سلسلة المشاريع مفتوحة المصدر

المشروع رقم ١

تأليف/ أحمد سمير فايد

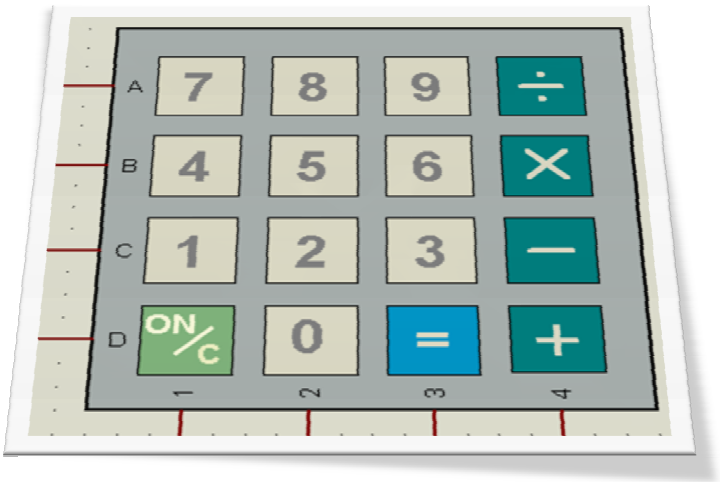
Ahmad_s_f@hotmail.com

مشروع التحكم في 48 ريلاي(relay) باستخدام keypad و LCD

لدينا في هذه الدائرة 48 ريلاي يتم ترقيمهم من صفر إلى 47 .

كيف تستخدم الدائرة ؟؟

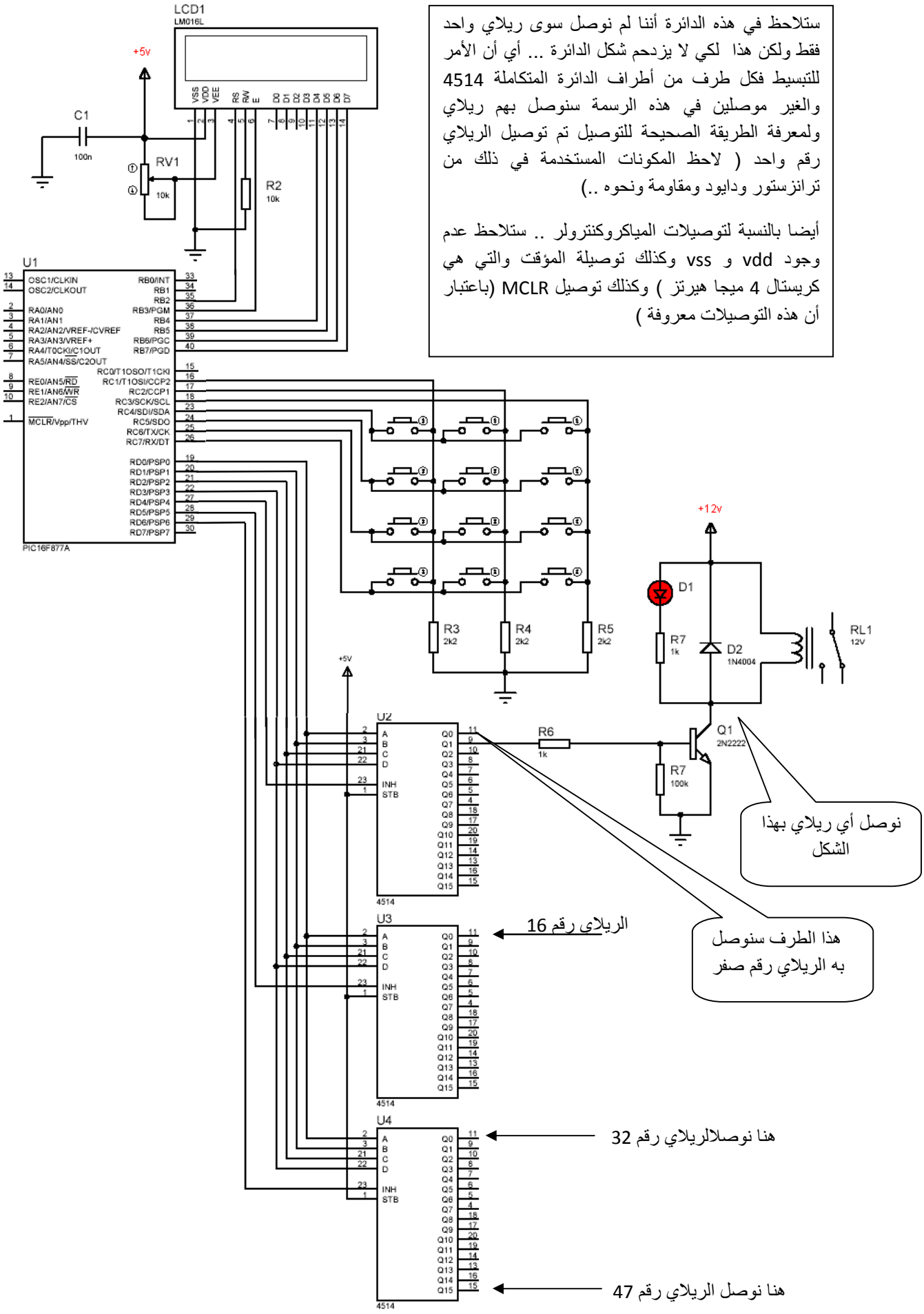
يتم كتابة رقم الريلاي المراد تشغيله باستخدام keypad ثم يتم الضغط على "=" حينها يعمل الريلاي صاحب الرقم الذي كتبناه . ولتغيير الريلاي الذي يعمل يتم الضغط على "ON/C" حينها لن يعمل أي ريلاي فنكتب رقم ريلاي آخر و الذي نريد تشغيله ثم بعد ذلك نضغط على "=" وهكذا كما نريد ..



هذه صورة الـ keypad المستخدمة في هذا المشروع علما أنا العمود الأول غير مستخدم والذي فيه علامات + - × /

ستلاحظ في هذه الدائرة أننا لم نوصل سوى ريلاي واحد فقط ولكن هذا لكي لا يزدحم شكل الدائرة ... أي أن الأمر للتبسيط فكل طرف من أطراف الدائرة المتكاملة 4514 والغير موصلين في هذه الرسمة سنوصل بهم ريلاي ولمعرفة الطريقة الصحيحة للتوصيل تم توصيل الريلاي رقم واحد (لاحظ المكونات المستخدمة في ذلك من ترانزستور ودايود ومقاومة ونحوه ..)

أيضا بالنسبة لتوصيلات المياكروكنترولر .. ستلاحظ عدم وجود vdd و vss وكذلك توصيلة المؤقت والتي هي كريستال 4 ميجا هيرتز) وكذلك توصيل MCLR (باعتبار أن هذه التوصيلات معروفة)



الكود الخاص بالبرنامج

```
/****************************************
Project : this project is designed to control of 48 relays by typing the number of relay by using keypad
        then this number displayed on LCD  and this relay which has this number will be ON if you push "="
        button.  and you can clear LCD to write another number so the another relay  will be ON ,you can do
        this if you push"ON/c" button.hardware: I use in this project pic16f877a , LCD 2*16 , IC 4514,relays
        and keypad.  you can use another hardware by some adjusting.
Author    : Ahmad Samir Fayed.
Country   : Egypt.
Date      :26/2/2009
copyright : all muslims and Arabs can use this code but without replacing   my name to another when using
                the same code.   (plese don't waste my fatiguing)
****************************************/
char m;
char result=0;
char number[]={'x',' ','9','8','7','x','6','5','4','x','3','2','1','x','x','0','x'};
char number2[]={'x',' ',9,8,7,'x',6,5,4,'x',3,2,1,'x','x',0,'x'};
char tens=0,ones=0;
void on() {
         result=(tens*10)+ones;
         if(result<16){result.f4=0;result.f5=1;result.f6=1;}
         else if(result>=16&&result<=31){result=result-16;result.f4=1;result.f5=0;result.f6=1;}
         else if(result>=32&&result<=47){result=result-32;result.f4=1;result.f5=1;result.f6=0;}
         else if(result>47){Lcd_Cmd(Lcd_Clear);lcd_out_cp("error");delay_ms(1000);return;}
         portd=result;
        }
void main()
{
trisd=0;portd=0xF0;
Keypad_Init(&PORTC);
```

```
lcd_init(&portb);
lcd_cmd(LCD_CURSOR_OFF);
lcd_out(1,1,"Enter the number");
lcd_out(2,1,"    of relay");

loop:              //testing the first pushing down
m=keypad_read();
if(m==0)     {goto loop;}
else if(m==14){goto loop;}
else if(m==16){Lcd_Cmd(Lcd_Clear);portd=0xF0;goto loop;}
else        {Lcd_Cmd(Lcd_Clear);tens=number2[m];lcd_chr_cp(number[m]);delay_ms(500);}

test2:             //testing the second pushing down
m=keypad_read();
if(m==0)     {goto test2;}
else if(m==14){ones=tens;tens=0;on();goto test3;}
else if(m==16){Lcd_Cmd(Lcd_Clear);portd=0xF0;goto loop;}
else        {ones=number2[m];lcd_chr_cp(number[m]);delay_ms(500);}

test3:           //testing the third pushing down
m=keypad_read();
if(m==0)     {goto test3;}
else if(m==14){on();goto test3;}
else if(m==16){Lcd_Cmd(Lcd_Clear);portd=0xF0;goto loop;}
else        {goto test3;}
}
```

وإليكم الآن شرح موجز عن الكود السابق :

أول 12 سطر عبارة عن مقدمة (تعليق) وبهم شرح كيفية عمل الدائرة .. وبما أنهم جميعا محصورين بين العلامتين */ /* فإنهم جميعا لن يلتفت إليهم البك عند تنفيذ المشروع ولن تتم عملية ترجمة هذه الأسطر .

في هذا المشروع تم استخدام كل من :

1- الشاشة من النوع LCD 2*16
2- لوحة مفاتيح keypad
3- الدائرة المتكاملة 4514

وسأشرح الآن الكود الخاص باستخدام كل جزء ثم بعد ذلك أشرح الكود كاملا .

أولا : الشاشة LCD :-

في أي مشروع يتم فيه استخدام LCD يجب أن أحدد للبك في أي منفذ (port بورت) سيتم توصيل الشاشة هل هو PORTB أم PORTC أم .... وذلك بكتابة الأمر التالي :

```
lcd_init(&portb);
```

وفي حالة كتابتي للأمر السابق فهذا يعني أني وصلت LCD بالشكل التالي .

```
D7 → portb.7
D6 → portb.6
D5 → portb.5
D4 → portb.4
E  → portb.3
RS → portb.2
RW → ground
```

والآن كيف سأظهر كتابة معينة على LCD ؟؟

يمكن ذلك بعدة طرق..

فمثلا عندما نريد أن نكتب في السطر الأول من الشاشة ومن بداية العمود الأول (تذكر أن في مشروعنا الشاشة عبارة عن 2×16 أي سطرين و 16 عمود .) فإننا نكتب الأمر التالي :

```
lcd_out(1,1,"Enter the number");
```

الأمر السابق سيظهر الجملة Enter the number في السطر الأول وستبدأ هذه الجملة من السطر الأول .. ملحوظة : في الأمر السابق كان يمكن أن نضع الجملة التي نريد أن نظهرها في مصفوفة من المتغيرات فمثلا الأمر السابق سيؤدي نفس الوظيفة التي سيؤديها الأمر التالي :

```
lcd_out(1,1,text);
```

فهذا الأمر سيظهر النص الموجود في المصفوفة التي اسمها text في السطر الأول وتبدأ الجملة من العمود الأول . وهذا طبعا يتطلب أن أقوم بتعريف هذه المصفوفة في بداية البرنامج كالتالي:

```
Char lcd[]=""Enter the number";
```

إذا كتبت الأوامر السابقة فقط فستجد مؤشر يظهر ويختفي وهو ما يسمى بـ CURSOR هذا المؤشر سيكون مكانه بعد آخر حرف كتبته الشاشة وهو في مثالنا حرف r . ويمكنني أن أخفي هذا المؤشر CURSOR باستخدام الأمر التالي .

```
lcd_cmd(LCD_CURSOR_OFF);
```

يمكنني أيضا أن أظهر أي حرف أو رقم على الشاشة (وليس نص بل حرف واحد فقط) وذلك عن طريق الأمر التالي مثلا :

```
lcd_chr (2,1,'m');
```

في الأمر السابق سيظهر الحرف m في السطر الثاني – العمود الأول . ولكن ماذا لو أردنا أن يظهر حرف معين ولكن باستخدام المتغيرات أي ليس بشكل مباشر كما في الأمر السابق .

```
lcd_chr (2,1,tt);
```

في الأمر السابق سيتم إظهار ما يحتويه المتغير tt في السطر الثاني العمود الأول ولكن كما تعلم لابد أن أكون قد عرفت المتغير tt في بداية البرنامج كما في الأمر التالي

```
Char tt='m';
```

لابد أنك لاحظت العلامة ' ' هذه العلامة لابد منها فهي تجعل ما يحتويها محول إلى الاسكي كود والذي من خلاله نستطيع عرض الحرف على الشاشة فمثلا لو كتبت الأمر السابق بدون العلامة ' ' فلن يتم إظهار الحرف m على الشاشة . وهذا أمر هام يجب الانتباه إليه فمثلا لو أردت أن أعرض الرقم 3 على الشاشة فيجب حينها أن أعرف المتغير بالشكل التالي :

```
Char kk='3';
```

وبعده الأمر :

```
lcd_chr (2,1,kk);
```

أما إذا كتبت الأمر char kk=3 ففي هذه الحالة لن يظهر الرقم 3 على الشاشة لأننا لم نضع العلامة ' ' . ملحوظة : بالنسبة للكود الذي كتبته قمت بوضع مجموعة من المتغيرات في مصفوفة بكتابة الامر التالي .

```
char number[]={'x',' ','9','8','7','x','6','5','4','x','3','2','1','x','x','0','x'};
```

هذه المصفوفة عبارة عن 17 متغير من النوع char . ماذا يعني هذا الكلام ؟؟ إنه يعني أن الأمر السابق سيؤدي نفس الوظيفة التي تؤديها الأوامر التالية (وهي تعريف 17 متغير من النوع char)

```
Char Number0='x';
Char Number1=' ';
Char Number2='9';
Char Number3='8';
Char Number4='7';
```

```
Char Number5='x';
Char Number6='6';
Char Number7='5';
Char Number8='4';
Char Number9='x';
Char Number10='3';
Char Number11='2';
Char Number12='1';
Char Number13='x';
Char Number14='x';
Char Number15='0';
Char Number16='x';
```

هل أدركت الآن مدى أهمية المصفوفة ؟؟!!.. إنها توفر علينا الكثير من الأسطر . والآن ماذا لو أردت أن أعرض الرقم 4 على LCD ؟ سانظر إلى المصفوفة وأعتبر أن أول عنصر فيها هو عنصر ترتيبه صفر والعنصر الذي يليه ترتيبه واحد .. وهكذا . وبالتالي العنصر الذي فيه الرقم 4 هو عنصر ترتيبه 8 و لكي أظهر الرقم 4 على الشاشة سأكتب الأمر التالي .

```
lcd_chr (2,1,number[8]);
```

ولكي أظهر الرقم صفر على الشاشة سأكتب الأمر التالي :

```
lcd_chr (2,1,number[15]);
```

لأن ترتيب الرقم '0' في المصفوفة number هو 15 .

وإليك هذه المعلومة : ماذا لو أردت أن أظهر حرف معين على الشاشة في آخر مكان وصل عنده المؤشر . ببساطة سأكتب الأمر التالي :

```
lcd_chr _CP(number[15]);
```

في الأمر السابق سيظهر الرقم صفر على الشاشة ولكن في آخر مكان وصل إليه المؤشر . ولو كتبت الأمر التالي بعد الأمر السابق .

```
lcd_chr _CP(number[8]);
```

فسيظهر الرقم 4 بجوار الرقم صفر .

كما يمكنني أن أظهر رقم معين (أو حرف) بطريقة أخرى وذلك باستخدام متغير إضافي فمثلا لو أردت أن أعرض الرقم 9 على الشاشة (ترتيب الرقم 9 في المصفوفة هو 2) يمكنني أن أكتب الكود التالي :

```
m=2;
lcd_chr _CP(number[m]);
```

الأمرين السابقين سيؤديا نفس الوظيفة التي يؤديها الأمر التالي :

```
lcd_chr _CP(number[2]);
```

وطبعا لا تنسى في بداية البرنامج أن تكون قد عرفت المتغير m

```
Char m;
```

وأخيرا إذا أردت أن تمسح الشاشة (أي لا يظهر عليها أي شيء) اكتب الأمرالتالي :

```
Lcd_cmd(lcd_clear);
```

ثانيا : لوحة المفاتيح keypad :

الكاييباد تتكون من مجموعة من المفاتيح (السويتشات) هذه المفاتيح متصلة بالبك بطريقة معينة (وبما أن الكاييباد تتكون من صفوف وأعمدة من السويتشات وكل صف يحتوي على سويتشات متصلة ببعض بسلك معين وكل عمود متصلة به السوتيشات بسلك معين) وعلى هذا فإن الكاييباد التي من النوع 4×4 لها أطراف معينة أربعة منهم للصفوف وأربعة منهم للأعمدة . والكاييبات التي 3×4 لها أطراف معينة أربعة منهم للصفوف وثلاثة منهم للأعمدة .

ولذلك فإنه عند الضغط على أحد هذه المفاتيح فإن الصف الذي به هذا السويتش يوصل إشارة معينة للبك وكذلك فإن العمود الذي به هذا المفتاح يوصل إشارة معينة للبك على طرف آخر من البك . ولهذا أحتاج لجمل if كثيرة لكي أستطيع استخدام 16 سويتش مثلا ..

ولكن الأمر أسهل بكثير مع لغة مايكروسي MikroC . فهناك مكتبة جاهزة للكاييباد تستطيع من خلال أمر واحد فقط معرفة أي سويتش تم الضغط عليه . وأمر آخر إضافي تكتبه لكي تخبر فيه البك عن المكان الذي تم توصيل الكاييباد به .

فمثلا لو وصلت الكاييباد بـ portc فسأكتب الأمر التالي في بداية البرنامج : -

```
Keypad_Init(&PORTC);
```

وأياً كانت طريقة توصيلك فلن تفرق كثيراً في عملية البرمجة

والآن كيف أعرف ما هو المفتاح الذي تم الضغط عليه .. ببساطة بواسطة الأمر التالي :-

```
keypad_read();
```

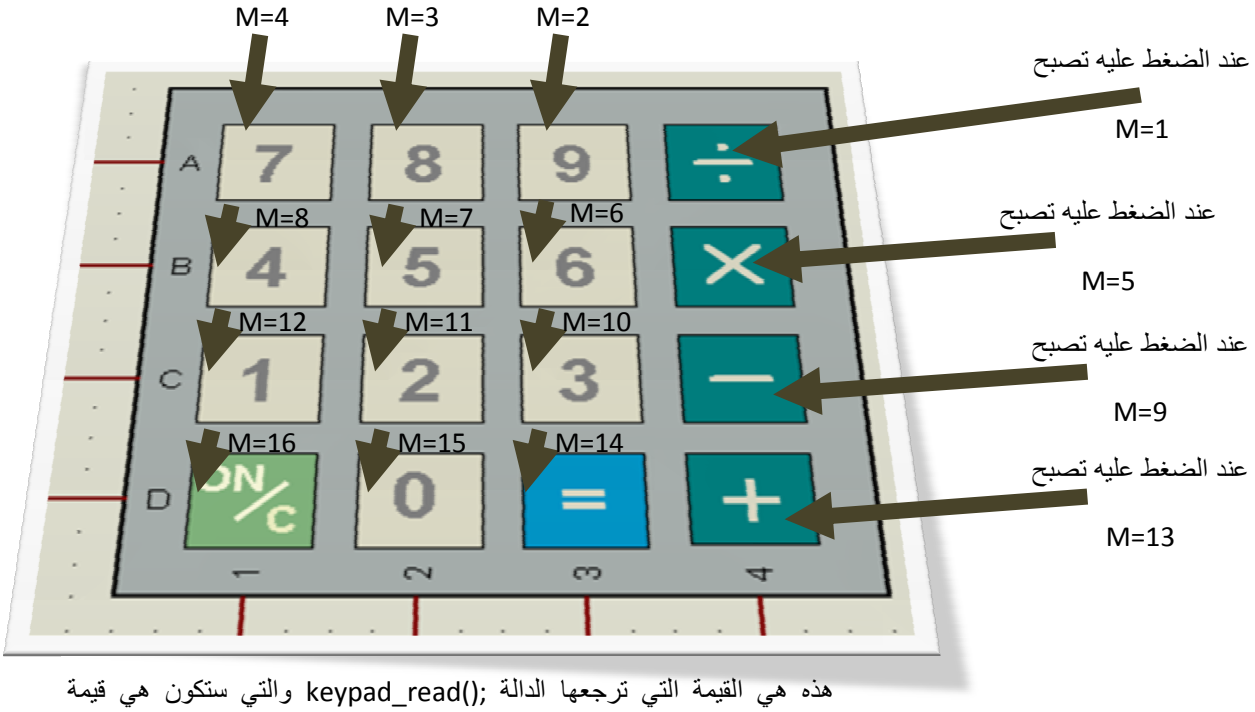
ماذا يفعل الأمر السابق ؟؟ إن هذا الأمر هو عبارة عن استدعاء لدالة معينة داخل مكتبة الكاييباد الموجودة تلقائيا في لغة MikroC . ولكن ماذا أستفيد من استدعاء هذه الدالة بالضبط؟؟

أخي الحبيب هذه الدالة تختبر السويتشات جميعا فإذا لم يتم الضغط على أي ستويتش فإن هذه الدالة ترجع القيمة صفر . وإذا تم الضغط على أحد السويتشات فإنها ترجع قيمة معينة إما أن تكون واحد أو اثنين أو .... أو 16 . ولتوضيح الأمر أكثر سأضرب لك هذا المثال :

نفرض أننا الآن عرفنا متغير اسمه m وساوينا هذا المتغير بهذه الدالة :

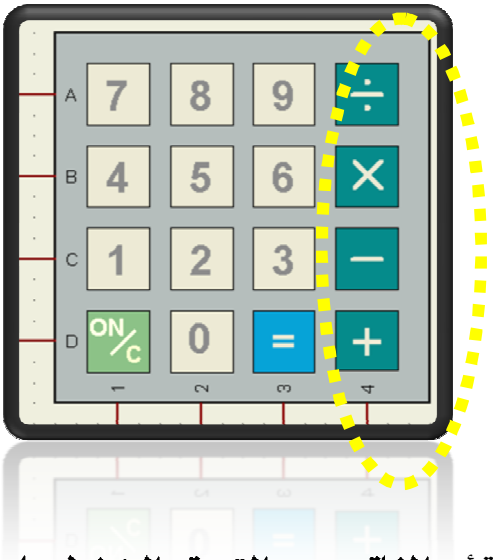
```
m=keypad_read();
```

فإنه في حالة عدم الضغط على أي سويتش فإنه في هذه الحالة ستكون قيمة المتغير m تساوي صفر . وإذا وصلت الكاييباد بطريقة معينة فإنه عند الضغط على السويتش ÷ فستكون قيمة المتغير m بواحد . وعند الضغط على السويتش 9 فإن المتغير m تصبح قيمته 1 وعند الضغط على السويتش 8 فإن المتغير m تصبح قيمته 2 ... وعند الضغط على الزر ON/c فإن قيمة المتغير m تصبح 16 .. وهكذا . ومن الكلام السابق يتضح أن لكل زر (أو مفتاح) رقم معين هذا الرقم سيصبح موجود في المتغير m . وفي مشروعنا هذا تم توصيل الكاييباد بحيث أن كل مفتاح سيكون له رقم معين كما يوضح ذلك الشكل التالي :



هذه هي القيمة التي ترجعها الدالة keypad_read(); والتي ستكون هي قيمة المتغير m لأننا ساوينا قيمة المتغير بالقيمة التي ترجعها الدالة .**وهذا في حالة أننا وصلنا جميع الأعمدة والصفوف بالبك .**

ولعلك لاحظت في رسمة الدائرة أننا لم نوصل العمود الأول وبالتالي ÷ و × و – و + كلها لن تعمل لأننا لم نوصل العمود الخاص بها وعلى هذا فإنه عند الضغط على أي زر (مفتاح) من هذه المفاتيح فإنه لن يعمل وستصبح حينها قيمة m تساوي صفر .



عند الضغط على أي مفتاح من هذه المفاتيح فإنها لن تعمل لأن هذا العمود غير موصل في مشروعنا هذا وبالتالي عند الضغط علي أي مفتاح من هذه المفاتيح ستصبح قيمة m تساوي صفر

والآن بعد أن تمكنا من معرفة أي المفاتيح هي التي تم الضغط عليها يمكننا بسهولة وباستخدام جملة If مثلا أن نجعل البك ينفذ أمر معين في حالة الضغط على سويتش معين .

ثالثا : الدائرة المتكاملة 4514 (4-to-16 Line Decoders):

هذه الدائرة المتكاملة مهمة جدا في مشروعنا هذا ... فكما تعلم أنه من المفترض أن يتم التحكم في 48 ريلاي فكيف يتم ذلك وليس لدينا في البك pins (رجول) كافية لذلك . فلابد حينها من استخدام دائرة متكاملة مثل 4514 . فمن خلال هذه الدائرة تستطيع أن تتحكم في 16 خرج من خلال أربع مداخل فقط .

فالمداخل في هذه الدائرة هي : A B C D (4 مداخل)

والمخارج هي من Q0 إلى Q15 . (16 خرج)

فمثلا إذا أردت أن أخرج إشارة موجبة على الرجل رقم Q0 فإنني أرسل إلى المداخل(a,b,c,d) الأرقام التالية: 0000

وإذا أردت أن أخرج إشارة موجبة على الرجل رقم Q8 فإنني أرسل إلى المداخل (a,b,c,d) الأرقام التالية : 1000

وإذا أردت أن أخرج إشارة موجبة على الرجل رقم Q15 فإنني أرسل إلى المداخل (a,b,c,d) الأرقام التالية : 1111

وللعلم فإنه إذا تم جعل Q15 مثلا تخرج إشارة موجبة فإن بقية الرجول من Q0 إلى Q14 لن تخرج إشارة موجبة . . . وهكذا . بمعنى أنه في هذه الدائرة المتكاملة خرج واحد فقط هو الذي يخرج إشارة موجبة والبقية لا .

ستلاحظ وجود رجل في هذه الدائرة تسمى STB (يجب توصيلها بالموجب) ارجع للرسم .

وستلاحظ أيضا وجود رجل(طرف) تسمى INH وهي اختصار لكلمة inhibit وهي تعني يمنع أو يعيق ... هذا الطرف إذا تم إرسال له إشارة موجبة hi (واحد منطقي) فإن الدائرة المتكاملة4514 لن تعمل . وإذا تم إرسال له إشارة سالبة low (صفر منطقي) فإن المتكاملة تعمل .

لعلك لاحظت أننا في هذا المشروع لدينا ثلاث دوائرة متكاملة من النوع 4514 وتم توصيل المداخل ببعض a,b,c,d وهذا يعني أنه في حالة أن هذه المداخل تم إرسال إليها الإشارة التالية

0010

فإن الرجل رقم Q2 ستخرج جهد موجب (إشارة موجبة) ولكن هذا في الثلاث دوائر معاً أي أن الرجل Q2 في المتكاملة الأولى ستخرج جهد موجب وكذلك الرجل Q2 في المتكاملة الثانية ستخرج جهد موجب وكذلك الرجل Q2 في المتكاملة الثالثة ستخرج جهد موجب . وهذا غير مطلوب في مشروعنا . ولهذا فإنه عندما نريد أن نشغل الرجل Q2 فإننا نجعل الرجلINH في

المتكاملة الأولى تساوي صفر (وبالتالي ستعمل هذه المتكاملة) وسنجعل الرجلINH في المتكاملة الثانية تساوي واحد (وبالتالي لن تعمل هذه المتكاملة) وسنجعل INH في المتكاملة الثالثة أيضا بواحد (وبالتالي لن تعمل هذه المتكاملة) وهذا هو المطلوب .

ولنفرض الآن أن رقم الريلاي المراد تشغيله هو الريلاي رقم 4 (الموصل طبعا بالمتكاملة الأولى) والموصل بـ Q4 ففي هذه الحالة سنجعل المداخلa,b,c,d تساوي 0100 وسنجعل الـ INH في المتكاملة الأولى بصفر (لكي تعمل المتكاملة الأولى) وسنجله بواحد لكل من المتكاملتين الثانية والثالثة . وفي هذه الحالة فإنه من البديهي استنتاج المعادلة التالية :

الرقم الذي سنضعه على المداخلa,b,c,d = رقم الريلاي المراد تشغيله

وهذا القانون في حال أن رقم الريلاي المراد تشغيله محصور بين 0 و 15

ولكن ماذا لو أردنا أن نخرج جهد موجب على الرجل Q4 مثلا في المتكاملة الثانية لكي يعمل الريلاي رقم 20 الموصل بها . في هذه الحالة سنستخدم المعادلة البديهية التالية :

الرقم الذي سنضعه على المداخلa,b,c,d = الريلاي المراد تشغيله – 16

وعلى هذا يجب أن نجعل INH الخاص بالمتكاملة الثانية = صفر (لكي تعمل) وسنجعل INH الخاص بالمتكاملة الأولى والثالثة = واحد (وبالتالي لن يعملا).

المعادلة السابقة نطبقها فقط في حال أن رقم الريلاي المراد تشغيله محصور بين16 و31 .

أما إذا أردنا تشغيل الريلاي رقم 40 مثلا فإنه في هذه الحالة نستخدم القانون التالي

الرقم الذي سنضعه على المداخلa,b,c,d = رقم الريلاي المراد تشغيله – 32

وهذه المعادلة نطبقها في حال أن رقم الريلاي المراد تشغيله محصور بين 31 و 47

أما إذا كتب المستخدم رقم أكبر من 47 فإنه في هذه الحالة تظهر رسالة خطأ على الشاشة، ولن يعمل أي ريلاي .

والآن بعد أن تم شرح كل من LCD و Keypad و 4514 فسأذكر هذه التلميحات على الكود بصفة عامة :

عندما يضغط المستخدم على رقم معين من الكاياد وليكن واحد ثم يضغط بعد ذلك على الرقم 8 فإنك سترى على الشاشة الرقم 18 . ولكن كيف سيفهم البك أن هذا الرقم هو الرقم 18؟؟

في الكود الذي كتبته جعلت البك يخزن أول رقم يتم الضغط عليه (واحد في هذا المثال) في متغير اسمه tens و سيخزن ثاني رقم يتم الضغط عليه في متغير اسمه ones . وتكون القيمة الفعلية (التي هي 18) في متغير آخر اسمه result . ومن البديهي أن نكون المعادلة التالية:

```
result=(tens*10)+ones;
```

اقرأ هذه المعادلة من اليسار لليمين

وهذا يعني أن القيمة الفعلية تساوي 1 ضرب عشرة + 8 وبالتالي ستساوي 18 .

وهذا يعني أن القيمة الموجودة في المتغير result تساوي 18 . وهذه القيمة هي رقم الريلاي الذي نريد تشغيله في هذا المثال . وبما أن رقم الريلاي هذا 18 فإننا في هذه الحالة نستخدم القانون

```
الرقم الذي سنضعه على المداخلa,b,c,d  =  الريلاي المراد تشغيله – 16
```

اقرأ هذه المعادلة من اليمين لليسار

وبالتالي سنكتب الكود التالي :

```
if(result>=16&&result<=31)
{
result=result-16;
result.f4=1;result.f5=0;result.f6=1;
}
```

لو أن رقم الريلاي المطلوب تشغيله محصور بين 16 و 31 نفذ الأوامر التي بين القوسين التاليين .

اطرح من قيمة المتغير 16 لأننا نريد معرفة الرقم الذي سنضعه على المداخل الخاصة بالمتكاملة الثانية

ثم بعد ذلك نكتب الأمر التالي :

```
Portd=result;
```

ولعلك تتسائل ماذا يعني كل من السطرين البرمجين

```
result.f4=1;result.f5=0;result.f6=1;
```

وكذلك `Portd=result;`

سأبسط لك الأمر : الآن نحن نريد أن نشغل الريلاي رقم 18 أليس كذلك ؟؟ وبالطبع الريلاي رقم 18 هو الريلاي الموصل بـ Q2 الخاصة بالمتكاملة الثانية .

ولهذا سنضع القيمة 0010 على المداخل والتي هي(2 بالنظام العشري) ولكن كيف نحصل على (2) هذه ؟؟ بالتأكيد بأن نطرح من المتغير result القيمة 16 فتكون المحصلة هي

18- 16 =2 . وذلك باستخدام الأمر `result=result-16;`

أي أننا سنضع القيمة 0010 على المداخل ويمكننا فعل ذلك مباشرة بكتابة الأمر

```
Portd=2;
```

أو أن نكتب الأمر `portd=result;` لأن القيمة الحالية لـ result هي 2 (0010). ولكن المشكلة الآن هي أن الرجل رقم Q2 ستعمل في المتكاملة الأولى والثانية والثالثة . ونحن نريد أن نجعل المتكاملة الثانية فقط هي التي تعمل والمتكاملة الأولى والثالثة لا تعمل . فيمكننا فعل ذلك بجعل INH الخاص بالمتكاملة الثانية يساوي صفر ونجعل INH الخاص بالمتكاملة الأولى والثالثة يساوي واحد . ولعلك لاحظت في دائرة المشروع أننا وصلنا الـ INH الخاص بالمتكاملة الأولى بـ D4 و الـINH الخاص بالمتكاملة الثانية بـ D5 و الـ INH الخاص بالمتكاملة الثالثة بـ D6 . وعلى هذا

فإننا يجب أن نجعل كل من D4,D6 بواحد ونجعل D5 بصفر لكي تعمل المتكاملة الثانية فقط.

وذلك بكتابة الأوامر التالية :

```
Portd.f4=1;

Portd.f5=0;

Portd.f6=1;
```

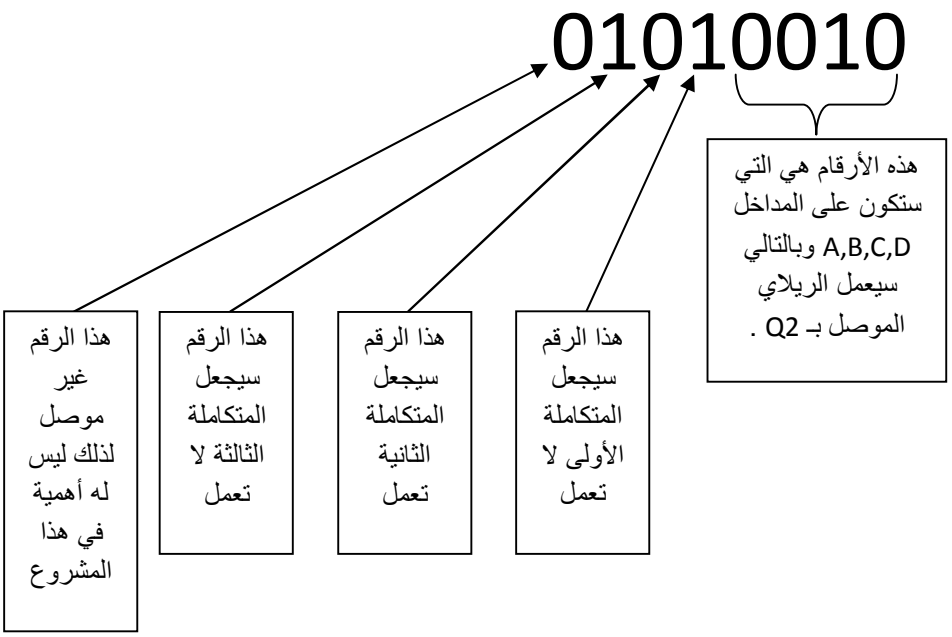
ومن الممكن أيضا أن نكتب الكود بطريقة مختلفة وهي أن نجعل ذلك على المتغير result ثم نساوي portd بالمتغير result كما يلي :

```
result.f4=1;result.f5=0;result.f6=1;
```

ثم نكتب الأمر التالي :

```
Portd=result;
```

وهكذا ستصبح قيمة portd كالتالي :



وبنفس الطريقة على بقية الإحتمالات وهي أن يكون رقم الريلاي أصغر من 16 أو رقم الريلاي محصور بين 31 و 47 . فملخص ذلك كله في الكود التالي وهو بنفس الفكرة السابقة:

```
result=(tens*10)+ones;
if(result<16){result.f4=0;result.f5=1;result.f6=1;}
else if(result>=16&&result<=31){result=result-16;result.f4=1;result.f5=0;result.f6=1;}
else if(result>=32&&result<=47){result=result-32;result.f4=1;result.f5=1;result.f6=0;}
else if(result>47){Lcd_Cmd(Lcd_Clear);lcd_out_cp("error");delay_ms(1000);return;}
portd=result;
```

الآن عند تشغيل الدائرة سينتظر البك إلى أن يدخل المستخدم الرقم الأول والذي سيخزن في المتغير tens ثم بعد ذلك ينتظر البك إلى أن يدخل المستخدم الرقم الثاني والذي سيخزن في المتغير ones ثم بعد ذلك ينتظر البك إلى أن يضغط المستخدم على الزر = .

ثم بعد ذلك يقوم البرنامج باستدعاء دالة هذه الدالة (ستقوم بتنفيذ مجموعة من الأوامر) وهي الأوامر السابقة . ولقد تم تسمية هذه الدالة بإسم on فبمجرد أن أكتب الكود التالي :

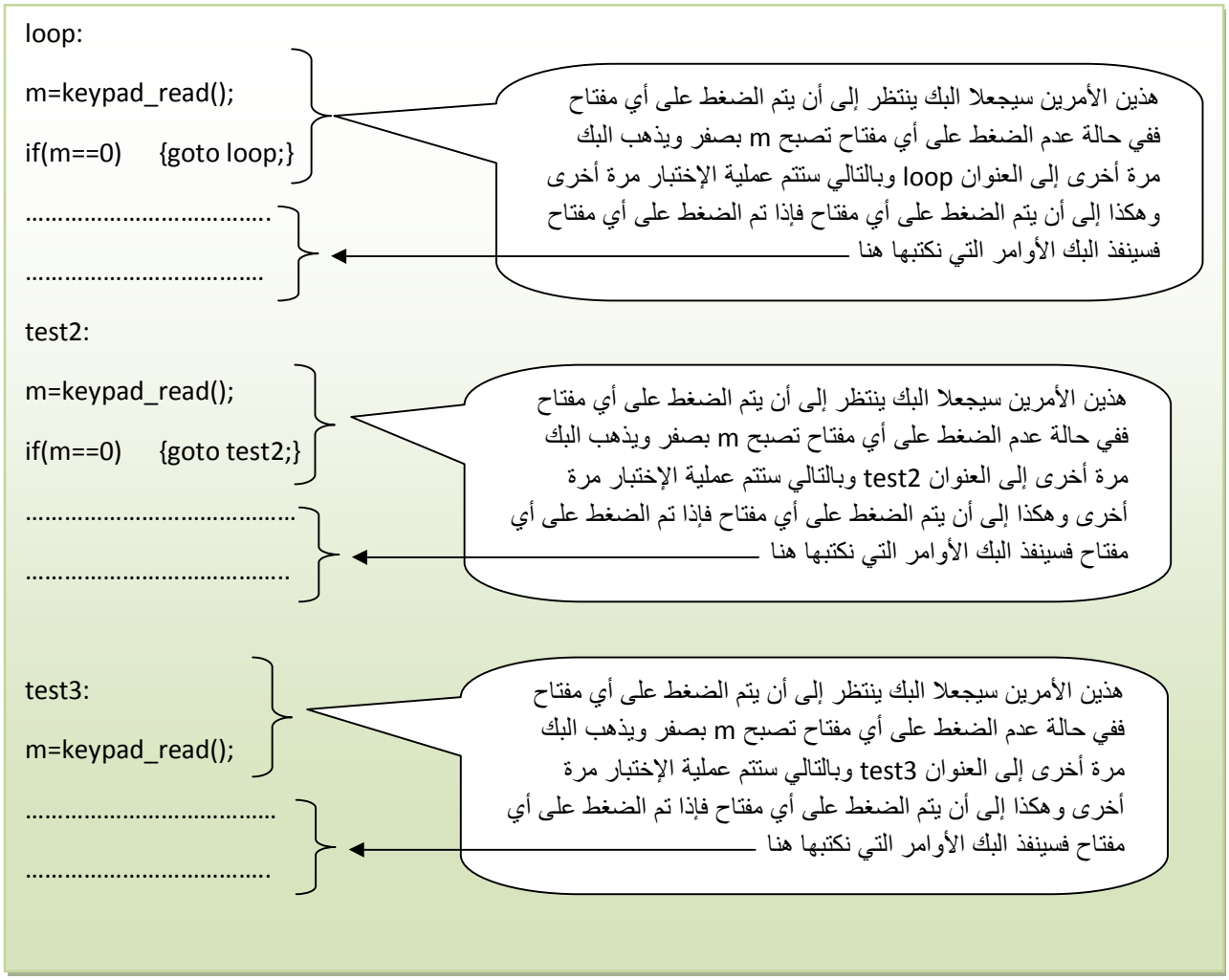
```
On();
```

فإن البرنامج يقوم بتنفيذ جميع الأوامر التي تحتويها هذه الدالة والتي هي

والآن لنتطرق إلى نقطة أخرى وهي كيف سأجعل البك يختبر أي السويتشات التي سيتم الضغط عليها ويخزن أول رقم يتم الضغط عليه في المتغير tens ثم ينتظر إلى أن يتم الضغط على رقم آخر ليخزنه في المتغير ones ثم ينتظر إلى أن يتم الضغط على علامة التساوي وحينها يستدعي الدالة on والتي بدورها ستشغل الريلاي الذي هو صاحب الرقم المطلوب.

الآن وكما تعلم عند استخدام الكايباد فإنه يمكنني أن أعرف هل تم الضغط على أي سويتش أم لا وذلك بأن أختبر قيمة المتغير m فإذا كان بصفر فهذا معناه أنه لم يتم الضغط على أي

سويتش في هذه اللحظة وإذا كان بأي رقم آخر فإن قيمة m لن تكون بصفر وستكون بقيمة أخرى تشير إلى المفتاح الذي تم الضغط عليه . والآن أريد أن أكتب كود معين يجعل البك ينتظر إلى أن يتم الضغط على أي مفتاح ثم إذا تم الضغط على أي مفتاح يقوم بتنفيذ أوامر معينة ثم ينتظر مرة أخرى إلى أن يتم الضغط على أي مفتاح وإذا تم الضغط على أي مفتاح ينفذ أوامر معينة ثم ينتظر مرة أخرى إلى أن يتم الضغط على مفتاح معين ليشغل الريلاي المطلوب . كما تلاحظ فإننا نحتاج إلى أن نختبر ثلاث مرات الأختبار الأول سميته loop والإختبار الثاني سميته test2 والإختبار الثالث سميته test3 . . وإليك الكود التالي :



من المعلوم أنه عند الضغط على على = فإن قيمة m تصبح 14 وعند الضغط على المفتاح ON/c فإن قيمة m تصبح 16 لذلك ستجدني استخدمت جمل if تختبر هل m تساوي14 أم 16 أم أنها تساوي قيمة أخرى

```
loop:           //testing the first pushing down
m=keypad_read();
if(m==0)    {goto loop;}
else if(m==14){.....}
else if(m==16){.....}
else        {.....}

test2:          //testing the second
m=keypad_read();
if(m==0)    {goto test2;}
else if(m==14){.....}
else if(m==16){..... }
else        {.....}

test3:          //testing the third pushing down
m=keypad_read();
if(m==0)    {goto test3;}
else if(m==14){..... }
else if(m==16){..... }
else        {.....}
```

هنا نكتب الأوامر التي ستنفذ في حالة تم الضغط على مفتاح يساوي ولاحظ أن هذا في الاختبار الأول أي أن المستخدم ضغط يساوي قبل أن يضغط على أي مفتاح آخر وبالتالي لابد أن أجعل البك لا يفعل شيء إلى أن يتم الضغط على مفتاح آخر وذلك بأن يذهب إلى loop

هنا نكتب الأوامر التي ستنفذ في حالة تم الضغط على مفتاح on/c ولاحظ أن هذا في الاختبار الأول أي أن المستخدم ضغط on/c قبل أن يضغط على أي مفتاح آخر وبالتالي لابد أن أجعل البك يمسح الشاشة ولا يتخذ إجراء آخر وذلك بالذهاب إلى Loop ليختبر من جديد .

هنا نكتب الأوامر التي سينفذها البك في حال أن الشروط السابقة جميعا لم تتحقق أي أن المستخدم ضغط على أي مفتاح غير يساوي وغير ON/c

وهنا يجب التنبيه لعدة نقاط مهمة وهي :

النقطة الأولى : وهي في الإختبار الثاني test2 ماذا لو ضغط المستخدم على رقم معين خلال الإختبار الأول Loop بالتأكيد سيخزن هذا الرقم في العشرات tens ثم بعد ذلك في الإختبار الثاني test2 ضغط المستخدم على علامة يساوي . هل تخيلت ما هي المشكلة الآن ؟؟ المشكلة أن الرقم تم تخزينه في العشرات وعند استدعاء الدالة on فإنه سيعمل ريلاي آخر غير المطلوب .

ولأوضح لك الأمر أكثر : سأضرب لك هذا المثال . قام أحد المستخدمين لهذه الدائرة بتشغيلها وبعد ذلك ضغط على الرقم 3 ثم بعد ذلك ضغط يساوي . وكان من المفترض أن يضغط صفر ليخزن هذا الرقم في العشرات ثم بعد ذلك يضغط على الرقم 3 ليخزن في الآحاد ثم يضغط يساوي . (وبالتالي لن تكون هناك أي مشكلة) ولكن هذا المستخدم ضغط 3 مباشرة ثم ضغط يساوي .

هنا من الناحية البرمجية يوجد عدة حلول الحل الأول هو أن أجعل البك لا ينفذ أي شيء (أثناء الإختبار الثاني) إذا ضغط المستخدم على علامة يساوي . وذلك بأن أكتب التالي:

```
test2:          //testing the second pushing down
m=keypad_read();
if(m==0)    {goto test2;}
else if(m==14){goto test2;}
else if(m==16){..................................... }
else        {.....................................}
```

في الإختبار الثاني إذا ضغط المستخدم على علامة يساوي لا تفعل شيء واختبر مرة أخرى وذلك بالذهاب إلى test2

في الحل السابق أجبرت المستخدم على طريقة معينة في كتابة رقم الريلاي وهي إذا كان رقم الريلاي أقل من 10 فإنه مضطر أن يكتب صفر ثم يكتب الرقم (مثال 03) ثم يضغط يساوي .

الحل الثاني : وأرى أنه الأفضل وهو في حالة ضغط المستخدم على الرقم ثم يساوي مباشرة

(مثال 3) فسيخزن البك الرقم 3 في الآحاد ones وليس العشرات tens ولكن كيف سأجعل البك يفعل ذلك وهو أساساً في الإختبار الأول loop قام بتخزين الرقم 3 في العشرات tens لذلك عند الضغط على يساوي في الإختبار الثاني test2 سأأمر البك بأن يجعل قيمة المتغير ones تساوي قيمة المتغير tens ثم بعد ذلك أجعل المتغير tens يساوي صفر . وهذا ما فعلته في الكود الذي كتبته لهذا المشروع .

```
test2:          //testing the second pushing down
m=keypad_read();
if(m==0)    {goto test2;}
else if(m==14){goto test2;}
else if(m==16){ ones=tens;tens=0;on();goto test3;.}
else        {.....................................}
```

اجعل الرقم الذي تم إدخاله (وليكن 3 مثلا) والذي تم تخزينه في tens اجعله في الآحاد ones ثم بعد ذلك اجعل tens=0 ثم استدعي الدالة on والتي ستشغل الريلاي المطلوب ثم اذهب إلى test3 .

على المبرمج أن ينتبه لمثل هذه الأشياء التي تساعد المستخدم وتجعل استخدام الدائرة أكثر سهولة فيجب عليك أن تضع في بالك الإحتمالات التي يمكن أن تحدث . ومن هذه الإحتمالات أيضا احتمال أن يضغط المستخدم على رقم أكبر من 47 كأن يضغط مثلا على 74 ففي هذه الحالة يفضل أن تظهر رسالة خطأ ولا تجعل البك يشغل أي ريلاي كما فعلت ذلك في الكود الخاص بالدالة on :

```
else if(result>47){Lcd_Cmd(Lcd_Clear);lcd_out_cp("error");delay_ms(1000);return;}
```

الأمر return سجعل البك يعود إلى الدالة الرئيسية (وبهذا يكون انتهى من تنفيذ الدالة on ولن يكمل الأوامر التي في الدالة . وسيعود للدالة الرئيسية لكي يختبر المفاتيح من جديد .

ولتشغيل ريلاي آخر في هذه الحالة لابد من الضغط على ON/c ثم رقم الريلاي ثم يساوي .

النقطة الثانية : وكما أذكر ذلك دائما في شروحاتي وهو أنه يجب التنبه إلى أن البك ينفذ الأوامر بسرعة كبيرة جدا . فلو لم أنتبه إلى ذلك الأمر سيحدث العديد من المشاكل من هذه المشاكل أنه عند تشغيل المستخدم للدائرة ثم الضغط على رقم 6 مثلا فإنه من المفترض أن يخزن البك هذا الرقم في العشرات tens ثم بعد ذلك ينتظر إلى أن يضغط المستخدم على مفتاح آخر . ولكن هذا ربما لا يحدث وتجد أن البك اختبر المفتاح الذي ضغط عليه المستخدم والذي هو 6 فيقوم بتخزين الرقم 6 في tens بسرعة ثم ينتقل للإختبار الثاني test2 (هذا كله في غاية السرعة أي في الوقت الذي ضغط فيه المستخدم على الرقم 6 ولم يرفع يده بعد) وبالتالي سيخزن البك الرقم 6 أيضا في ones . وهذا ما لا يريده المستخدم فالمستخدم يريد مثلا كتابة الرقم 6 فقط . ويفاجأ المستخدم حينها بأن الشاشة ظهر عليها الرقم 66 بدلا من 6 . ولحل هذه المشكلة يجب أن أجعل البك ينتظر لمدة معينة بعد أن يضغط المستخدم . ولتكن ثانية مثلا أو نصف ثانية . كما في الأمر التالي :

```
else        {ones=number2[m];lcd_chr_cp(number[m]);delay_ms(500);}
```

الأمر delay فائدته أن ينتظر البك ولا يذهب مباشرة للإختبار الثاني test2 حتى لا يخزن البك الرقم الذي يدخله المستخدم في الآحاد وفي العشرات معا دون أن يضغط المستخدم على الرقم مرتين . وهذه نتيجة غير مطلوبة بالنسبة للمستخدم .

النقطة الثالثة : لعلك لاحظت في كود المشروع الذي كتبته هو أنني عرفت مصفوفتين هما :

```
char number[]={'x',' ','9','8','7','x','6','5','4','x','3','2','1','x','x','0','x'};
char number2[]={'x',' ',9,8,7,'x',6,5,4,'x',3,2,1,'x','x',0,'x'};
```

المصفوفة الأولى اسمها number والمصفوفة الثانية اسمها number2 . ولعلك لاحظت من شرحي عن LCD الفائدة من استخدام المصفوفة number . ففائدتها تظهر عندما أريد أن أكتب رقم معين على الشاشة وليكن 8 مثلا فإنه من الممكن أن أكتب الأمر التالي :

```
lcd_chr _CP('8');
```

أو من خلال المصفوفة number بكتابة الأمر التالي :

```
lcd_chr _CP(number[3]);
```

وربما يتبادر إلى ذهنك لماذا تم ترتيب عناصر المصفوفة بهذا الشكل ؟؟ الإجابة : من أجل أن أربط بين الرقم الذي يتم الضغط عليه من الكاييباد وبين ما يظهر على الشاشة وسأوضح لك الأمر أكثر . الآن عندما يضغط المستخدم على الرقم 8 من الكاييباد .فإنه في هذه الحالة ستصبح قيمة m تساوي 3 (ارجع لشرح الكاييباد keypad) وبما أن المستخدم ضغط على الرقم 8 فإنه من المفترض أن يظهر على الشاشة الرقم 8 فيمكننا أن نكتب الأمر التالي :

```
If(m==8) lcd_chr _CP('8');
```

ولكني إذا اتبعت هذه الطريقة فسأكتب الكثير والكثير من جمل If لكي تختبر قيمة m لو كانت 1 أو 2 أو ..... أو 16 .

لذلك فكرت في فكرة أخرى وهي أن أنشئ مصفوفة هذه المصفوفة تحتوي على أرقام ورتبت عناصر هذ المصفوفة بطريقة معينة . فعند الضغط على الرقم 8 مثلا فستكون قيمة m تساوي 3 لذلك يجب أن يكون العنصر رقم 3 في المصفوفة number هو الرقم '8' . وعندها يمكنني أن أكتب الأمر التالي :

```
lcd_chr_cp(number[m]);
```

عند الضغط على أي مفتاح ستتغير قيمة m فمثلا عند الضغط على المفتاح 8 ستصبح قيمة m تساوي 3 وعندها عندما أكتب الأمر
lcd_chr_cp(number[m]);

فكأنني كتبت الأمر التالي

lcd_chr_cp(number[3]);

لأن m في هذه الحالة تساوي 3

ومن السهل استنتاج أنه عند الضغط على المفتاح 0 ستكون قيمة m تساوي 15 وبالتالي يجب أن أضع الرقم '0' بحيث يكون ترتيبه في المصفوفة هو العنصر الخامس عشر

عند الضغط على المفتاح 1 ستكون قيمة m تساوي 12 وبالتالي يجب أن أضع الرقم '1' بحيث يكون ترتيبه في المصفوفة هو العنصر الثاني عشر

عند الضغط على المفتاح 2ستكون قيمة m تساوي 11 وبالتالي يجب أن أضع الرقم '2' بحيث يكون ترتيبه في المصفوفة هو العنصر الحادي عشر

عند الضغط على المفتاح 3 ستكون قيمة m تساوي 10 وبالتالي يجب أن أضع الرقم '3' بحيث يكون ترتيبه في المصفوفة هو العنصر العاشر

عند الضغط على المفتاح 4 ستكون قيمة m تساوي 8 وبالتالي يجب أن أضع الرقم '4' بحيث يكون ترتيبه في المصفوفة هو الثامن

عند الضغط على المفتاح 5 ستكون قيمة m تساوي 7 وبالتالي يجب أن أضع الرقم '5' بحيث يكون ترتيبه في المصفوفة هو العنصر السابع

عند الضغط على المفتاح 6 ستكون قيمة m تساوي 6 وبالتالي يجب أن أضع الرقم '6' بحيث يكون ترتيبه في المصفوفة هو العنصر السادس

عند الضغط على المفتاح 7 ستكون قيمة m تساوي 4 وبالتالي يجب أن أضع الرقم '7' بحيث يكون ترتيبه في المصفوفة هو العنصر الرابع

عند الضغط على المفتاح 8 ستكون قيمة m تساوي 3 وبالتالي يجب أن أضع الرقم '8' بحيث يكون ترتيبه في المصفوفة هو العنصر الثالث

عند الضغط على المفتاح 9 ستكون قيمة m تساوي 2 وبالتالي يجب أن أضع الرقم '9' بحيث يكون ترتيبه في المصفوفة هو العنصر الثاني

أما العناصر = و ON/c فإنه عند الضغط عليها ستنفذ أوامر أخرى ولن تعرض على الشاشة قيمة العنصر المقابل لها في المصفوفة (لأننا جعلنا البك يختبر هذين المفتاحين أولا قبل اختبار أي مفتاح آخر وفي حالة عدم الضغط عليهما سينفذ البك ما تحتويه جملة else التي تليهما)

```
test2:          //testing the second pushing down
m=keypad_read();
if(m==0)    {goto test2;}
else if(m==14){goto test2;}
else if(m==16){ ones=tens;tens=0;on();goto test3;.}
else        {..... lcd_chr_cp(number[m]);.....}
```

اختبر هل تم الضغط على المفتاح ذو علامة التساوي
إذا حدث ذلك نفذ الأمر الذي بين القوسين { }

اختبر هل تم الضغط على المفتاح ON/c
إذا حدث ذلك نفذ الأمر الذي بين القوسين { }

إذا لم يتم تحقق الشرطين السابقين أي لم يتم الضغط على أي من المفتاحين السابقين يساوي أو ON/c نفذ الأوامر التي بين القوسين التاليين { } والذين من ضمنهم الأمر الذي سيعرض الرقم على الشاشة

بالنسبة لبقية المفاتيح الأخرى فهي غير مهمة لأنها غير موصلة بالبك ويمكننا أن نضع مكانها أي علامة أو رقم مثل 'x' أو ' ' . كما تريد فلن نستفيد منها في هذا المشروع .

والآن لنتحدث عن المصفوفة الثانية والتي هي number2 :

```
char number[]={'x',' ','9','8','7','x','6','5','4','x','3','2','1','x','x','0','x'};
```

ستجدني في الكود جعلت قيمة ones أو tens تساوي عنصر معين في المصفوفة number2 فلماذا استخدمت مصفوفة أخرى ولم أستخدم المصفوفة الأولى number ؟؟؟

من المفترض عند الضغط على المفتاح 7 مثلا فسيظهر على الشاشة الرقم سبعة وسيصبح قيمة المتغير tens (العشرات) يساوي 7 لذلك من البديهي أن نفكر ونقول لماذا لا نستخدم الأمر

```
Tens=number[m];
```

ولكن إذا فعلنا هذا فسيحدث مشكلة كبيرة وهي خطأ في الحسابات .. لماذا ؟؟ لأن الرقم 7 الموجود في المصفوفة number له العلامة ' ' (مثال '7') هذه العلامة تحول هذا الرقم إلى ما يسمى بالأسكي كود ASCI Code . وبالتالي ستكون قيمة الرقم 7 هي 55 انظر لجدول الأسكي كود التالي :

NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
SPC	!	"	#	$	%		'	(	)	*	+	,	-	.	/
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

جزء من ASCI Code وستجد الرقم سبعة أنه يتم تشفيره إلى الرقم 55

وبالتالي لن تكون قيمة المتغير tens بسبعة ولكنها ستصبح بـ 55 . لذلك يجب علينا أن ننشئ مصفوفة أخرى هذه المصفوفة لها نفس ترتيب المصفوفة number ولكن الأرقام 0 و1 و2 و ..9 لا تحتوي على العلامة ' ' لكي لا تتحول إلى الأسكي كود . وبذلك نكتب الأمر التالي .

```
Tens=number2[m];
```

حيث أن المصفوفة الجديدة التي أنشأنها اسمها number2 . ولا ننسى أن نقوم بتعريفها في بداية البرنامج :

```
char number2[]={'x',' ',9,8,7,'x',6,5,4,'x',3,2,1,'x','x',0,'x'};
```

طبعا أي عنصر عدا صفر و 1 و 2 و ...9 ليس له أي أهمية لذلك يمكننا أن نضعه بأي قيمة سواءاً بعلامة ' ' أو بدونها

إذن المصفوفة الأولى number نستخدمها في حالة أننا نريد أن نعرض رقم معين على الشاشة حيث أن هذا الرقم لابد أن يكون بالأسكي كود لكي يتم عرضه على LCD لذلك لابد من علامة ' ' .

أما المصفوفة الثانية Number2 نستخدمها في حالة أننا نريد أن نخزن الرقم الذي يدخله المستخدم سواءا في المتغير tens أو ones ولابد أن **لا** يكون بالأسكي كود أي **بدون** علامة ' '

والآن بعد أن شرحت أهم النقاط التي في الكود انظر للكود مرة أخرى بل وعدة مرات وحاول أن تفهمه جيدا وإذا استشكل عليك أمر ما لا تتردد بأن تسأل ذوي الخبرة ويمكنك زيارة صفحة المشروع الأصلية على موقع القرية الإلكترونية على هذا الرابط حيث يمكنك أن تضع أسئلتك هناك http://www.qariya.com/vb/showthread.php?t=41812 وستجد إن شاء الله من يرد على تساؤلاتك ... أويمكنك أن تراسلني على الإيميل ..

وكما أنصح من ليس ليده الخبرة في برمجة المايكروكنترولر الإطلاع **أولا** على دورة المايكروكنترولر التي أقدمها وهي بعنوان

أسرع طريق لإحتراف برمجة الـ PIC بإستخدام MikroC فستساعدك كثيراً إن شاء الله على فهم الكود .

ستجدها في هذا الرابط http://www.qariya.com/vb/showthread.php?t=40760

ملحوظة : ربما تكون هناك طرق أفضل لكتابة هذا الكود .. لذلك إذا كان لديك أي إقتراح أو تطوير للكود الخاص بهذا المشروع فيمكنك أن تضع ذلك في الصفحة الخاصة به في منتدى القرية الالكترونية ..

في النهاية إذا استفدت من هذا المشروع ولو استفادة بسيطة فلا تنساني أخي الفاضل (أو أختي الفاضلة) بدعوة في ظهر الغيب فهذا كل ما أتمناه منك وأسأل الله العلي العظيم أكرم الأكرمين أن يرزقنا الإخلاص وأن يثيبنا على أعمالنا الصالحة وأن يتجاوز عن أخطاءنا جميعا وأن يرزقنا وإياكم العلم النافع والعمل الصالح .

وهذه دعوة مني للجميع لنشر العلم وبذل الجهد والوقت من أجل ذلك الهدف حتى ولو كانت المعلومات بسيطة (فربما لا يعرفهاالكثير) واعلموا أن من يشارك في هذا الأمر ويشارك في نشر العلم ويحاول توصيله للجميع فإنه إن شاء الله يعتبر من المشاركين في نهضة هذه الأمة . وأعتقد أننا نحن العرب بحاجة إلى بذل الكثير من الجهد في عدة مجالات في مجال المشاريع مفتوحة المصدر خصوصاً وفي مجال الإلكترونيات والبرمجة عموماً.

أخوكم :أحمد سمير فايد