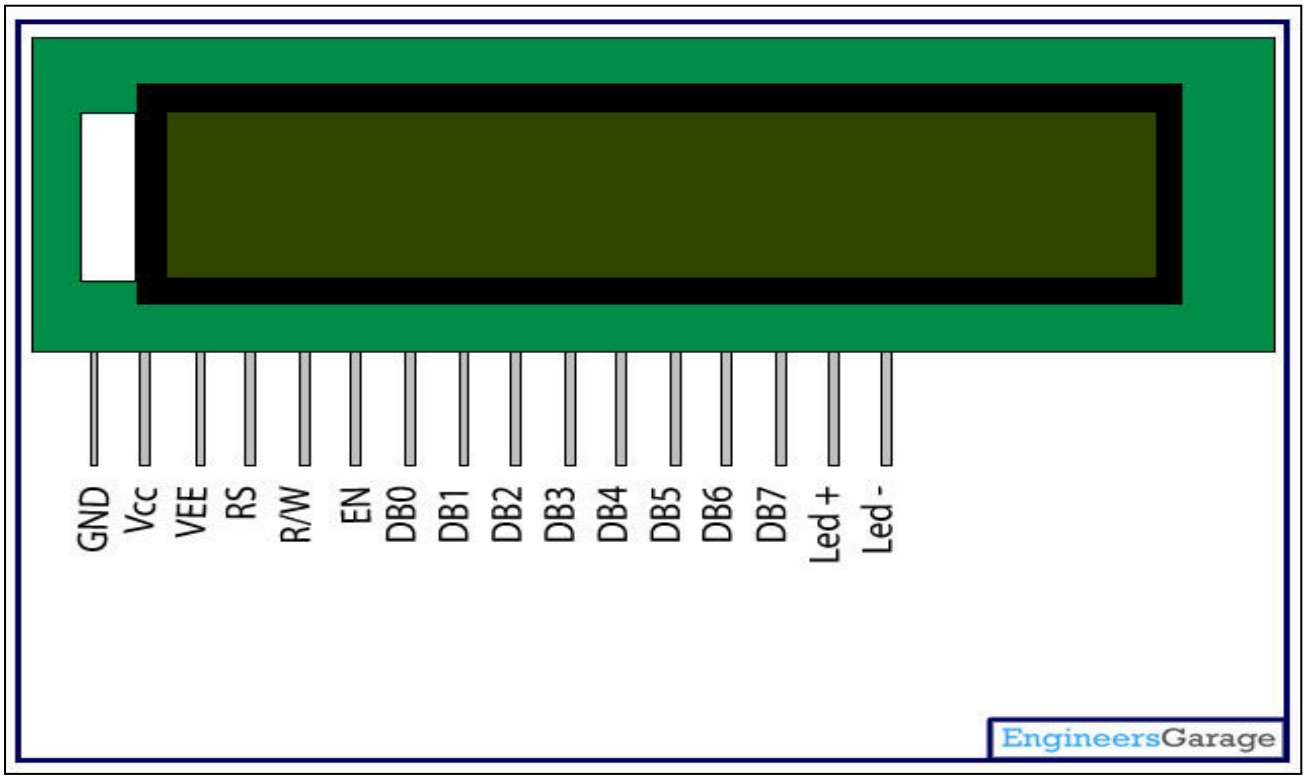
# بسم الله الرحمن الرحيم

# الفصل الثالث

# الشاشات LCD

تأليف : عيد فتحي

## اولا شكل الشاشة ووظيفة كل طرف



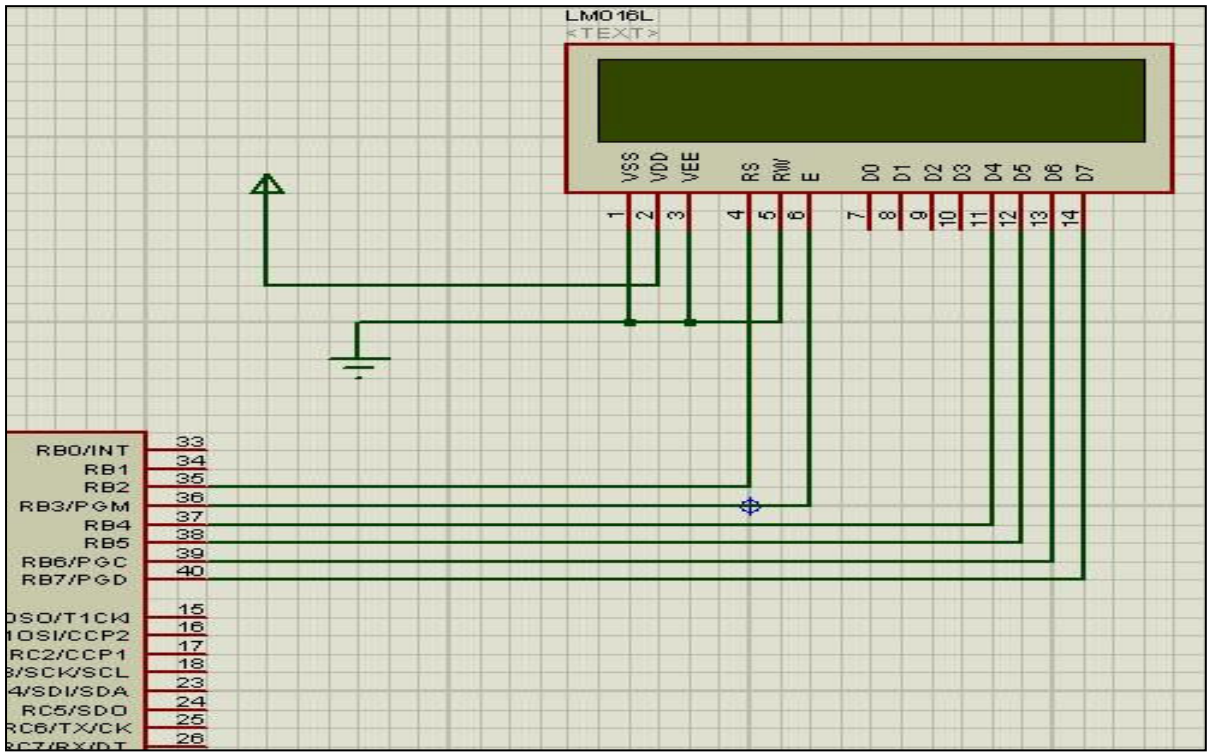

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

١- Vss ارضي
٢- Vdd كهربي ٥ فولت
٣- Vo ضبط المراقبة
٤- RS تحديد أشارة السجل
٥- R/W قرائة / كتابة الاشارة
٦- EN تفعيل بيانات القرائة والكتابة
٧- D0 خط نقل البيانات غير مستخدم
٨- D1 خط نقل البيانات غير مستخدم
٩- D2 خط نقل البيانات غير مستخدم
١٠- D3 خط نقل البيانات غير مستخدم
١١- D4 خط نقل البيانات مع المعالجة
١٢- D5 خط نقل البيانات مع المعالجة
١٣- D6 خط نقل البيانات مع المعالجة
١٤- D7 خط نقل البيانات مع المعالجة
١٥- +LED الطرف الموجب لاضائة الشاشة ٥ فولت
١٦- -LED الطرف الارضي لاضائة الشاشة

## ثانيا توصيل الاطراف بالميكروكنترولر PIC16F877A

الطرف ١ و٣ و٥ بالارضي GND
الطرف Vdd مع الكهربي ٥ فولت
الطرف RS مع الطرف RB2
الطرف EN مع الطرف RB3
الطرف D4 مع الطرف RB4
الطرف D5 مع الطرف RB5
الطرف D6 مع الطرف RB6
الطرف D7 مع الطرف RB7
الطرف +LED مع الكهربي ٥ فولت
الطرف -LED مع الارضي

## كما في الشكل التالي




## ملحوظة

**لا يوجد في بروتس طرفان اضائة الشاشة**

## ثانيا شرح مكتبة LCD

Start Page | temp.c

```
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;
```

**هذا الكود خاص بتعريف أطراف الشاشة الموصلة مع الميكروكنترولر**

**ثانيا**

```
Lcd_Init();
```

**خاص بتعريف مكتبة LCD**

**ثانيا**

**Lcd_Cmd**

**هذا الامر خاص بأرسال الاوامر الى الشاشة**
**وله معامل وهو أمر الارسال**
**مثال**

```
Lcd_Cmd(_LCD_CLEAR);
```

**حيث يقوم هذا الامر بمسح الشاشة ويستخدم عادة في مقدمة البرنامج**

**Lcd_Cmd(_LCD_CURSOR_OFF);**
**هذا الامر خاص بإيقاف مؤشر الكتابة في الشاشة ويكتب ايضا في بداية البرنامج**
**Lcd_Cmd(_LCD_FIRST_ROW);**
**ينقل مؤشر الكتابة الى السطر الاول**
**Lcd_Cmd(_LCD_SECOND_ROW);**
**ينقل مؤشر الكتابة الى السطر الثاني**

**Lcd_Cmd(_LCD_UNDERLINE_ON);**
**يظهر مؤشر الشاشة**

**Lcd_Cmd(_LCD_MOVE_CURSOR_LEFT);**
**يذهب بمؤشر الشاشة الى اخر حرف شمال**

Lcd_Cmd(_LCD_MOVE_CURSOR_RIGHT);
**يذهب بمؤشر الشاشة الى اخر حرف على اليمين**

**Lcd_Cmd(_LCD_TURN_ON);**

**تشغيل الشاشة**
**Lcd_Cmd(_LCD_TURN_OFF);**
**غلق الشاشة**

**الامر**
**Lcd_Chr_Cp('e');**
**طباعة حرف على الشاشة اخر الحروف المكتوبة**

**Lcd_Chr**

**طباعة حرف على الشاشة وله ثلاث معاملات**

**رقم السطر**

**رقم العمود**

**الحرف**

**مثال**

```
Lcd_Chr(2, 3, 'i');
```

**الامر**

```
Lcd_Out_Cp("Here!");
```

**كتابة نص عند نهاية اخر حرف مكتوب على الشاشة**

**الامر**

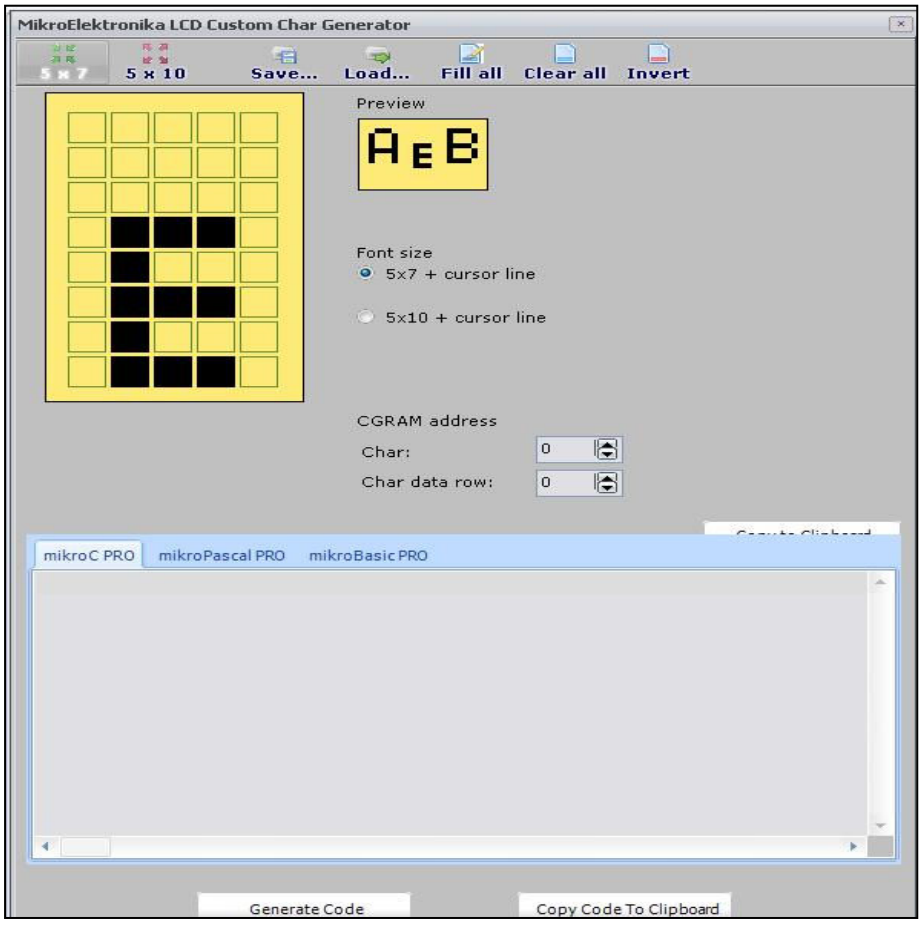
**Lcd_Out**

**كتابة نص على الشاشة وله ثلاث معاملات**
**رقم السطر**
**رقم العمود**
**النص**
**مثال**

```
Lcd_Out(1, 3, "Hello!");
```

**كما يمكن الرسم على الشاشة**
**من خلال قائمة tool كما في الشكل التالي**

**لتظهر النافذة التالية**



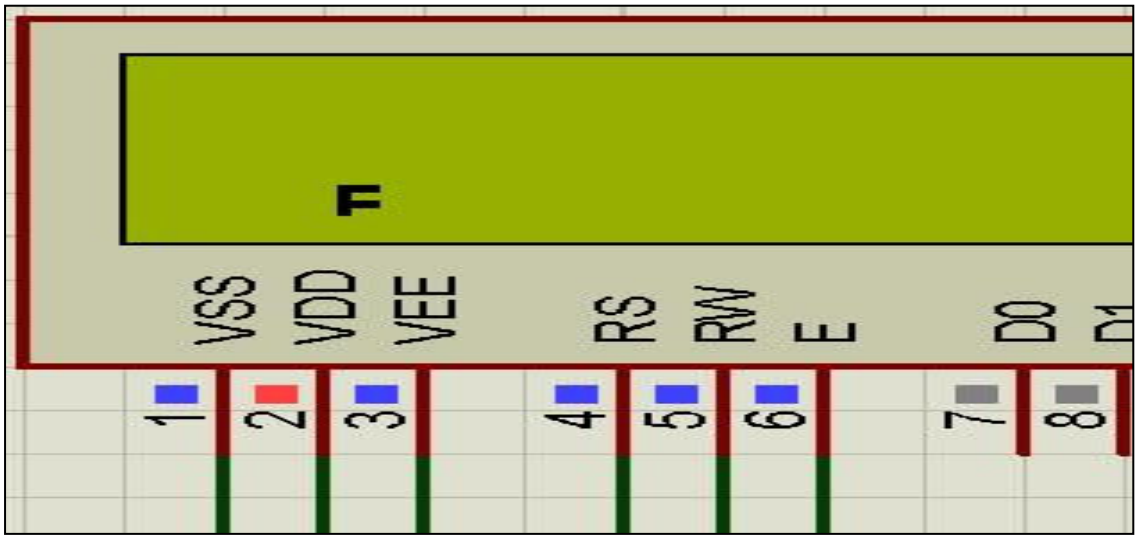
**ثم تحدد بالماوس الشكل الذي تريدة ثم اضغط زر توليد الكود**
**ثم زر النسخ**
**في قسم المتغيرات العامة**

```
// الرسم على الشاشة
const char character[] = {0,0,0,14,8,14,8,14};

void CustomChar(char pos_row, char pos_char) {
  char i;
    Lcd_Cmd(64);
    for (i = 0; i<=7; i++) Lcd_Chr_CP(character[i]);
    Lcd_Cmd(_LCD_RETURN_HOME);
    Lcd_Chr(pos_row, pos_char, 0);
}
```

**ثم اكتب**

```
CustomChar(2, 2);
```

لتظهر النتيجة كالتالي

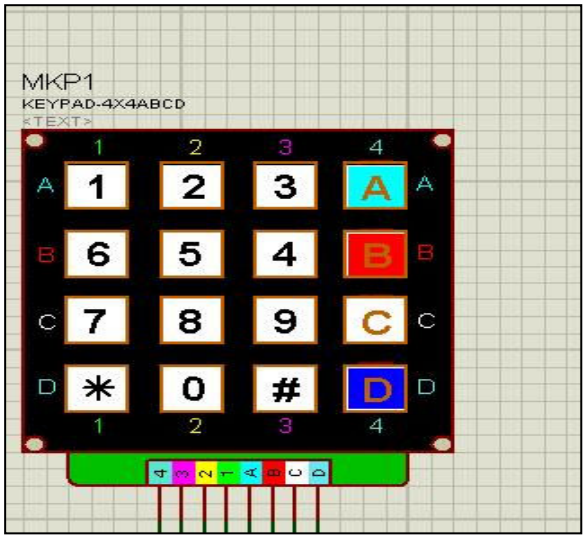



**وفي النهاية يارب اكون قد وفقت في شرح الشاشة LCD**

# بسم الله الرحمن الرحيم
# الفصل الخامس
# التعامل مع لوحة الارقام
# Keypad

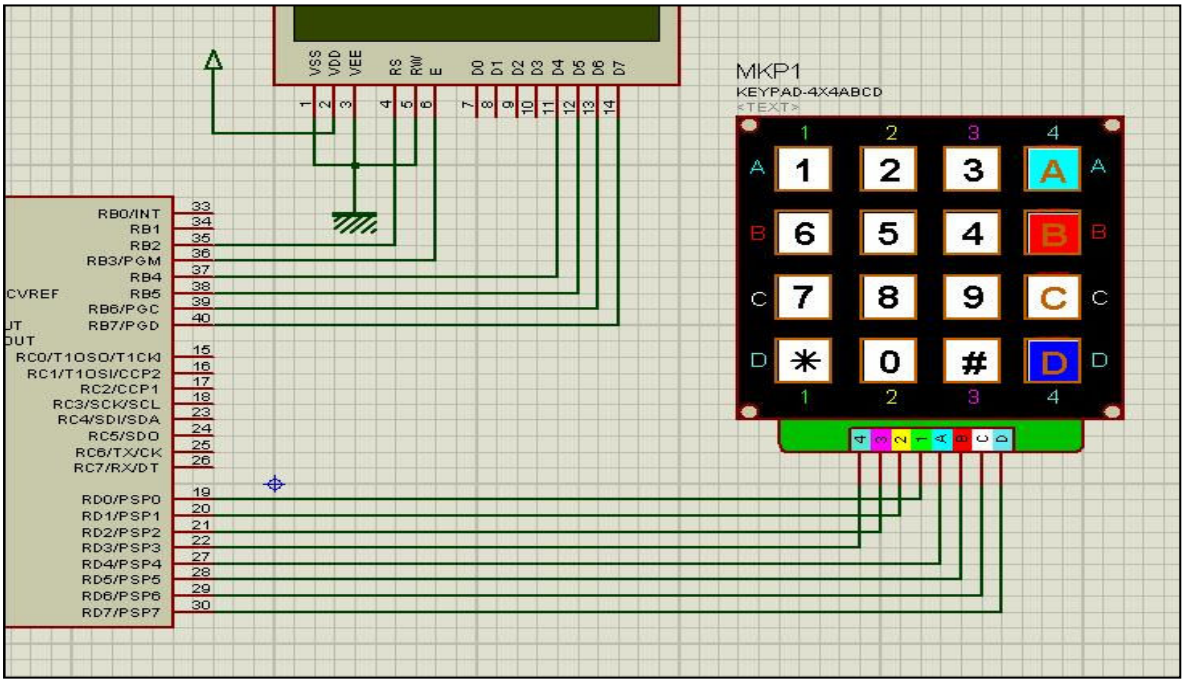
تأليف عيد فتحي

## اولا شكل الكيباد




## ويوجد منها نوعان 3x4 و 4x4
## وسوف نتعامل مع 4x4

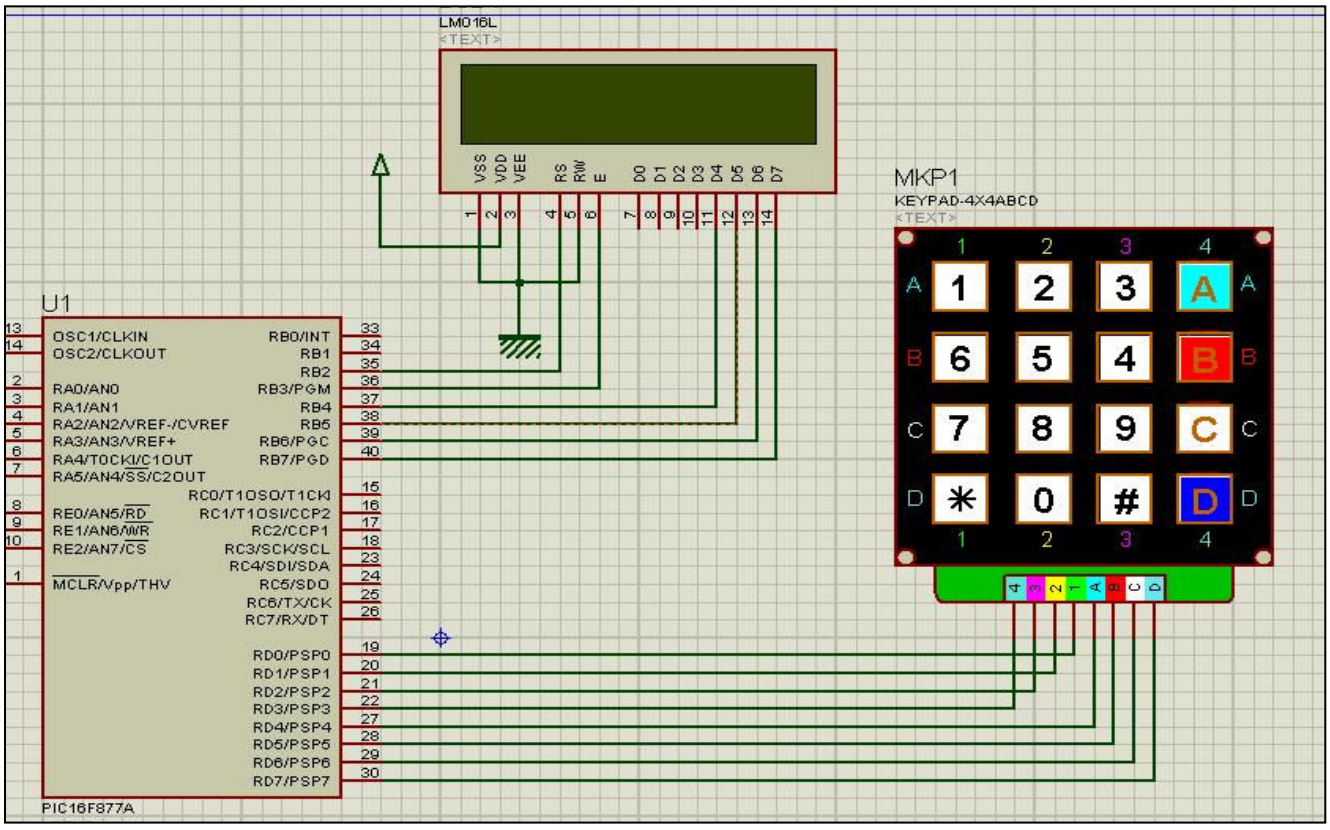
## ثانيا توصيل الاطراف



**حيث**
**الطرف ١ مع RD0**
**الطرف 2 مع RD1**
**الطرف 3 مع RD2**
**الطرف ٤ مع RD3**
**الطرف A مع RD4**
**الطرف B مع RD5**
**الطرف C مع RD6**
**الطرف D مع RD7**

## ثالثا شرح كود الكيباد على الميكروسي

**اختصارا للوقت سوف نقوم بعمل مشروع يظهر الارقام على شاشة بالضغط على الكيباد**
**اولا شكل المشروع على برنامج بروتس**



## ثانيا الكود

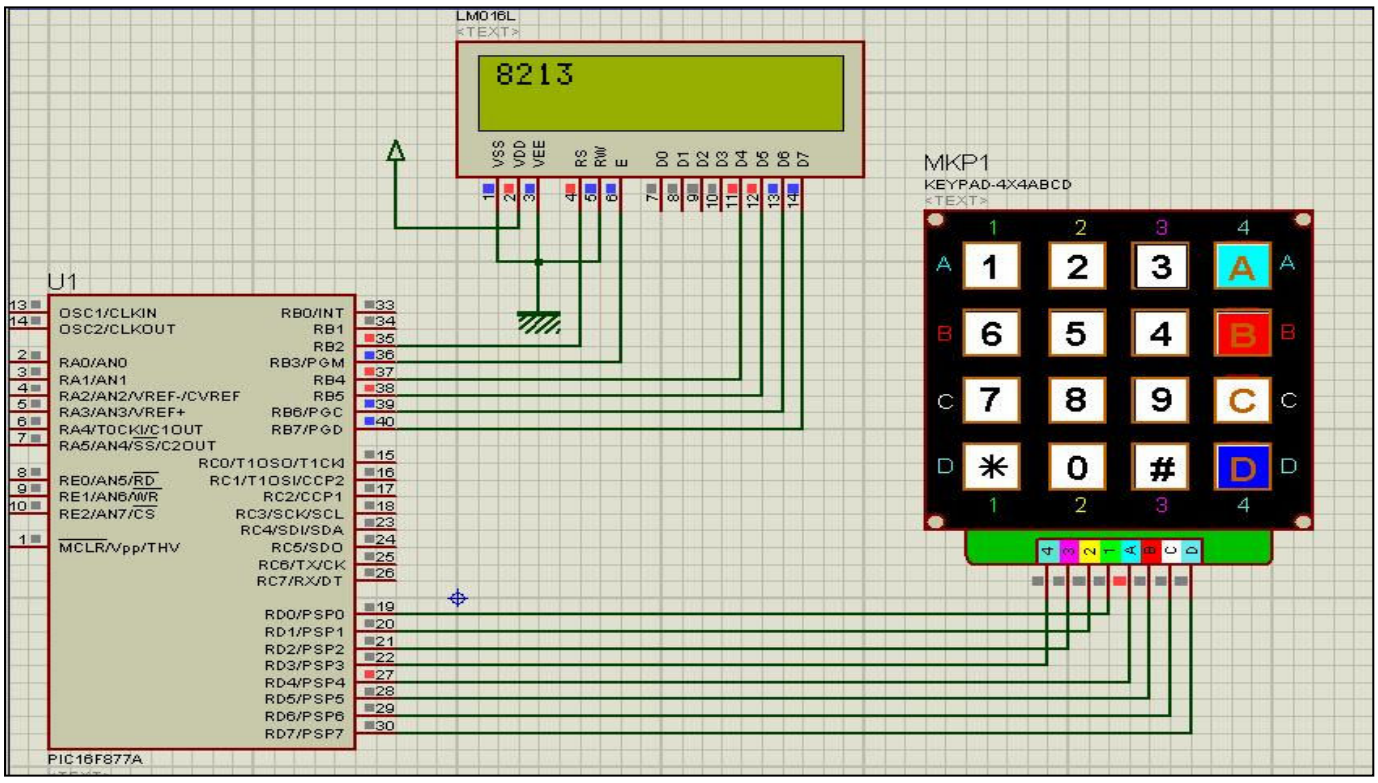
```
// اطراف الكيبباد
char keypadPort at PORTD;
unsigned short kp ;
// اطراف الشاشة
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;
```

**تم تعريف متغير من النوع short ليساوي مفاتيح الضغط**

```
void main() {
//  تعريف الكيبباد
 Keypad_Init();
 //  تعريف الشاشة
Lcd_Init();
//مسح الشاشة
Lcd_Cmd(_LCD_CLEAR);
// الغاء مؤشر الكتابة
Lcd_Cmd(_LCD_CURSOR_OFF);
```

```
    while (!kp){
      // الضغط على المفاتيح
      kp = Keypad_Key_Click();
      }
switch (kp) {


      case  1: kp = 49; break; // 1
      case  2: kp = 50; break; // 2
      case  3: kp = 51; break; // 3
      case  4: kp = 65; break; // A
      case  5: kp = 52; break; // 4
      case  6: kp = 53; break; // 5
      case  7: kp = 54; break; // 6
      case  8: kp = 66; break; // B
      case  9: kp = 55; break; // 7
      case 10: kp = 56; break; // 8
      case 11: kp = 57; break; // 9
      case 12: kp = 67; break; // C

      }

    // اظهار النتيجة على الشاشة
    Lcd_Chr_Cp(kp);
```

## لتكون النتيجة كالتالي



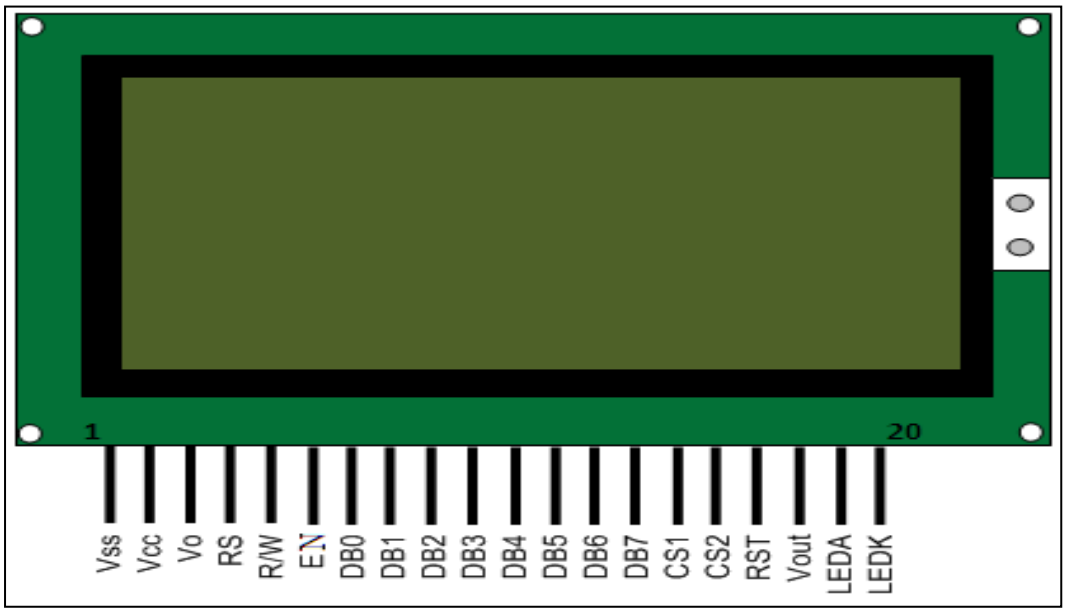
**ورغم عدم الشرح الكافي لهذه المكتبة الا اني اعتقد أنها من السهولة لدرجة لاتحتاج اليها شرح**

**والله الموفق**
**عيد فتحي**

# بسم الله الرحمن الرحيم
# الفصل الرابع
# الشاشة GLCD
# الشاشات الجرافيك

**تأليف : عيد فتحي**

## اولا شكل الشاشة ووظيفة كل طرف

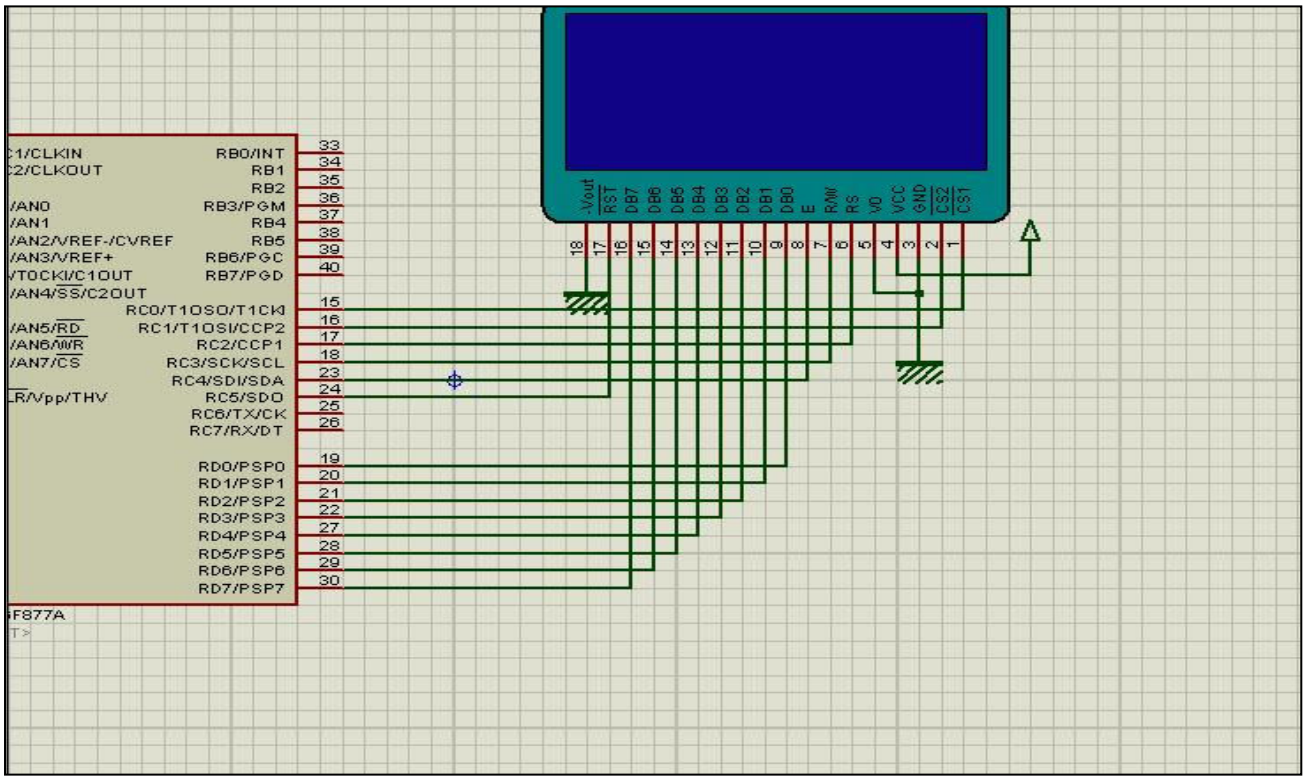



| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**١- Vss أرضي**
**٢- Vcc كهربي ٥ فولت**
**٣- Vo ضبط المراقبة**
**٤- RS تحديد اشارة السجل**
**٥-R/W قرائة كتابة الاشارة**
**٦- EN تفعيل بيانات القرائة والكتابة**
**٧- D0 خط البيانات**
**٨- D1 خط البيانات**
**٩- D2 خط البيانات**
**١٠- D3 خط البيانات**

١١- D4 خط البيانات
١٢- D5 خط البيانات
١٣- D6 خط البيانات
١٤- D7 خط البيانات
١٥- CS1 خط اختيار الشريحة
١٦- CS2 خط اخيار الشريحة
١٧- RST خط اعادة التشغيل
١٨- Vout خرج الجهد للشاشة
١٩- LEDA الخط الكهربي لاضائة الشاشة ٥فولت
٢٠- LEDK الخط الارضي لاضائة الشاشة

## ثانيا توصيل الشاشة




اولا
D0 حتى D7 مع RD0 حتى RD7
CS1 مع RC0
CS2 مع RC1
RS مع RC2
RW مع RC3

**EN مع RC4**
**RSTمع RC5**
**Vss مع الارضي**
**Vcc كهربي ٥ فولت**
**Vo مع الارضي**
**Vout مع الارضي**
**LEDA كهربي ٥ فولت**
**LEDK ارضي**

**ملحوظة برنامج بروتس لا يوجد به اطراف اضائة الشاشة**

# ثالثا شرح مكتبة GLCD

```
char GLCD_DataPort at PORTD;
sbit GLCD_CS1 at RC0_bit;
sbit GLCD_CS2 at RC1_bit;
sbit GLCD_RS  at RC2_bit;
sbit GLCD_RW  at RC3_bit;
sbit GLCD_EN  at RC4_bit;
sbit GLCD_RST at RC5_bit;

sbit GLCD_CS1_Direction at TRISC0_bit;
sbit GLCD_CS2_Direction at TRISC1_bit;
sbit GLCD_RS_Direction  at TRISC2_bit;
sbit GLCD_RW_Direction  at TRISC3_bit;
sbit GLCD_EN_Direction  at TRISC4_bit;
sbit GLCD_RST_Direction at TRISC5_bit;
// توصيل الاطراف
```

**هذا الكود خاص بتعريف أطراف الشاشة الموصلة مع الميكروكنترولر ويكتب قبل الدالة الرئيسية**

```
Glcd_Init();
```
**تعريف المكتبة**

```
Glcd_Fill(0);
```
**يقوم بمسح الشاشة او ملئ الشاشة بلون محدد**
**وله معامل واحد اللون وصفر يعني ابيض**

```
Glcd_Dot(0, 0, 2);
```
**رسم نقطة وله ثلاث معاملات**
**الاول تحديد النقطة على المحور الافقي ويأخذ القيمة من ٠ حتى ١٢٧**
**الثاني تحديد النقطة على المحور الرأسي ويأخذ القيمة من ٠ حتى ٦٣**
**الثالث تحديد اللون**

```
Glcd_Line(0, 0, 20, 30, 1);
```
**رسم خط ولها خمس معاملات**
**الاول بداية الخط على المحور الافقي ويأخذ القيمة من ٠ حتى ١٢٧**
**الثاني بداية الخط على المحور الرأسي ويأخذ القيمة من ٠ حتى ٦٣**
**الثالث نهاية الخط على المحور الافقي**
**الرابع نهاية الخط على المحور الرأسي**
**الخامس اللون**

```
Glcd_V_Line(5, 25, 10, 1);
```
**رسم خط عمودي وله اربع معاملات**
**الاول بداية الخط على المحور الرأسي**
**الثاني تحديد المحور الرأسي**
**الثالث نهاية الخط على المحور الرأسي**
**الرابع تحديد اللون**

```
Glcd_H_Line(10, 50, 20, 1);
```
**رسم خط افقي وله اربع معاملات**
**الاول بداية الخط على المحور الافقي**
**الثاني تحديد المحور الرأسي**
**الثالث نهاية الخط على المحور الافقي**
**الرابع تحديد اللون**

```
Glcd_Rectangle(5, 5, 40, 40, 1);
```
**رسم مستطيل وله خمس معاملات**
**الاول بداية المستطيل على المحور الافقي من الجهه اليسرى**
**الثاني بداية المستطيل على المحور الرأسي من الجهة اليسرى**
**الثالث نهاية المستطيل على المحور الافقي من الجهة اليمنى**
**الرابع نهاية المستطيل على المحور الرأسي من الجهة اليمنى**
**الخامس تحديد اللون**

```
Glcd_Rectangle_Round_Edges(5, 5,
40, 40, 12, 1);
```
**رسم مستطيل دائري الحواف وله ستة معاملات**
**الاول بداية المستطيل على المحور الافقي من الجهه اليسرى**
**الثاني بداية المستطيل على المحور الرأسي من الجهة اليسرى**
**الثالث نهاية المستطيل على المحور الافقي من الجهة اليمنى**
**الرابع نهاية المستطيل على المحور الرأسي من الجهة اليمنى**
**المعامل الخامس نصف قطر دائرة الحافة**
**المعامل السادس تحديد اللون**

```
Glcd_Rectangle_Round_Edges_Fill(5,
5, 40, 40, 12, 1);
```
**رسم مستطيل دائري الحواف ملون وله ستة معاملات هي نفس السابقة**

```
Glcd_Circle(50, 50, 10, 1);
```
**رسم دائرة ولها اربع معاملات**
**الاول مركز الدائرة على المحور الافقي**
**الثاني مركز الدائرة على المحور الرأسي**
**الثالث نصف قطر الدائرة**
**الرابع لون الدائرة**

```
Glcd_Circle_Fill(50, 50, 10, 1);
```
**رسم دائرة ملونة لها اربع معاملات هي نفس السابقة**

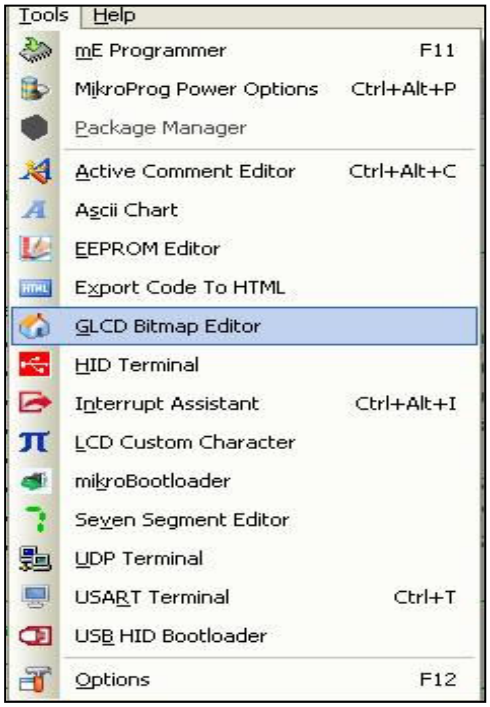
```
Glcd_Write_Char('C', 10, 2, 1);
```

**كتابة حرف ولها اربع معاملات**
**الاول الحرف**
**الثاني المحور الافقي**
**الثالث المحور الرأسي**
**الرابع لون الحرف**

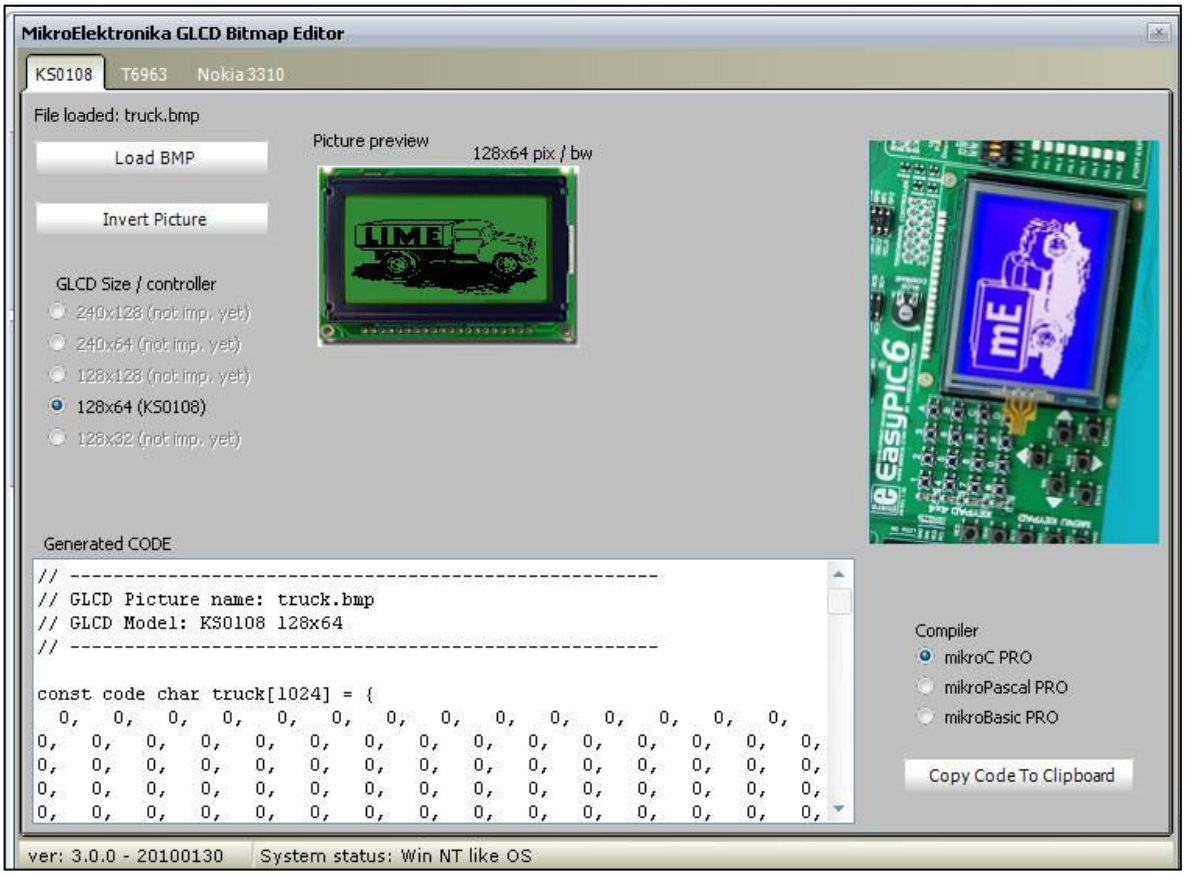
```
Glcd_Write_Text("Hello world!",
10, 2, 1);
```

**كتابة نص ولها اربع معاملات**
**الاول النص**
**الثاني بداية النص على المحور الافقي**
**الثالث بداية النص على المحور الرأسي**
**الرابع لون النص**

**رسم صورة**
**من قائمة Tools**
**اختر**



**لتظهر النافذة التالية**



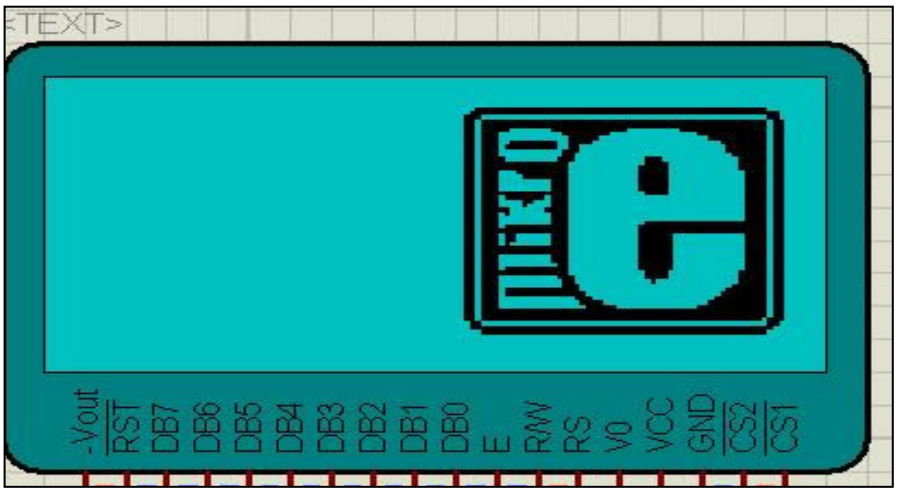

**من زر load BMP حدد الصورة ثم انسخ الكود من زر copy code**
**ثم ضع الكود في قسم المتغيرات العامة**
**كالتالي**

```
// GLCD Picture name: bank.bmp
// GLCD Model: KS0108 128x64
// -------------------------------------------------

const code char bank[1024] = {
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  0,   0,   0,   0, 128, 128,   0,  64, 128,   0,  96, 128,  48,  96,  24,  24,  32,  12,  20,  18,   6,  12,
  0,   0,   0,   1,   1,   1, 129,   1,  63,  33,  33,  33,  33, 123,  63,  37,  37,  37,  63, 187,  33,  33,
  0,   0,   0,   0,   0,   0,   0,   0, 255,   0,   0,   0,   0, 255,  86,   0,   0,   0,   0, 255,   0,   0,
  0,   0,   0,   0,   0,   0,  64,  64, 127,  64,  64,  64,  64, 127, 107,  64,  64,  64,  64, 127,  64,  64,
  0,   0,   0,   0,   0,   0,   6,   6,   6,   6,   6,   6,   6,   6,   6,   6,   6,   6,   6,   6,   6,   6,
};
// -------------------------------------------------
```

**ثم أكتب**

**Glcd_Image(bank); كلمة bank هي اسم الصورة**

**لتظهر النتيجة**
**كالتالي**




```
Glcd_Ext_Image(153608);
```

**يعرض صورة خارجية**

```
Glcd_PartialImage(10, 12, 10, 15,
16, 32, image);
```

**يعرض جزء من الصورة في المنطقة المحددة وله سبع معاملات**
**الاول تحديد بداية الصورة على المحور الافقي**
**الثاني تحديد بداية الصورة على المحور الرأسي**
**الثالث المطلوب من عرض الصورة**
**الرابع المطلوب من طول الصورة**
**الخامس عرض الصورة الاصلية**
**السادس طول الصورة الاصلية**
**السابع اسم الصورة**

```
Glcd_Ext_PartialImage(10, 12, 10,
15, 16, 32, 0);
```
**يعرض جزء من صورة خارجية في المنطة المحددة وله سبع معاملات نفس السابقة**

```
Glcd_Write_Text_Adv("GLCD", 0, 0);
```
يكتب النص على الاحداثيات الافقي والرأسي
وله ثلاث معاملات
١- النص
٢- المحور الافقي
٣- المحور الرأسي

```
Glcd_Set_Font_Adv(&myfont, 0, 0);
```
**يحدد نوع الخط الذي يكتب على الشاشة**
**وله ثلاث معاملات**
**الاول نوع الخط**
**الثاني لون الخط**
**الثالث تحديد اتجاه الخط**

```
Glcd_Set_Ext_Font_Adv(173296, 5,
7, 32);
```
**يحدد نوع الخط الذي يكتب على الشاشة ولكن خط خارجي**
**وله ثلاث معاملات**
**عنوان الخط المستخدم على جهاز الكمبيوتر**
**الثاني لون الخط**
**الثالث تحديد اتجاه الخط**

```
Glcd_Set_Font(&myfont, 5, 7, 32);
```
يحدد نوع الخط الذي يكتب على الشاشة
وله ثلاث معاملات
الاول اسم الخط

الثاني لون الخط
الثالث تحديد اتجاه الخط

يارب أكون وفقت في شرح مكتبة GLCD
او الشاشات الجرافيك

عيد فتحي

بسم الله الرحمن الرحيم

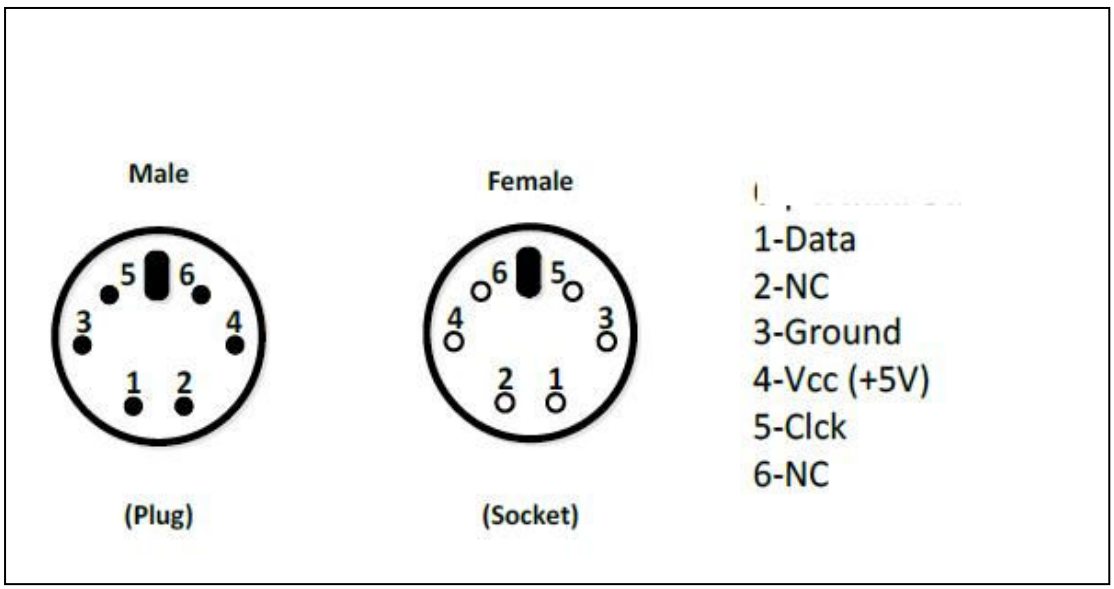
# مكتبة لوحة المفاتيح Keyboard

تأليف عيد فتحي

## مكتبة Ps2

هي مكتبة قرائة الميكروكنترولر للوحة المفاتيح

## Ps2 يتكون من ٦ بنات او اطراف كما في الصورة التالية



**وظائف الاطراف**

**١- بيانات DATA**
**٢- غير مستعمل NC**
**٣- ارضي GROUND**
**٤- كهربي ٥ فولت Vcc**
**٥- تردد النبضة CLCK**
**٦- غير مستعمل NC**

**ويتم توصيل طرفين فقط مع الميكروكنترولر**
**الطرف الاول Data والطرف الثاني CLCK والطرفين غير محديين بأطراف محددة**

**معاملات المكتبة**

**١- Ps2_Config**
**وظيفته تعريف Ps2**

**٢- Ps2_Key_Read**
**قرائة المفتاح المضغوط علية من لوحة المفاتيح**

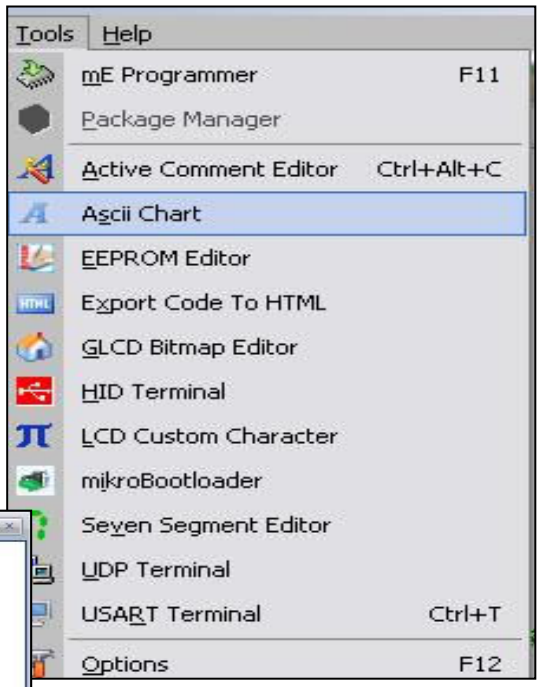
**وله ثلاث معاملات**
**أ- القيمة :- قيمة المفتاح المضغوط عليه ويتم كتابته بحروف أسكي ASCII**
**ب- خاص :- هل هو مفتاح خاص مثل أنتر ام حرف عادي مثل الارقام والاحرف الهجائية**
**اذا كان حرف خاص يأخذ القيمة ١**
**اذا كان حرف عادي يأخذ القيمة ٠**

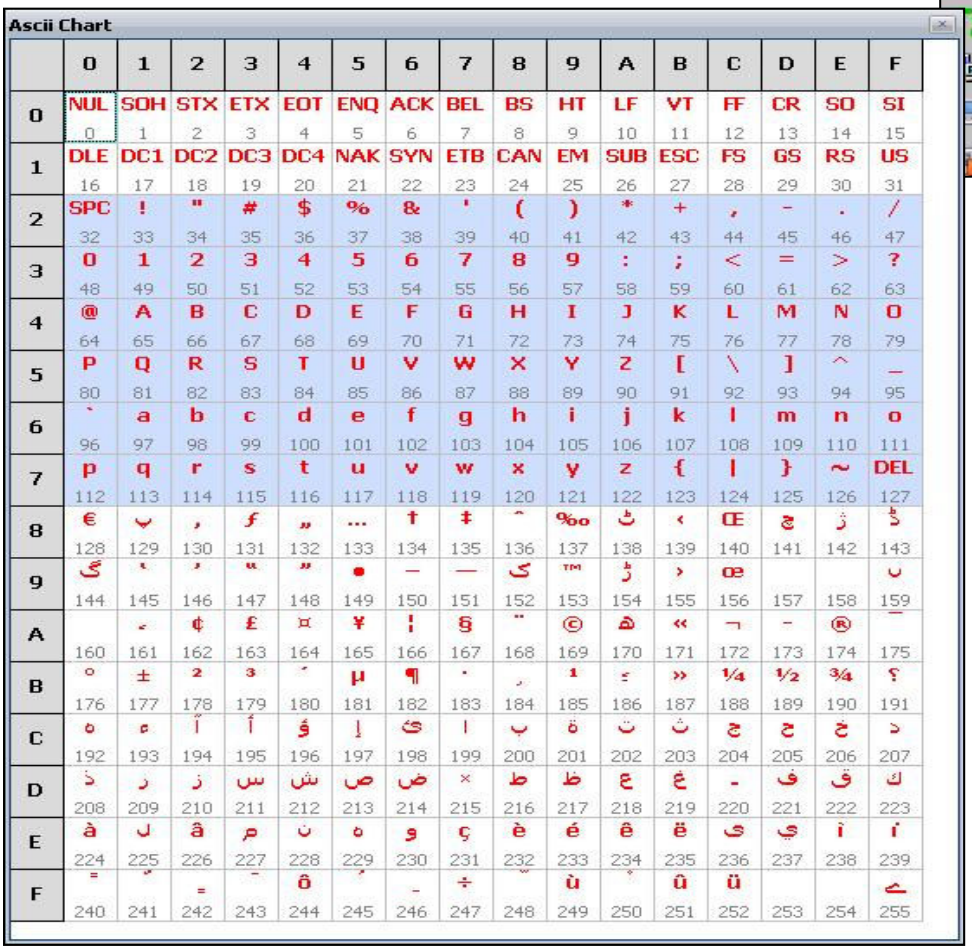
ج – طريقة ارجاع قيمة الضغط على المفاتيح هل يتم الضغط ام تم تحرير المفتاح (أي رفع يدك من على المفتاح )
ويأخذ القيم التالية

- ٠ تم تحرير مفتاح الضغط
- ١ يتم الضغط حاليا

ويفضل اختيار ٠ حتى لايتم كتابة حروف متكررة

ولمعرفة قيمة حروف أسكي أضغط من القائمة TOOLS




| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NUL 0 | SOH 1 | STX 2 | ETX 3 | EOT 4 | ENQ 5 | ACK 6 | BEL 7 | BS 8 | HT 9 | LF 10 | VT 11 | FF 12 | CR 13 | SO 14 | SI 15 |
| 1 | DLE 16 | DC1 17 | DC2 18 | DC3 19 | DC4 20 | NAK 21 | SYN 22 | ETB 23 | CAN 24 | EM 25 | SUB 26 | ESC 27 | FS 28 | GS 29 | RS 30 | US 31 |
| 2 | SPC 32 | ! 33 | " 34 | # 35 | $ 36 | % 37 | & 38 | ' 39 | ( 40 | ) 41 | * 42 | + 43 | , 44 | - 45 | . 46 | / 47 |
| 3 | 0 48 | 1 49 | 2 50 | 3 51 | 4 52 | 5 53 | 6 54 | 7 55 | 8 56 | 9 57 | : 58 | ; 59 | < 60 | = 61 | > 62 | ? 63 |
| 4 | @ 64 | A 65 | B 66 | C 67 | D 68 | E 69 | F 70 | G 71 | H 72 | I 73 | J 74 | K 75 | L 76 | M 77 | N 78 | O 79 |
| 5 | P 80 | Q 81 | R 82 | S 83 | T 84 | U 85 | V 86 | W 87 | X 88 | Y 89 | Z 90 | [ 91 | \ 92 | ] 93 | ^ 94 | _ 95 |
| 6 | ` 96 | a 97 | b 98 | c 99 | d 100 | e 101 | f 102 | g 103 | h 104 | i 105 | j 106 | k 107 | l 108 | m 109 | n 110 | o 111 |
| 7 | p 112 | q 113 | r 114 | s 115 | t 116 | u 117 | v 118 | w 119 | x 120 | y 121 | z 122 | { 123 | \| 124 | } 125 | ~ 126 | DEL 127 |
| 8 | € 128 | پ 129 | ‚ 130 | ƒ 131 | „ 132 | … 133 | † 134 | ‡ 135 | ˆ 136 | ‰ 137 | ٹ 138 | ‹ 139 | Œ 140 | چ 141 | ژ 142 | ڈ 143 |
| 9 | گ 144 | ‘ 145 | ’ 146 | “ 147 | ” 148 | • 149 | – 150 | — 151 | ک 152 | ™ 153 | ڑ 154 | › 155 | œ 156 | 157 | 158 | ں 159 |
| A | 160 | ء 161 | ¢ 162 | £ 163 | ¤ 164 | ¥ 165 | ¦ 166 | § 167 | ¨ 168 | © 169 | ھ 170 | « 171 | ¬ 172 | - 173 | ® 174 | ¯ 175 |
| B | ° 176 | ± 177 | ² 178 | ³ 179 | ´ 180 | µ 181 | ¶ 182 | · 183 | ¸ 184 | ¹ 185 | ے 186 | » 187 | ¼ 188 | ½ 189 | ¾ 190 | ؟ 191 |
| C | ہ 192 | ۂ 193 | آ 194 | أ 195 | ؤ 196 | إ 197 | ئ 198 | ا 199 | ب 200 | ة 201 | ت 202 | ث 203 | ج 204 | ح 205 | خ 206 | د 207 |
| D | ذ 208 | ر 209 | ز 210 | س 211 | ش 212 | ص 213 | ض 214 | × 215 | ط 216 | ظ 217 | ع 218 | غ 219 | ـ 220 | ف 221 | ق 222 | ك 223 |
| E | à 224 | ل 225 | â 226 | م 227 | ن 228 | ه 229 | و 230 | ç 231 | è 232 | é 233 | ê 234 | ë 235 | ى 236 | ي 237 | î 238 | ï 239 |
| F | ّ 240 | ٌ 241 | ٍ 242 | ً 243 | ô 244 | ُ 245 | ِ 246 | ÷ 247 | ّ 248 | ù 249 | ْ 250 | û 251 | ü 252 | 253 | 254 | ے 255 |

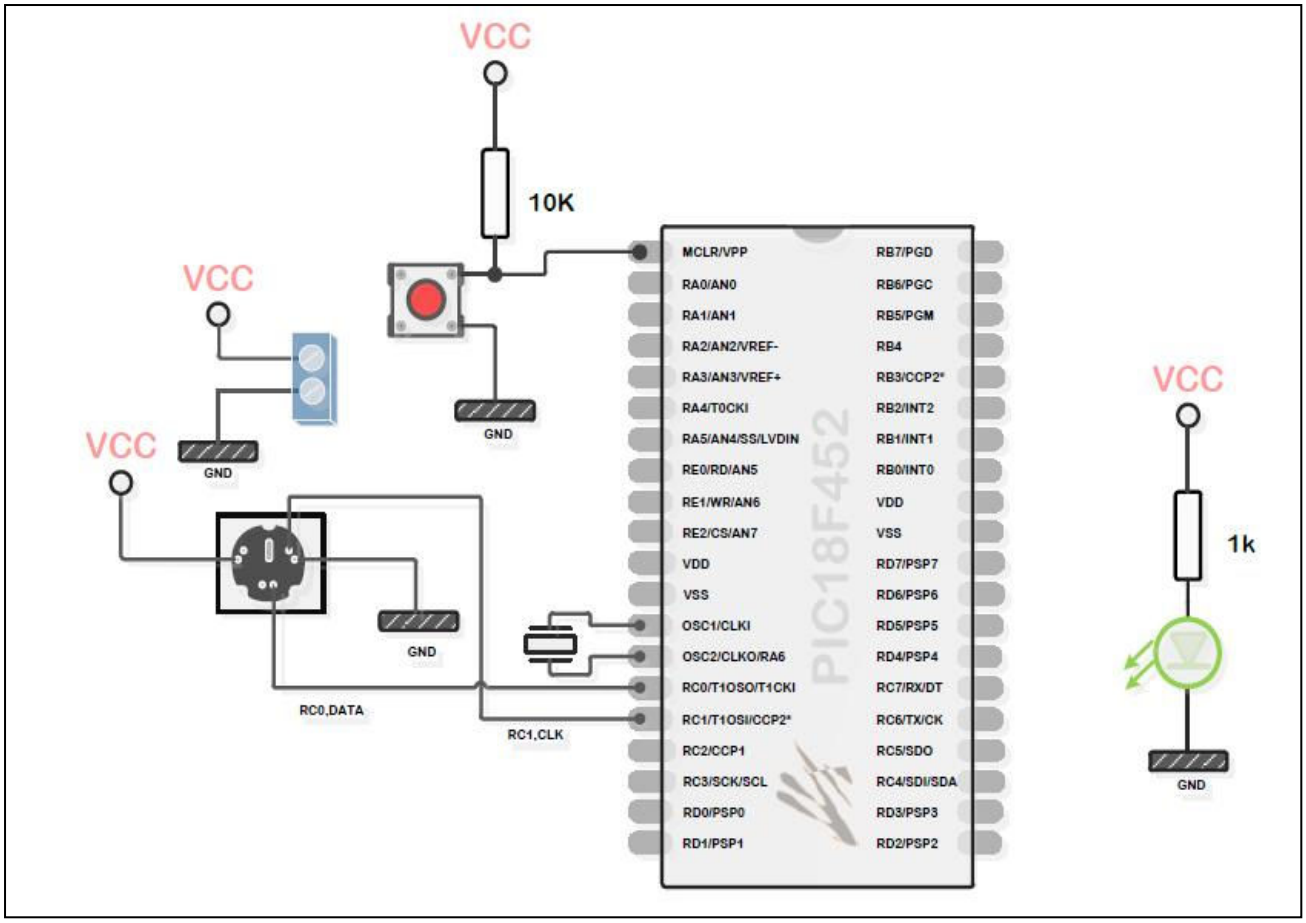
**مثال**
**في هذا المثال سوف يتم كتابة حرف E بعد الضغط على لوحة المفاتيح على شاشة**

**اولا الكود**

```
// مكتبة Ps2
sbit PS2_Data                at RC0_bit;
sbit PS2_Clock               at RC1_bit;
sbit PS2_Data_Direction      at TRISC0_bit;
sbit PS2_Clock_Direction     at TRISC1_bit;


// LCD
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;
```

```
void main() {
 Lcd_Init();// تعريف الشاشة
Lcd_Cmd(_LCD_CURSOR_OFF);  // الغاء مؤشر الكتابة

Ps2_Config(); //تعريف الكاي بورد


while(1){
if (Ps2_Key_Read(67, 0, 0)) {    // قرائة الحرف بعد رفع اليد من على الحرف
 //  حـــــرف E
 Lcd_Out_Cp("e");
       }


}


}
```

## ثانيا شكل الدائر




**ملحوظة يتم توصيل طرف Data وطرف CLCK**
**بمصدر جهد ٥ فولت ومقاومة ١كيلو اوم**

**والله الموفق**
**والية المصير**
***عيد فتحى***

بسم الله الرحمن الرحيم

# المتغيرات والدوال الاساسية
# في الميكروسي MikroC
## تأليف عيد فتحي

**اولا انواع المتغيرات**

**١- نصي char**
**٢- رقم قصير short**
**٣- رقم صحيح int**
**٤- رقم طويل long**
**٥- عدد كسري float**

**واذا اضيف عبارة Unsigned يعبر عن القيم الموجبة فقط**

**واليك الجدول التالي**
**وبه قيم كل متغير**

| *Tipo de variable* | *Tamaño en Bytes* | *Valores que soporta* |
|---|---|---|
| *bit* | *1* | *0 ó 1* |
| *char* | *1* | *-127 a 127* |
| *short* | *1* | *-127 a 127* |
| *int* | *2* | *- 32767 a 32767* |
| *long* | *4* | *- 2147483647 a 2147483647* |
| *float* | *4* | *-1.5x10^45 a 3.4x10^38* |
| *double* | *4* | *-1.5x10^45 a 3.4x10^38* |
| *unsigned char* | *1* | *0 a 255* |
| *unsigned short* | *1* | *0 a 255* |
| *unsigned int* | *2* | *0 a 65535* |
| *unsigned long* | *4* | *0 a 4294967295* |

**ثانيا اهم دوال الميكروسي**

**الدالة SizeOf**
**تعيد حجم المتغير بالبايت  وتستعمل مع المتغيرات النصية والارقام الصحيحة والاعداد الطويلة**
**مثال**

```
Char x="7";
SizeOf(x);
```

**الدالة abs**
**تعيد القيمة الموجبة اذا وجدت لرقم**
**مثال**

```
Result=abs(-12);//12
```

**الدالة Max**
**تعيد اكبر قيمة من قيمتين**
**مثال**

```
Result=Max(123,73);//123
```

**الدالة Min**
**تعيد القيمة الصغرى بين قمتين**
**مثال**

```
Result=Min(123,73);//73
```

**الدالة ()rend**
**تعيد قيمة عشوائية من ٠ الى ٣٢٧٦٧**

**مثال**

```
Result=rand();
```

**تكمن أهمية الدوال التي تحول الارقام الى سلسلة نصية في قرائة الشاشات لها بطريقة صحيحة**

**الدالةٍ ShotToStr**

**تحول الرقم القصير الى سلسلة نصية**

**مثال**

```
Short t=22;
Char txt[4];
ShortToStr(t,txt);
```

**الدالة IntTOStr**

**تحول الرقم الصيح الى سلسلة نصية**

```
IntToStr(t,txt);
```

**الدالة FloatToStr**

**تحول الرقم العشري الى سلسلة نصية**

```
FloatToStr(t,txt);
```

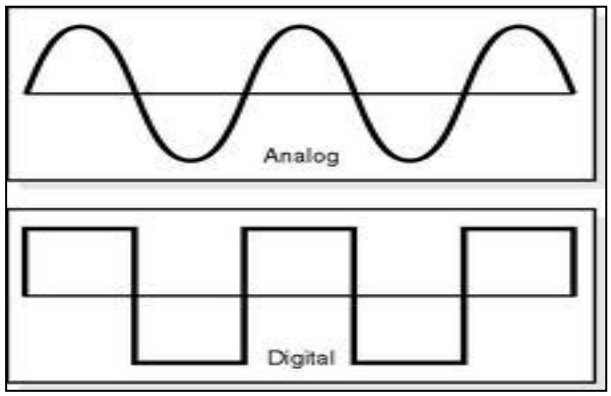
**هذا والله الموفق واليه المصير**

**عيد فتحي**

# بسم الله الرحمن الرحيم
# الفصل السادس
# المحول التماثلي الرقمي

تأليف عيد فتحي

اولا وظيفة المحول ADC
هي تحويل الاشارة الانالوج الي ديجتال حتى يستطيع الميكروكنترولر التعامل معها
ومن المعروف ان الاشارة الانالوج غير منتظمة ولها قيم متعددة بينما الاشارة الديجتال تأخذ قيمتين اما صفر او واحد



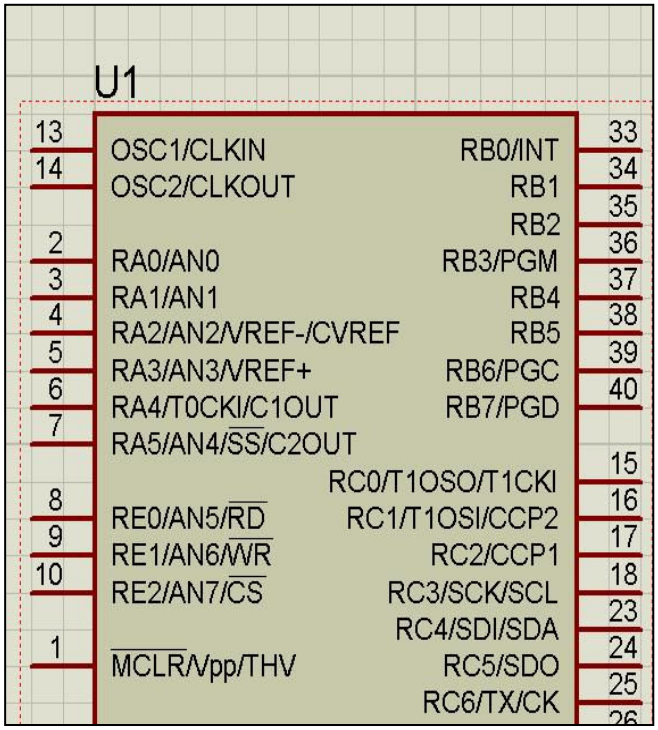

ويكمن أهمية المحول ADC في قرائة قيم السنسور مثل سنسور الضغط وسنسور درجة الحرارة وسنسور شدة الضوء

والسنسور هو اداة تحول القيم الفيزيائية الى قيمة كهربائية مثال على ذلك سنسور او حساس الوزن حيث أنه ظاهرة فيزيائية تدعى بيزو او الكهرباء الانضغاطية وهي قدرة بعض المواد على توليد كهرباء عند الضغط

عليها مثل الكوارتز ويمكن أن ترى ظاهرة بيزو في الولاعات
ومثال اخر للسنسور سنسور درجة الحرارة حيث أن التقاء معدنين مختلفين مثل الحديد والنحاس في نقطة وتعرض هذه النقطة للحرارة يتولد كهرباء
ومثال اخر المقاومة الضوئية حيث تقل مقاومة بعض المواد عند التعرض للضوء

ثانيا اطراف المحول في الميكروكنترولر

يكون في البورت **A** و **E** ويكتب بجواره **AN** اي انالوج




وعلى ذلك يوجد في **PIC16F877A** ٨ اطراف انالوج وهي **RA0** و **RA1** و **RA2** و **RA3** و **RA5** و **RE0** و **RE1** و **RE2**

## ثالثا مكتبة ADC

```
ADC_Init();
```

هذا الكود لتعريف المكتبة

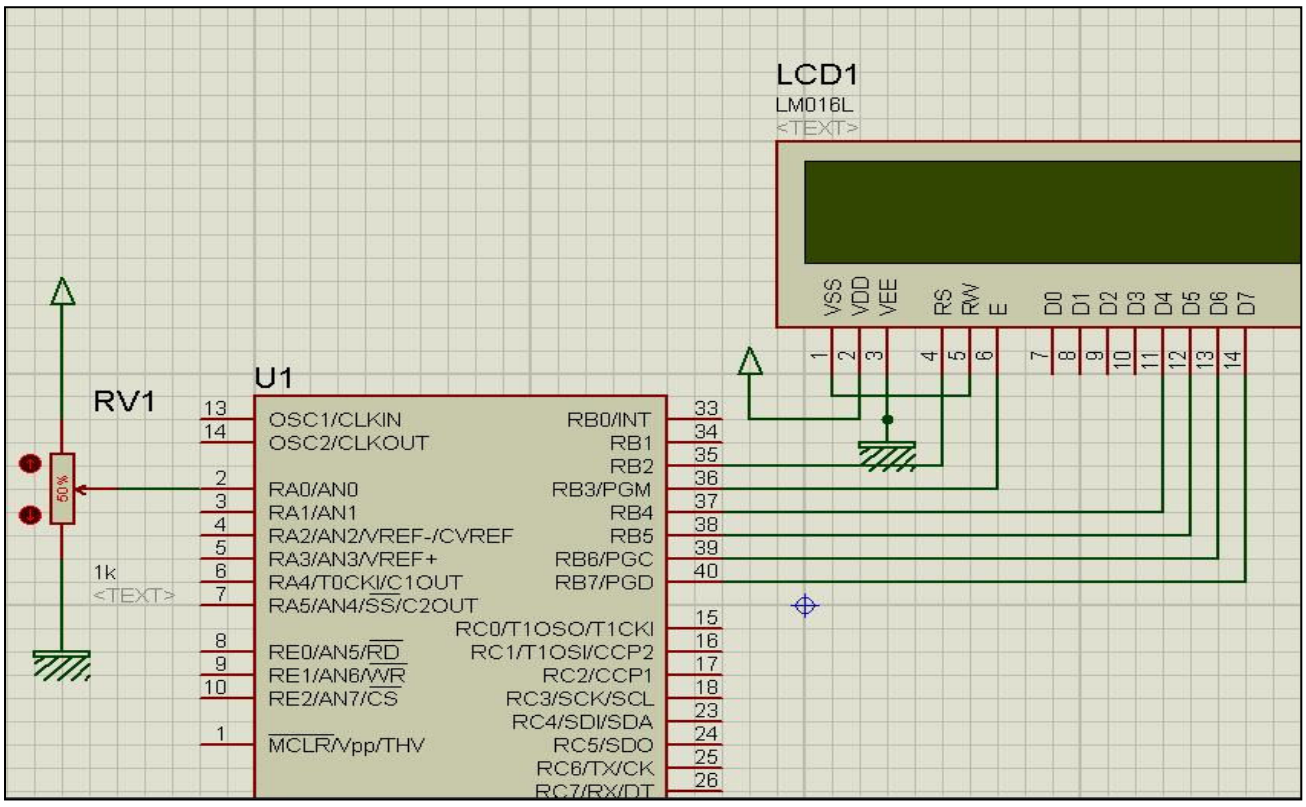
```
tmp = ADC_Read(2);
```

هذا الكود هو لقرائة القيم المدخلة وله معامل واحد هو رقم الطرف AN

## رابعا مثال على المكتبة ADC

وسوف نقوم بعمل افوميتر لقياس قيمة المقاومة

شكل المشروع



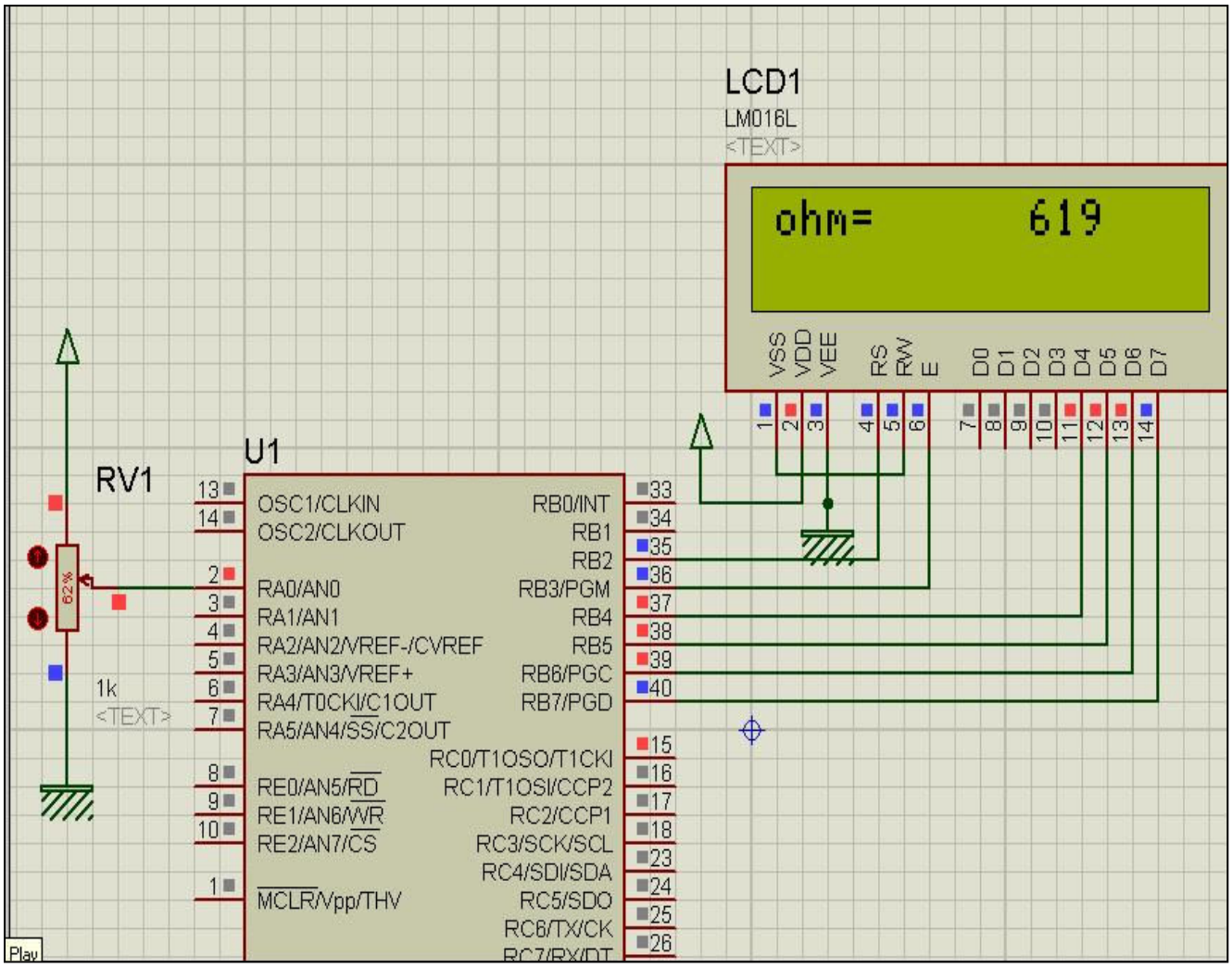
تم اضافة مقاومة متغيرة POT HG قيمتها ١ كيلو اوم وتوصيلها الى الطرف AN0 كما يظهر في الشكل

الكود

```
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;


// اطراف الشاشة



int ohm;     char txt[7];
//
```

```
void main(){
trisc.B0=0;
// تعريف الشاشة والمحور الرقمي
ADC_Init();          Lcd_Init();
//  الغاء مؤشر الكتابة
Lcd_Cmd(_LCD_CURSOR_OFF);

 // كتابة كلمة اوم بجوار الرقم
Lcd_Out(1,1,"ohm=");
 while(1){
 // قراءة قيمة المقاومة
 ohm = ADC_Read(0);

 ohm=(ohm*5.0*1000.0)/(1023.0*5.0);

 IntToStr(ohm, txt);
 Lcd_Out(1,8,txt);

 }
}
```

لتظهر النتيجة كالتالي




هذا والله الموفق

عيد فتحي

**بسم الله الرحمن الرحيم**

**طريقة حساب معادلة أي مجس او سنسور**

**Vt مجموع الفولت**
**Vomax اقصي قيمة للفولت**
**Vomin اقل قيمة للفولت**
**Vt=Vomax-Vomin**

**St مجموع وحدات السنسور**
**Smax القيمة العظمى للسنسور**
**Smin القيمة الصغرى للسنسور**
**St=Smax-Smin**

**K الثابت**
**K=Vt/St**

**Out المعادلة النهائية**
**Vonow الخرج الحالي للفولت**
**Out=(Vonow-Vomin)/K+Smin**

بسم الله الرحمن الرحيم

# المقاطعات
# Interrupts

مقدمة

تعتبر المقاطعات من المواضيع الشائكة في الميكروسي وان شاء الله سوف اقوم بمحاولة شرحها بسهولة ويسر حتي نرتقى بالمهندس العربي هذا والله الموفق

مؤلف الكتاب :: عيد فتحي

## اولا تعريف المقاطعة

عندما تمر بشخص تعرفه وهويتكلم مع أحد الاشخاص ثم تلقى عليه السلام فسوف يتوقف ليرد عليك السلام ثم يكمل حديثه

كذلك المقاطعة هي برنامج فرعي يطلب من المعالج البروسيسور التوقف لشرط ما مثل مرور فتره زمنيه او تغير الجهد على طرف ما

ولكن لاحظ الاتي أن المقاطعة ليست كأي برنامج فرعي بل هو برنامج له اسم محدد لا نستطيع تغييره فهو يبدأ قبل البرنامج الرئيسي بعبارة

```
Void Interrupt (){
هنا يكتب البرنامج
}
```

## أهمية المقاطعات

لاحظ مع الاتي أن الميكروكنترولر يقوم بتنفيذ الامر بالتوالي يعني الامر واحد ثم الذي يليه وعند استخدام عبارة مثل

**Delay_ms**

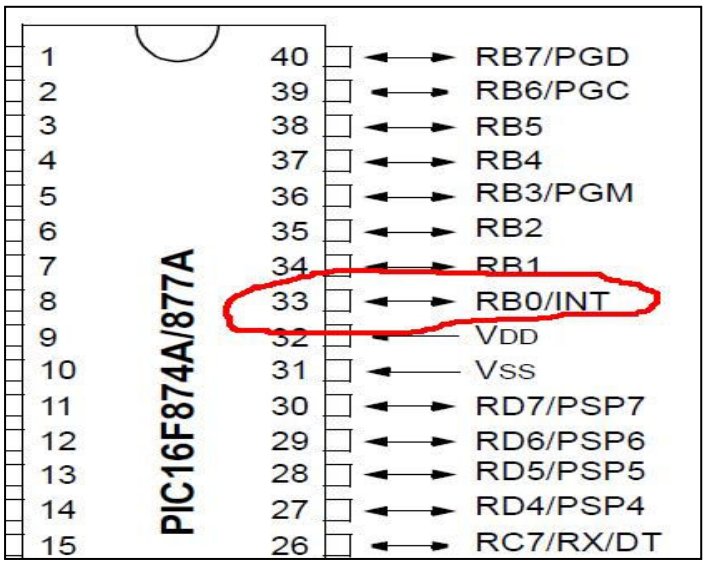
فإنه الميكرو يقوم بتكرار الامر أكثر من مليون مرة في الثانية الواحده مما يؤدي الى سرعة تلف الميكروكنترولر ثانيا استهلاك أكثر للطاقة وتوقف الميكروكنترولر لاستخدام اوامر البرنامج

بينما عند استخدام المقاطعات سوف نحافظ على الميكروكنترولر من التلف واستهلاك اقل للطاقة الكهربية ثالثا لن يتوقف سير البرنامج

## انواع المقاطعات

١- خارجية : مثل تغير الجهد عند الطرف

## RB0 INT




– داخلية مثل تغير احد سجلات الميكروكنترولر مثل سجل طفحان التايمر زيروTMR0

وتعتمد المقاطعات على سجلين فقط وهما

**INTCON**
**OPTION_REG**

وكل سجل يحتوي على ٨ بت لكل بت وظيفة سوف نقوم بتوضحها

## INTCONالسجل

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|

**Bit7**

**وظيفة هذا البت تفعيل أو تعطيل جميع أنواع المقاطعات وله قيمتان**

**0**

**تعطيل جميع انواع المقاطعات**

**1**

**تفعيل جميع انواع المقاطعات**

**ومن هنا يتضح أنك اذا اردت استخدام المقاطعات فلابد من تفعيل هذا البت**

**Bit6**

**ADCتفعيل أو تعطيل الاجهزة الملحقة مثل المحول التماثلي الرقمي**

**0**

**ADCتعطيل الاجهزة الملحقة**

**1**

**تفعيل الاجهزة الملحقة**

**Bit5**

**TMR0تفعيل أو تعطيل عداد التايمر زيرو**

**0**

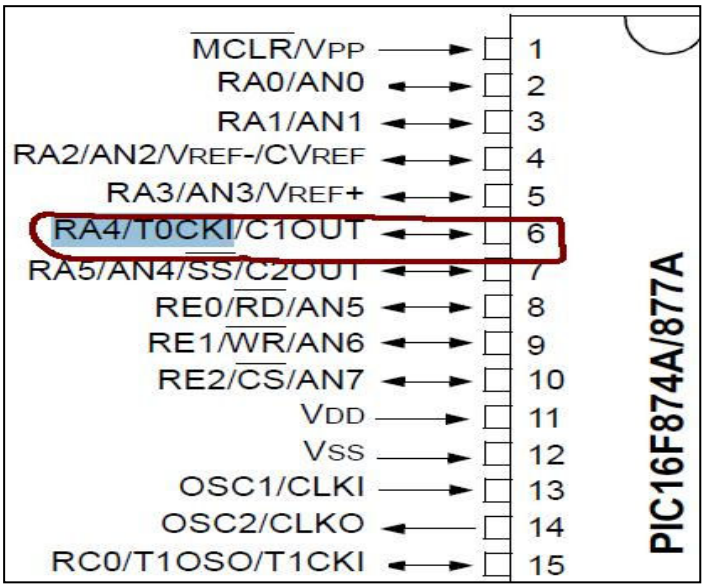
**تعطيل التايمر زيرو**

**١**

**تفعيل التايمر زيرو**

**والتايمر زيرو هو عداد داخلي يعد عدد النبضات سواء كانت نبضات الكرستالة أو العداد الخارجي الطرف**

**RA4/T0CKI**




**Bit4**

**تفعيل أو تعطيل مقاطعة تغير الجهد عند الطرف**

**RB0**

**0**

**RB0عدم تفعيل المقاطعة عند التغير في الجهد عند الطرف**

**1**

**RB0تفعيل مقاطعة تغير الجهد عند الطرف**

**Bit3**

**تفعيل أو تعطيل مقاطعة التغير في الجهد عند أحد الاطراف التالية**

**RB4 or RB5 or RB6 or RB7**

**0**

**تعطيل**

**١**

**تفعيل**

**Bit2**

**تفعيل أو تعطيل علم طفحان التايمر زيرو وبمعنى اخر**

**TMR0**

**التايمر يعد من ٠حتى ٢٥٥**

**وبعد ذلك يبدأ العد من جديد والطفحان هنا يعني تجاوز ٢٥٥ لذلك سوف يعود الى الصفر**

**وهذا العلم يخبرنا أذا كان التايمر تعدى ٢٥٥ ام لا ليعود ثانية الى الصفر فاذا كان صفر يعني أنه لم يصل قيمة التايمر الى ٢٥٥**

**واذا كان واحد يعني أن التايمر وصل الى ٢٥٥**

**ومن هنا يجب أن يكون هذا البت غير مفعل في بداية البرنامج لان التايمر لم يبدأ العد**

**Bit1**

**هو علم تغير الجهد عند**

**RB0**

**فاذا كان صفر يعني عدم تغير الجهد**

**واذا كان واحد يعني تغير الجهد**

**وكذلك يجب أن يكون في بداية البرنامج بصفر لان الجهد لم يتغير بعد**

**Bit0**

**علم تغير الجهد عند هذه الاطراف**

**RB4,RB5,RB6,RB7**

**فاذا كان بصفر يعني عدم تغير الجهد**

**واذا كان بواحد يعني تغير الجهد**

**وكذلك يجب أن يكون في بداية البرنامج بصفر لان الجهد على هذه الاطراف لم يتغير بعد**

**وتغير الجهد عند أحد اطراف الميكروكنترولر يعني أنه اذا كان بصفر يصبح ٥ فولت واذا كان ٥ فولت يصبح بصفر**

## السجلOPTION_REG

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|

**Bit7**

**وظيفة هذا البت تمكين أو تعطيل سجل**

**option_reg**

**من العمل**

**0**

**تعطيل**

**١**

**تمكين**

**Bit6**

**هذا البت خاص بكيفية التغير في فولت على الطرف**

**RB0**

**هل يكون التغير من صفر الى خمسة فولت ام من خمسه الى صفر فولت**

**وعلى هذا فأن هذا البت يتصل مع سجل المقاطعة**

**Intcon**

في البت ٤ الخاص بتغير الجهد على الطرف

**RB0**

**0**

من خمسة فولت الى صفر فولت

١

من صفر فولت الى خمسة فولت

**Bit5**

يحدد مصدر العد للتايمر زيرو هل العد من الكرستالة ام من طرف

**RA4**

**0**

**RA4**يصبح العد من الطرف الخارجي

**1**

يصبح العد من الكرستالة الخاصة بالميكروكنترولر

**Bit4**

يقوم بتحديد كيف يكون العد في الطرف

**RA4**

هل سيكون من خمسة فولت الى صفر ام من صفر الى خمسة فولت

**0**

يصبح العد من خمسة فولت الى صفر فولت

١

يصبح العد من صفر فولت الى خمسة فولت

وعلى هذا فان هذا السجل لا يتم التفكير فيه الا اذا بت خمسة يساوي صفر او عند استخدام العداد الخارجي

**Bit3**

يحدد هل سيكون معامل القسمة للتايمر زيرو او للعداد الخارجي الطرف **RA4**

وهذه الفقرة تحتاج الى شرح حيث أن

تردد الكرستالة او المذبذب هل سوف يقسم ام لا

وحيث أن اقصى سرعة للميكروكنترولر ينفذ بها العمليات تساوي ربع قيمة الكرستالة والعد في التايمر من ٠ حتى ٢٥٥

| تردد الكرستالة |
| --- |
| ٤*٢٥٦* معامل القسمة |

اذن معامل القسمة سوف يحدد سرعة التايمر زيرو واذا لم يحدد نعتبره بواحد

**0**

**TMR0**للتايمر زيرو

**1**

للطرف**RA4**

**Bit 2,1,0**

**هما قيمة معامل القسمة اذا فعل**

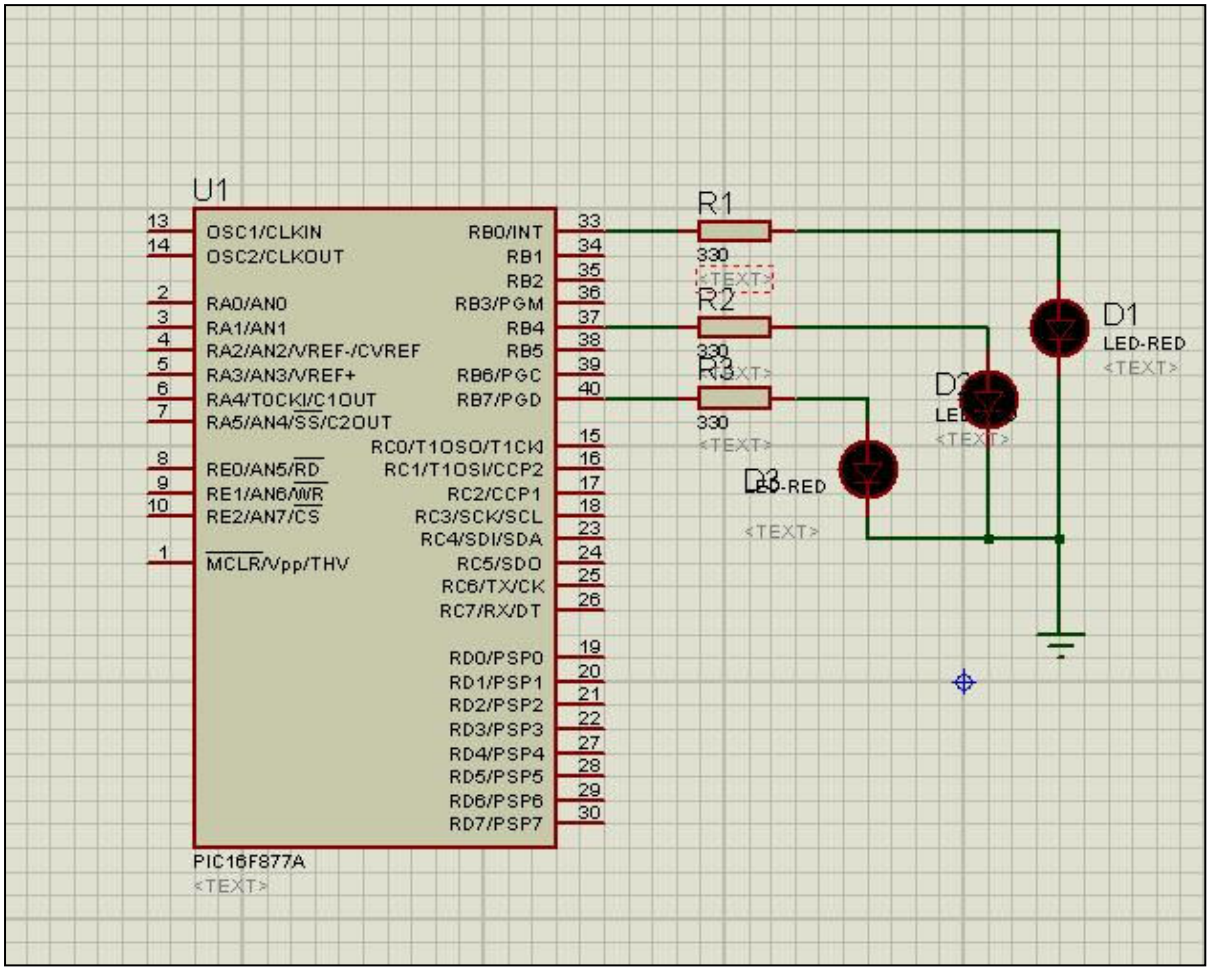
**ويفضل أن تكون هذه البتات بصفر حتى لا يحدث اخطاء**

## امثلة على المقاطعات

**اولا مثال على التايمر .**

**والمثال كتالي الليد يضئ على احد الاطراف ثم يطفأ لمدة ثانية**

**اولا قم برسم الدائرة على برنامج بروتس كالتالي**



## ثانيا كتابة الكود

```
void main() {
 trisb=0; // خرج
 portb=0;
 // بداية العد في التايمر ٠
 tmr0=0;
 // البت ٧ لتفعيل جميع انواع المقاطعات
 // البت ٥ لتفعيل التايمر زيرو
 intcon=0b10100000;

 //البت ٧ بواحد لتفعيل المسجل
 // البت ٥ بصفر حتى يعمل على التردد الداخلي للميكرو
 //البت ٣ بصفر حتى يكون معامل القسمة للتردد الداخلي للميكرو
 // البت ٢و١و٠ بصفر حتى تكون القسمة على ١
 option_reg=0b10000000;
 while(1){
  portb;

 }
}
```

وكتابة برنامج المقاطعة

ملحوظة :

الكرستالة المستعملة هي ٨ ميجا هيرتز

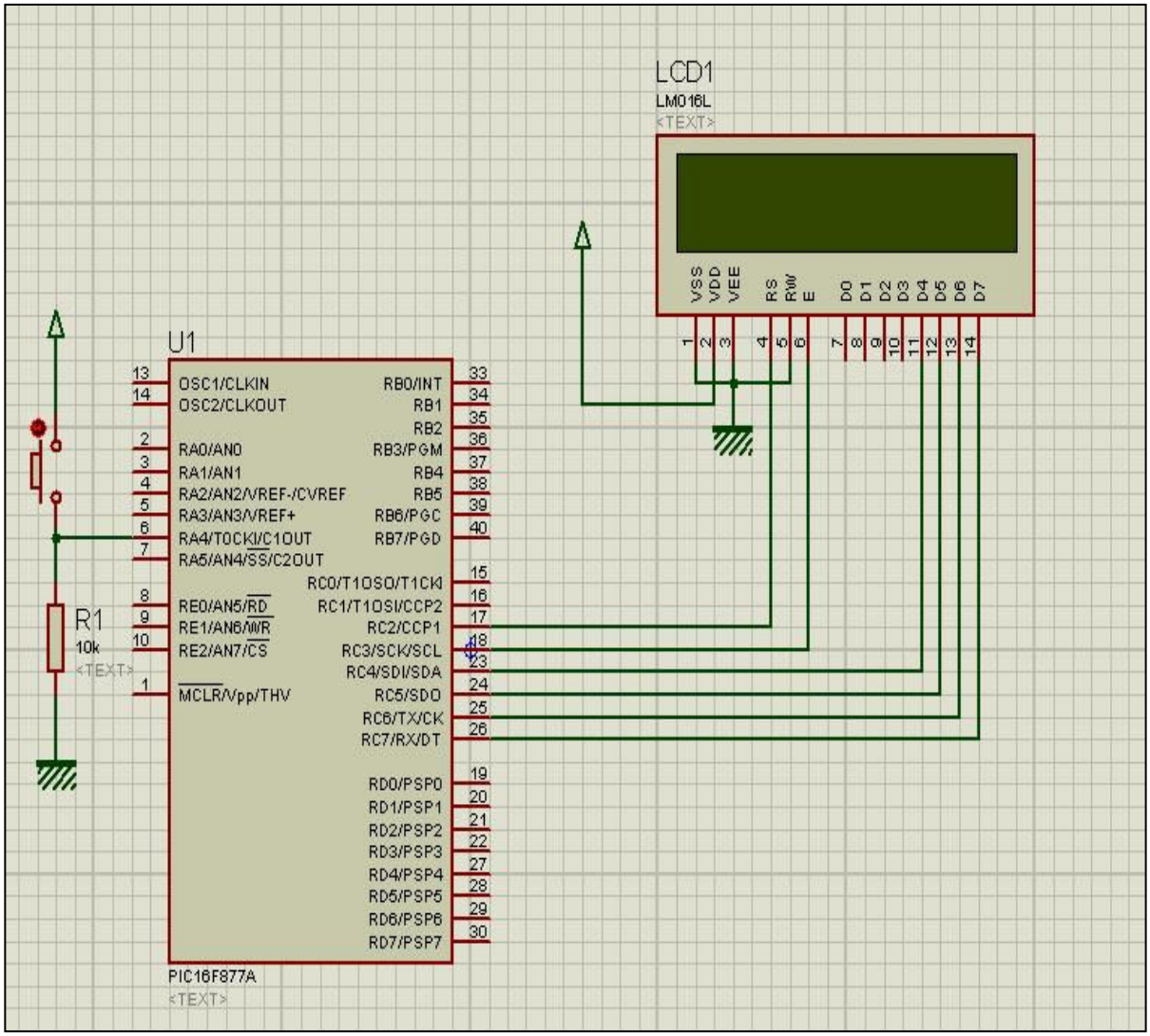
```
*/
int y;
void interrupt(){
 y++;// الزيادة عند طفحان التايمر زيرو
if(portb==0){
  if(y>=3906){ //الزمن واحد ثانية
   portb=255;
   y=0;

 }
  }else{

  if(y>=3906){
   portb=0;
   y=0;
  }
   }
  //علم مقاطعة التايمر زيرو
  intcon.f2=0;


}
```

**المشروع الثاني هو عداد خارجي يعد من الطرف**

**RA4**

**عدد النبضات او الضغطاط الداخلة**

**قوم بتنفيذ الرسم على برنامج بروتس كالتالي**




# الكود مع الشرح

```
//تعريف الشاشة
sbit LCD_RS at RC2_bit;
sbit LCD_EN at RC3_bit;
sbit LCD_D4 at RC4_bit;
sbit LCD_D5 at RC5_bit;
sbit LCD_D6 at RC6_bit;
sbit LCD_D7 at RC7_bit;
sbit LCD_RS_Direction at TRISC2_bit;
sbit LCD_EN_Direction at TRISC3_bit;
sbit LCD_D4_Direction at TRISC4_bit;
sbit LCD_D5_Direction at TRISC5_bit;
sbit LCD_D6_Direction at TRISC6_bit;
sbit LCD_D7_Direction at TRISC7_bit;
// متغيرات المستعملة مع الشاشة
int temp=0;     char txt[7];
// المتغير x
//حتى يتم العد بعد نهاية طفحان التايمر زيرو
int x;
```

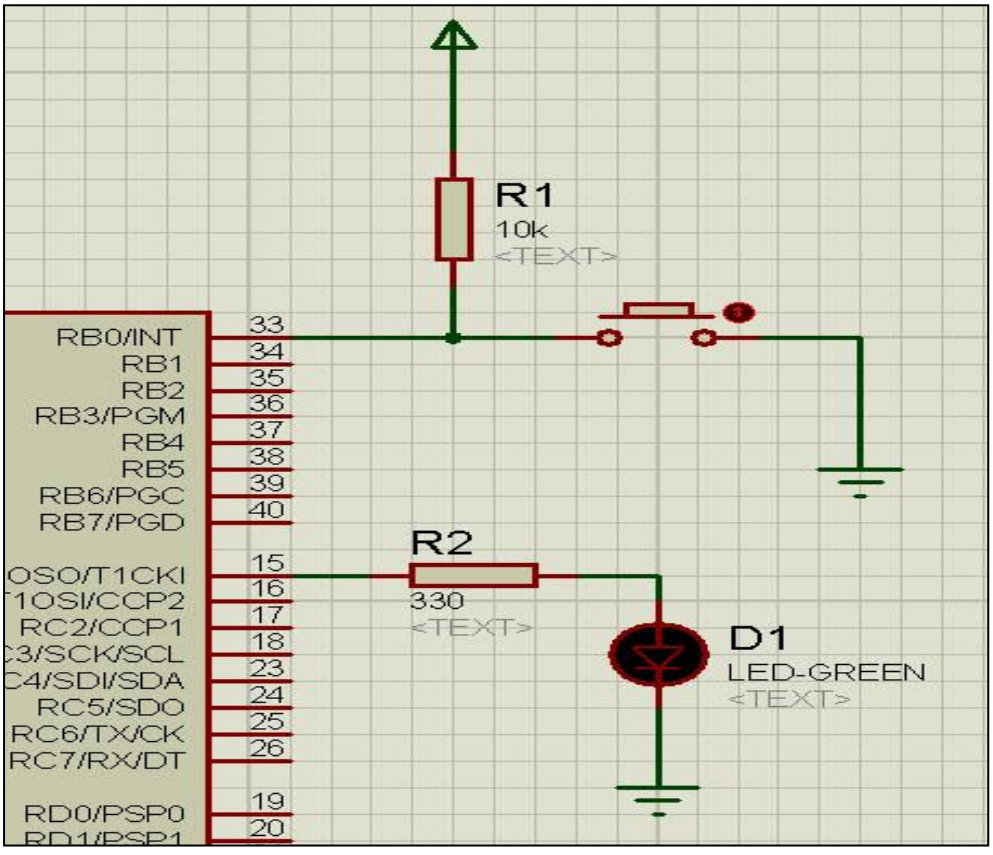
```
void main() {
trisc=0; // خرج
portc=0;
 Lcd_Init();// تعريف الشاشة
 Lcd_Cmd(_LCD_CURSOR_OFF); // منع السهم من الظهور
 intcon=0b10100000;
 // البت ٧ لتفعيل جميع انواع المقاطعات
 // البت خمسة لتفعيل التايمر زيرو

 OPTION_REG=0b10101000; //تعريف أن التايمر خارجي
 // البت ٧ لتمكين السجل من العمل
 // البت خمسة حتى يكون العد من RA4
 // البت ٣ حتى يكون معامل القسمة للعداد الخارجي
 // الطرف RA4

 while(1){
 // وضعنا هنا التايمر زيرو حتى يكون دائما نشطا
 // اما القيمة ٢٥٥ حتى يكون بعدما بصفر فتزيد القيمة x
 TMR0=255;
 temp=x;
 IntToStr(temp, txt);
 Lcd_Out(1,8,txt);
```

```
void interrupt(){
 x++; // الزيادة في العدد
 intcon.f2=0; // علم مقاطعة التايمر زيرو

}
```

**المشروع الثالث تغير الجهد عند الطرف**

**RB0**

**نفذ المشروع على برنامج بروتس كالتالي**




**ثانيا كتابة الكود**

```
void interrupt(){
 if (portc==0){
 portc=255;
  }else{
   portc=0;
  }
 intcon.f1=0; //علم مقاطعة الطرف RBO
}
void main() {
 trisb=1;
 trisc=0;
 portc=0;
 intcon=0b10010000;
 // البت ٧ مفعل لتفعيل جميع انواع المقاطعات
 // البت ٤ مفعل لتفعيل تغير الجهد عند الطرف RBO
 option_reg=0b10000000;
 //البت ٧ مفعل لتفعيل السجل
 // البت ٦ غير مفعل لان التغير في الجهد يكونن ٥ الى صفر فولت
 while(1){
  portc;
  portb;

 }
}
```

والله الموفق

تمت بحمد الله

عيد فتحي