

الحمد لله رب العالمين... والصلاة والسلام الأتمان الأكملان على سيدنا محمد بن عبد الله صاحب الخلق العظيم ...
أما بعد ...

حديثنا اليوم بإذن الله عن استخدام لوحة المفاتيح المصغرة Keypad لإدخال البيانات إلى المتحكم الصغري ...
لوحة المفاتيح ببساطة :

هي مصفوفة من القواطع الكهربائية ... يقوم كل زر فيها بوظيفة وصل السلك المار بصفه بالسلك المار بعموده ...

بعض أشكال لوحة المفاتيح :

لوحة مفاتيح الهاتف ...

يقوم الزر 1 بوصل السلك A في صفه ... بالسلك رقم 1 في عموده ..

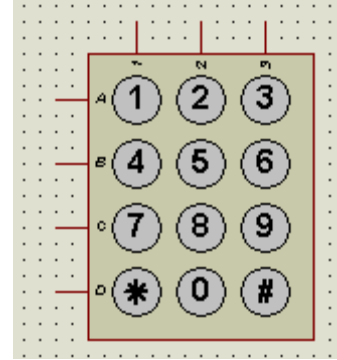
يقوم الزر 2 بوصل السلك A في صفه ... بالسلك رقم 2 في عموده

يقوم الزر 3 بوصل السلك A في صفه ... بالسلك رقم 3 في عموده

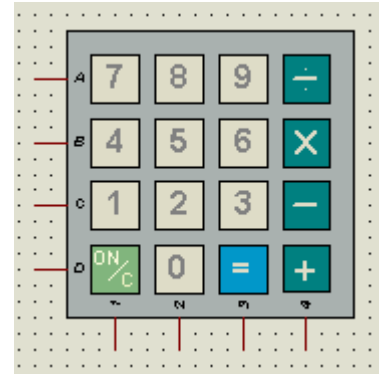
يقوم الزر 4 بوصل السلك B في صفه ... بالسلك رقم 1 في عموده

يقوم الزر * بوصل السلك D في صفه ... بالسلك رقم 1 في عموده

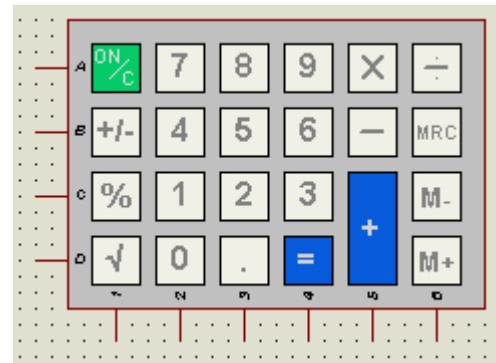
يقوم الزر # بوصل السلك D في صفه ... بالسلك رقم 3 في عموده وهكذا ..



لوحة مفاتيح الآلة الحاسبة البسيطة :

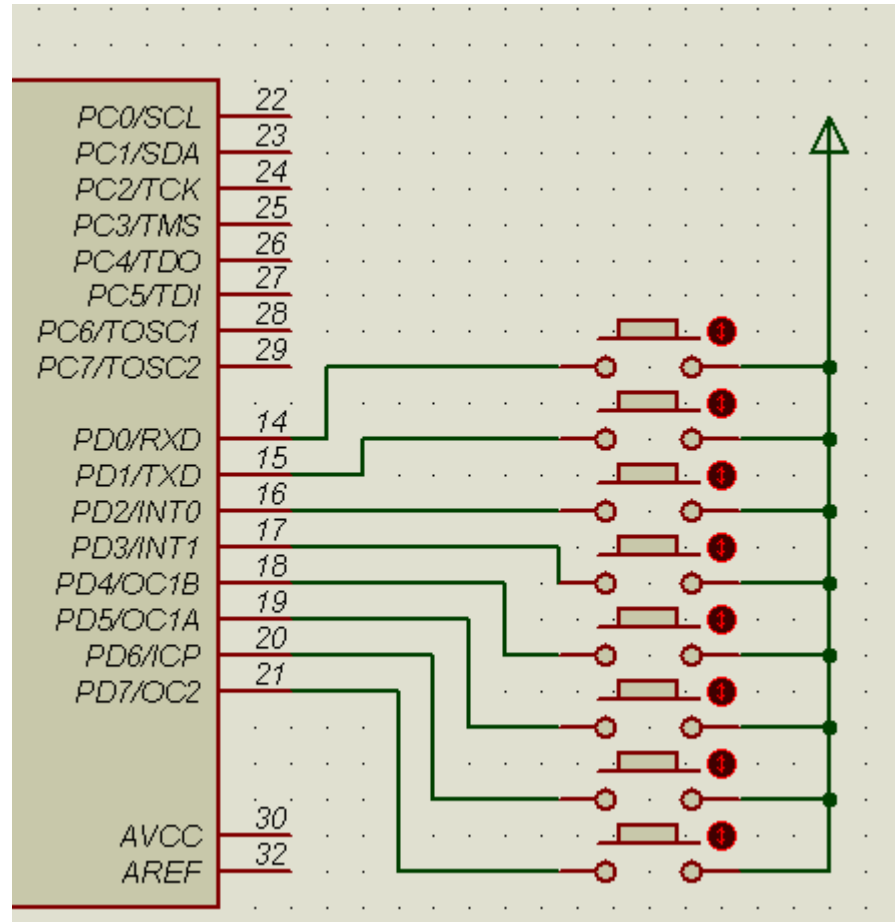


لوحة مفاتيح آلة حاسبة أكثر تطوراً ..



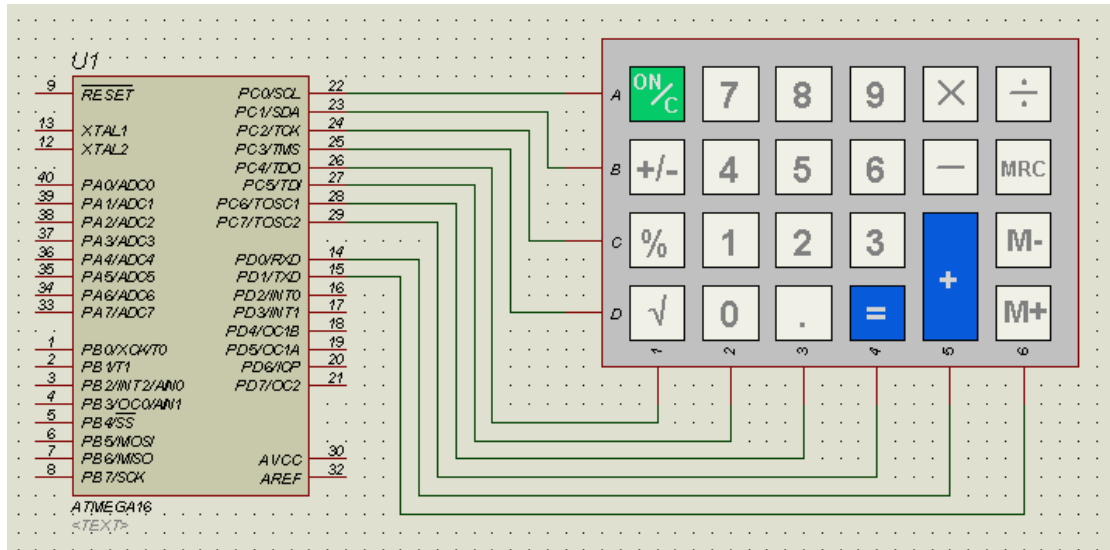
سيكون درس اليوم على لوحة المفاتيح الأخيرة ذات الأزرار الكثيرة ... لإزالة الرهبة من جميع لوحات المفاتيح ☺

لو أردنا وصل ثمانية أزرار إلى المتحكم الصغري بالطريقة التقليدية فإننا سنستهلك ثمانية أرجل للمتحكم .. كما يلي :



هذه الطريقة في التوصيل استهلك عدد من الأرجل مساويا لعدد الأزرار وهذا كثير جداً ...
فحتى نصل 32 زرا فقط ...سنستهلك كل أرجل المتحكم ...عدا عن الزمن اللازم لإنجاز هذا التوصيل والكلفة المادية...

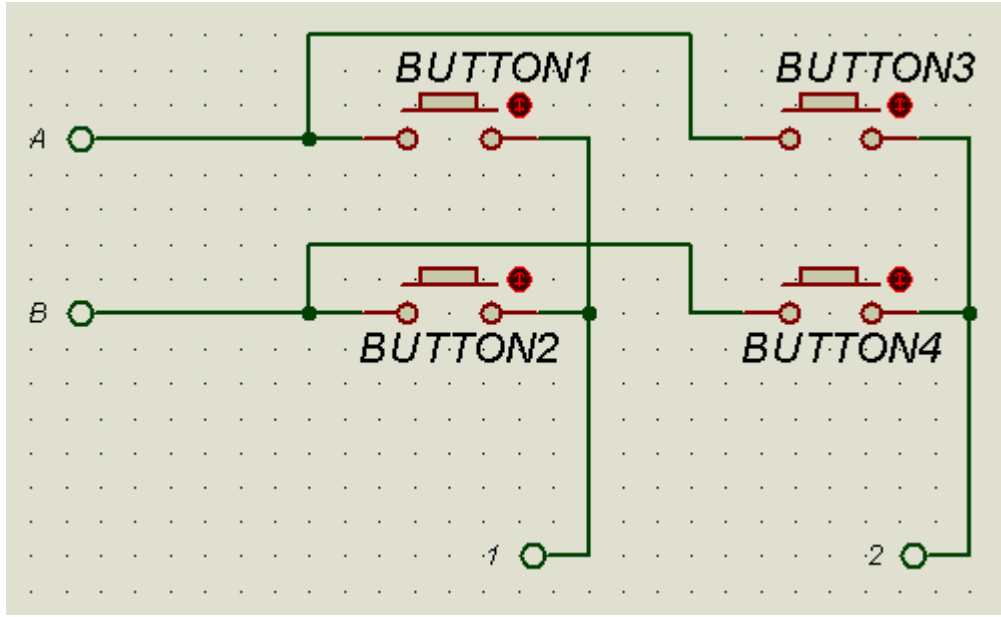
ولذلك كانت الفكرة البسيطة والعملية جداً .. باختراع لوحة المفاتيح ..



لاحظ كيف أننا وصلنا 24 زرّاً بـ 10 أرجل فقط .. بل ويمكننا وصل 100 زر إلى عشرين رجلا من أرجل المتحكم فقط

فكيف يتم ذلك ؟؟

لو افترضنا التوصيل البسيط التالي لأربع أزرار :



إذا قمنا بتطبيق جهد على النقطة A .. ثم ضغطنا الزر BUTTON1 فسيظهر الجهد على النقطة 1

أما لو ضغطنا الزر BUTTON3 فسيظهر الجهد على الرجل 2

ونلاحظ أن ضغط الأزرار BUTTON2 و BUTTON4 ليس له تأثير ...

أما لو قمنا بتطبيق الجهد السابق على النقطة B بدلا من A فسيكون التالي :

بضغط الزر BUTTON2 سيظهر الجهد على النقطة 1

بضغط الزر BUTTON4 سيظهر الجهد على النقطة 2

وبهذا نحصل على ثنائيات ... كل زر يدلنا على ضغطه ثنائية مميزة ..
كما يلي :

BUTTON1 → A,1

BUTTON2 → B,1

BUTTON3 → A,2

BUTTON4 → B,2

بالانتقال إلى المتحكم ... ما سنفعله هو اعتبار النقاط A,B,C,... خرجا للمتحكم

والنقاط 1,2,3,... دخلا للمتحكم ...

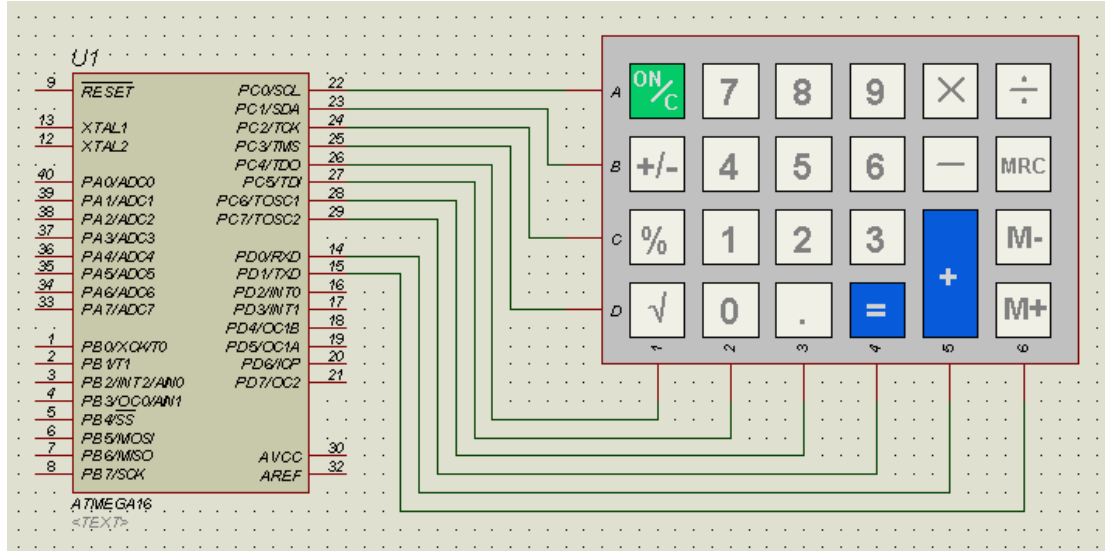
ونختبر كما يلي :

بتطبيق جهد (كخرج قيمته 1 منطقي) على النقطة A مثلاً ...

إذا كانت النقطة 1 (كدخل) قيمتها 1منطقي فلا بد أن الزر BUTTON1 مضغوط ... وإلا . انتقل لاختبار النقطة 2 وهكذا ... حتى نختبر جميع الأعمدة (كدخل) من أجل جميع الأسطر (كخرج) .

المثال التالي للتوضيح ..

ليكن لدينا الدارة المبينة في الشكل التالي :



لاحظ أن الأرجل الموصولة بالأسطر هي التالية : C0,C1,C2,C3 :

والأرجل الموصولة بالأعمدة هي C4,C5,C6,C7,D0,D1

والآن كل ما علينا فعله هو وضع القيمة 1 على الأرجل الموصولة بالأسطر كل على حدة

ومن أجل كل سطر نختبر جميع الأعمدة ... وأول قيمة 1 نصادفها نتوقف لأننا وجدنا الزر المضغوط ..

أو يمكننا عدم التوقف عند هذا الزر ومتابعة الاختبار حتى انتهاء الاحتمالات (وذلك بعد استدعاء التابع المرتبط بالزر الاول)

الكود التالي للتوضيح :

بداية سنقوم بتعريف مصفوفة ثنائية البعد عناصرها مطابقة لأزرار لوحة المفاتيح ...

في المثال التالي اخترت أن تكون المصفوفة تحتوي نصا يدل على الزر كما يلي :

```
char *keypad[4][6] =  
{  
    {"on"      , "7", "8", "9",      "mul"   , "div"},  
    {"sign"    , "4", "5", "6",      "minus" , "memory/read/clear"},  
    {"percent" , "1", "2", "3",      "plus"  , "memory-"},  
    {"sqrt"    , "0", ".", "result", "plus"  , "memory+"},  
};
```

والآن عملية اختبار الأزرار لمعرفة أيها هو المضغوط :

```
int i,j;
for(i=0;i<4;i++)
{
    PORTC.i=1;   نضع جهداً على الرجل رقم i
    for(j=4;j<8;j++)
    {
        حلقة الأعمدة (خاصة بأول 4 أعمدة)
        if(PINC.j==1)
            return keypad[i][j]
    }
    for(j=0;j<2;j++)
    {
        حلقة الأعمدة (خاصة بآخر عمودين)
        if(PIND.j==1)
            return keypad[i][j+4]
    }
    PORTC.i=0;   نزيل الجهد الذي وضعناه في البداية
}
.
```

ملاحظة هامة : الكود السابق للشرح فقط وهو غير قابل للتنفيذ مباشرة لأن عبارة PORTC.i مرفوضة من قبل المترجم لأن المتحول الذي يتلو النقطة يجب أن يكون ثابت ... نستعيز عن ذلك بعمليات and& و or| كما يلي :

لوضع واحد في البت رقم i نكتب : $PORTC|=1<<i$

لوضع صفر في البت رقم i نكتب : $PORTC\&=\sim(1<<i)$

شرح العملية السابقة :

على اليسار نضع الرقم ... في الوسط نضع عامل إزاحة لليمين أو اليسار... وعلى اليمين نضع عدد بتات الإزاحة ... وبذلك يكون $i<<1$ تعطي 000010000 حيث أن الواحد في الموضع i وعند استخدام المعامل ~ يصبح 111101111 حيث أن الواحد في الموضع i

الكود السابق يمكننا من إنشاء تابع يعيد سلسلة محرفية تدل على اسم الزر المضغوط ...

ويمكنك من خلال نفس الآلية فعل أي شيء بعد ذلك ...

مثلاً يمكنك عمل عدة توابع .. ثم تختبر الزر المضغوط فإذا كان '+' مثلاً ... تقوم باستدعاء تابع الجمع .. وهكذا..

باختصار ..بعد أن فهمت مبدأ لوحة المفاتيح .. ضع اللوحة حيث تريد واختر الأرجل المناسبة لك ... ثم اكتب الكود بالتوافق مع الدارة التي صممتهـا ...

هذا والله ولي التوفيق

والحمد لله رب العالمين