

المتحكمات الصغيرة و النظم المضمنة

محاضرة 4

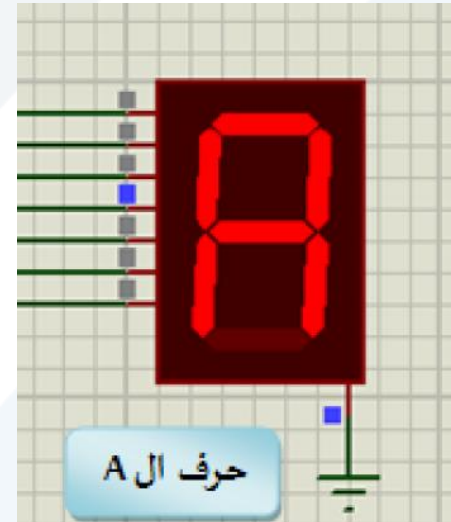
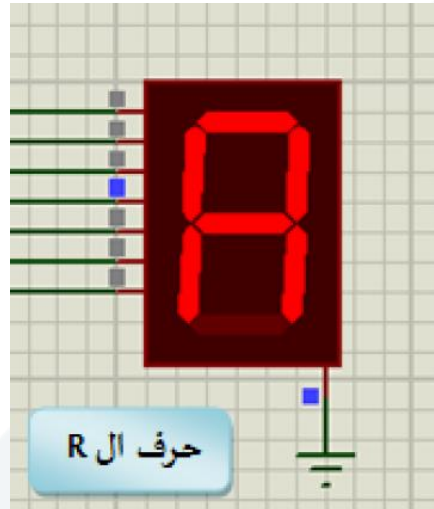
التعامل مع شاشات الـ LCD

د. فادي متوج

مشاكل ال 7-Segment

درسنا في المحاضرة الماضية ال 7-Segment وتعلمنا كيفية التعامل معها وتوصيلها وفائدتها ولكن هناك مشاكل في ال 7-Segment ينبغي علينا معرفتها :

1- لا تمكننا ال 7-Segment من عرض كل الحروف، فعلى سبيل المثال إذا حاولنا إضاءة ليدات معينة بحيث يظهر الحرف W فلن نستطيع ذلك، وأيضا هناك من الحروف التي تبدو متماثلة عند عرضها على ال 7-Segment فمثلا الحرف A والحرف R إذا أردنا عرضهم فسيظهرون بصورة طبق الأصل من بعضهما البعض كما بالشكل:



مشاكل ال-7 Segment

2- إذا أردنا أن نقوم بعرض كلمة أو جملة بال 7-Segment فهذا يتطلب عدد من ال-7 Segment يساوي عدد الحروف الموجودة في الجملة، فمثلا لو أردنا أن نقوم بعرض كلمة Display وبافتراض عدم وجود المشكلة الأولى وأننا نستطيع عرض أي حرف بال-7 Segment فإننا سنحتاج إلى سبعة منها، كل واحدة لعرض حرف من الكلمة وهذا ينتج عنه صعوبة في عمل البرنامج وصعوبة في عمل الهاردوير.



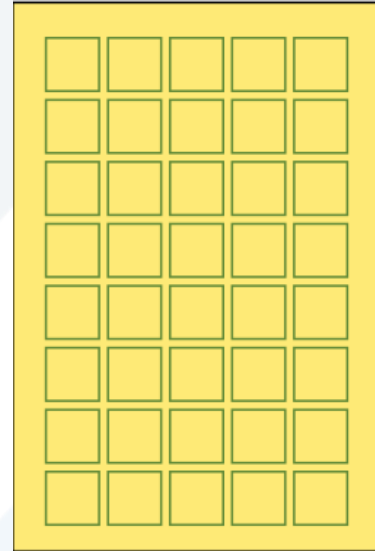
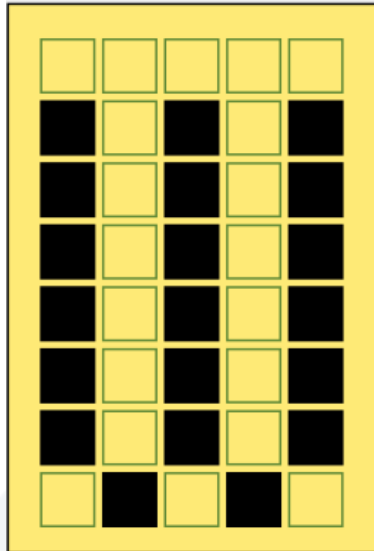
جامعة
المنارة
MANARA UNIVERSITY

شاشة LCD

ولذلك كان لابد من وضع حلول لهذه المشاكل وهي ما تمثلت في شاشة LCD

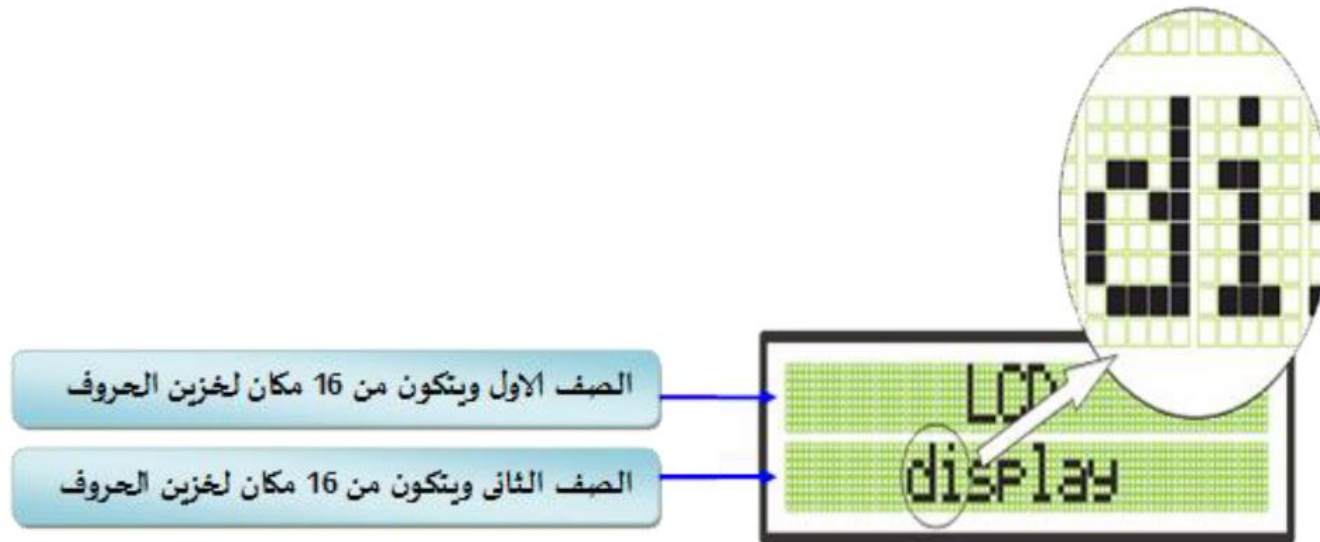
بالنسبة للمشكلة الأولى فتم التغلب عليها بزيادة عدد الليدات وتصغير حجمها وترتيبها بطريقة منتظمة كما بالشكل التالي بحيث تسمح بعرض أي حرف لم يكن متاح في الت 7-Segment، وبالتالي لنقوم بعرض نفس الحرف W كمثال فلن يكون ذلك صعبا، ويمكننا أيضا تجربة أي حرف آخر وتخيل شكله ومدى إتاحته من عدمه

...



شاشة LCD

- بالنسبة للمشكلة الثانية المتمثلة في عرض كلمة أو جملة فهذه المشكلة تم حلها أيضا عن طريق وضع عدد كبير من أجزاء مصفوفة الليدات الموجودة في الصورة السابقة والمستخدمة لعرض حرف واحد وبذلك يمكن عرض مجموعة حروف متجاورة لتكون جملة أو كلمة داخل الـ LCD
- أغلب الـ LCDs تحتوي على أكثر من صف وكل صف يتكون من أكثر من مكان لعرض الحروف ...



أنواع الـ LCD

Character LCD

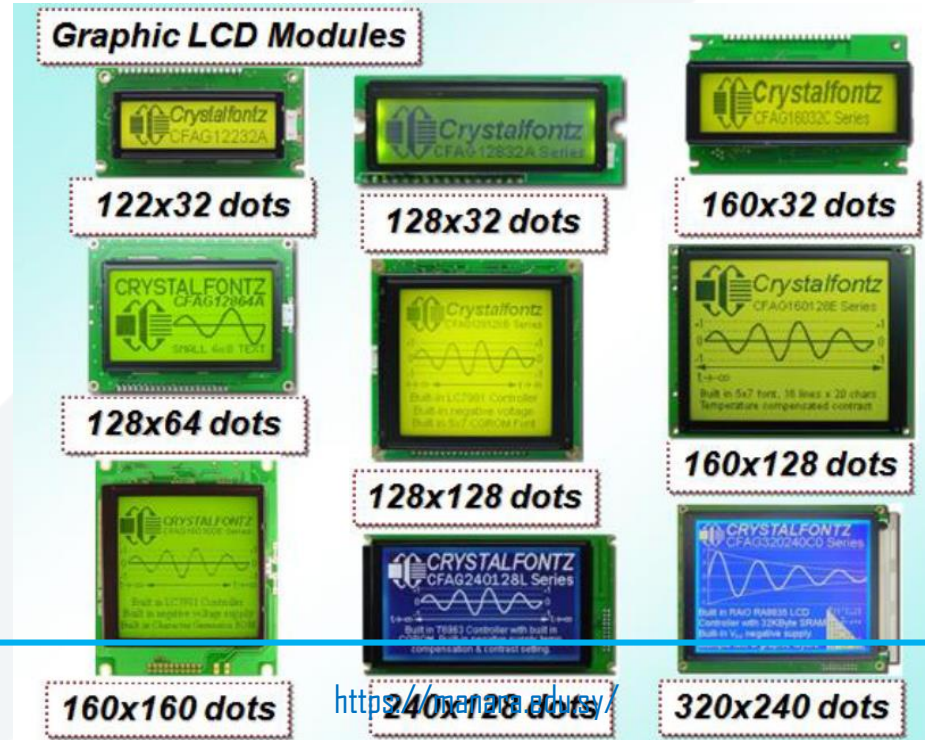
النوع الذي تحدثنا عنه منذ قليل يسمى Character LCD حيث يستخدم لعرض الحروف، وله أشكال مختلفة يختلف كل شكل عن الآخر في عدد السطور وعدد الحروف داخل كل سطر، فمثلا توجد أنواع بها صف واحد وهذا الصف يستوعب ثمانية أحرف ويوجد أنواع تحتوي على صفين وكل صف يستوعب ثمانية أحرف وغير ذلك كما هو مبين



أنواع الـ LCD

Graphical LCD

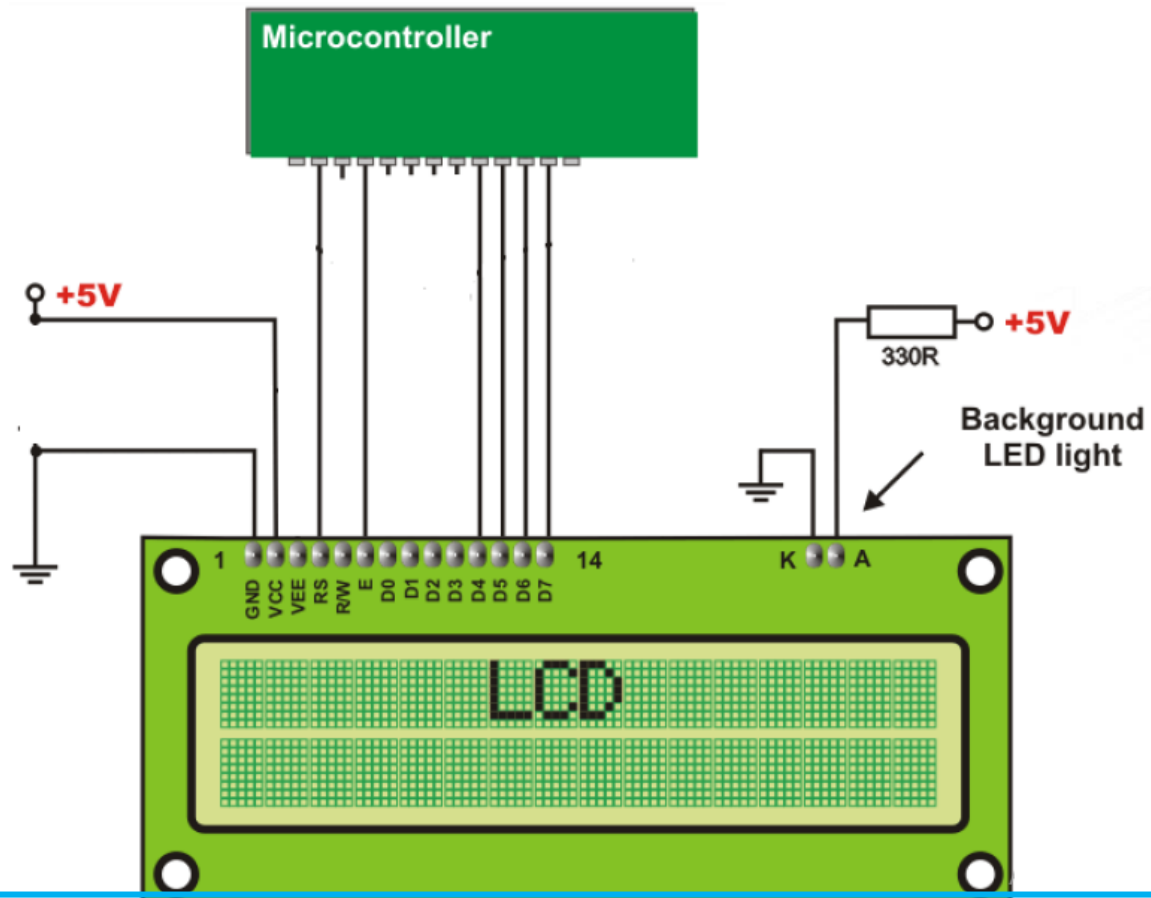
يستخدم لعرض الرسومات البيانية، وهو لا ينقسم إلى أجزاء كالنوع السابق وإنما يشبه في طريقة عمله شاشة الكمبيوتر حيث تكون عبارة عن وحدة واحدة بها عدد كبير من الـ Pixels مرتبة في صفوف وأعمدة ويمكنك إظهار أي شكل تريده عن طريقة التحكم في إضاءة أو إطفاء أي بيكسل وبترتيب البيكسل المضاءة والبيكسل المطفأة نحصل على الشكل المطلوب رسمه، ويوجد منها أحجام أيضا ولكن لا يتم التعبير عنها بعدد السطور وعدد الحروف في السطر وإنما بعدد البيكسل الأفقي مضروبا في عدد البيكسل الرأسي كما في الشكل التالي:





جامعة
المنارة
MANARA UNIVERSITY

توصيل الـ LCD بالميكروكنترولر



توصيل الـ LCD بالميكروكنتروлер

توجد في غالبية الشاشات 16 رجل مرقمين من واحد إلى 16 والذي نحتاجه فقط منها لتوصيله بالميكرو هي ستة أرجل و هي الآتية:

1- يوجد أربعة أرجل لنقل بيانات العرض والأوامر من الميكرو للـ LCD وأرقامهم هي 11 و 12 و 13 و 14 ويكون مكتوب عليهم في شاشة الـ LCD الآتي: D7 , D6 , D5 , D4

2- رجلين آخرين:

• الرجل RS: وتستخدم لكي يخبر الميكرو الشاشة عن طريقها ما إذا كان سيتم نقل أمر أم سيتم نقل بيانات الآن، والأوامر مثل (مسح الشاشة) والبيانات المقصود بها الكلام المرسل للشاشة لتعرضه.

• الرجل EN: وتستخدم لجعل الـ LCD مستعدة لاستقبال البيانات وإظهارها

3- ما سبق من أرجل هو ما يخص التوصيل بين الميكرو والـ LCD. ولكن هناك أطراف أخرى للـ LCD لا بد من توصيلها ولكن ليس بالميكرو وهم:

• الرجل رقم 2: ويوصل بالخمسة فولت.

• الأرجل 1 و 3 و 5: ويوصلوا جميعا بالأرضي.

• الرجلين 15 و 16: ويستخدمان لإضاءة ليد موجود في الشاشة، حيث يفيد هذا الليد حتى نتمكن من رؤية البيانات المعروضة جيدا، يتم توصيل الرجل 15 على الخمسة فولت والرجل 16 على الأرضي.

توابع الـ mikroC المستخدمة مع الشاشات

- توجد مكتبة تضم التوابع الخاصة بالتعامل مع شاشات الـ LCD
 - تتعامل هذه المكتبة مع الشاشات وبالتالي سيكون فيها توابع تقوم بإظهار حروف أو قيم على الشاشة كدرجة الحرارة مثلا، وتابع آخر يقوم بمسح الشاشة، وتابع آخر يقوم بتحريك الكلام الموجود على الشاشة لليمين أو لليسار ... إلخ.
 - تستخدم هذه التوابع داخل برنامج الميكرو بالطبع، ولكن السؤال هنا: التابع الذي يقوم بإرسال القيمة لتظهر على LCD مثلا سيقوم بإرسالهم على أي منفذ؟؟؟ هل المنفذ PORTB أم المنفذ PORTC أم غيرهما؟؟ بالطبع على المنفذ المتصل بالشاشة، ولكن كيف تعرف هذه التوابع هذا المنفذ؟؟؟
- :

المجموعة الأولى من المتغيرات

لنتذكر أنه توجد ستة أرجل توصل بين الشاشة والميكرو منهم أربعة فقط لنقل الأوامر والمعطيات وبالتالي توجد ستة متغيرات كل متغير يقابل رجل معينة من أرجل الشاشة وهم:

اسم المتغير	الطرف المختص به في الشاشة
LCD_RS	مختص بالطرف RS
LCD_EN	مختص بالطرف EN
LCD_D4	مختص بالطرف D4
LCD_D5	مختص بالطرف D5
LCD_D6	مختص بالطرف D6
LCD_D7	مختص بالطرف D7

المجموعة الأولى من المتغيرات

ولإعلام الميكرو بالأرجل المتصلة بالشاشة نقوم بوضع أسمائها في هذه المتغيرات عن طريق مجموعة أوامر مهمة جدا كالآتي:

```
sbit LCD_RS at RC2_bit;
```

ومن هذا الأمر تعرف التوابع أن الرجل RS الموجود في الشاشة متصل بالرجل رقم 2 في المخرج C ، وماذا عن باقي الأطراف؟ نفس الطريقة:

```
sbit LCD_EN at RC3_bit;
```

هذا الأمر تعرف منه التوابع أن الطرف EN الموجود في الشاشة متصل بالرجل رقم 3 في المخرج C، وماذا عن باقي الأرجل التي تستخدم في نقل البيانات والأوامر؟

```
sbit LCD_D4 at RC4_bit;
```

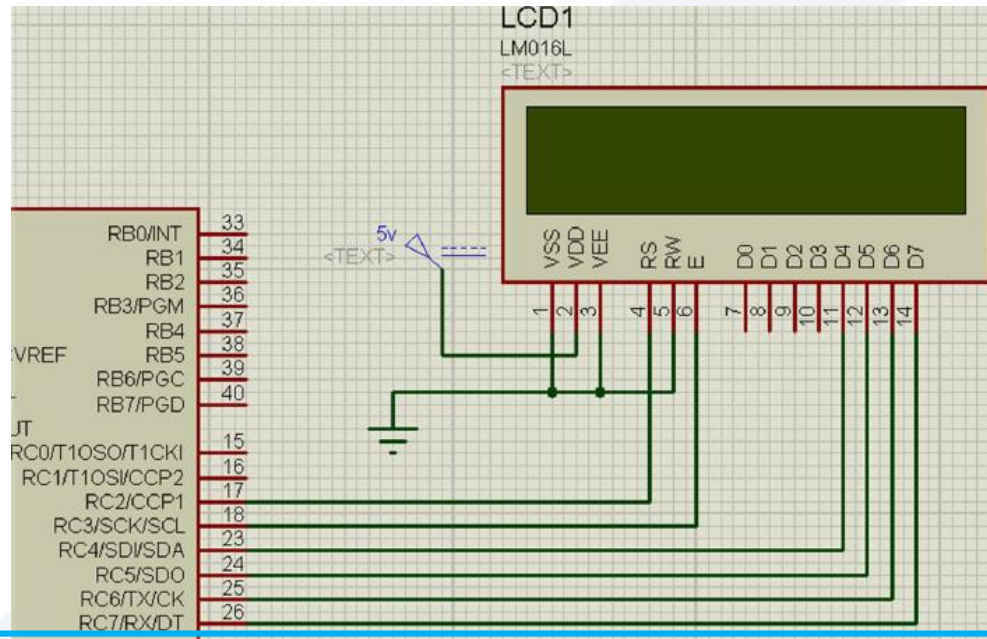
وهذا معناه أن الرجل المسماة D4 في الشاشة وهي الرجل رقم 11 فيها متصلة بالرجل رقم 4 في المخرج C، وهكذا باقي الأرجل...

المجموعة الأولى من المتغيرات

```
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;
```

وبالتالي تكون الستة أوامر كالاتي:

ملاحظة: هذه الأوامر يتم كتابتها في بداية كل برنامج يتعامل مع شاشة LCD وقبل التابع الرئيسي له.





جامعة
المنارة
MANARA UNIVERSITY

المجموعة الثانية من المتغيرات

نتذكر أننا كنا سابقا قبل أن نقوم بإخراج قيم على رجل من أرجل الميكرو لا بد أن نحدد اتجاه المعطيات على هذه الرجل باستخدام الأمر TRIS ... وبالتالي لا بد أيضا من تحديد اتجاه المعطيات على الأرجل المستخدمة مع الشاشة ... هذا أيضا سيكون بستة أوامر مشابهة للأوامر السابقة وهي كالآتي:

```
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;
```

نلاحظ في هذه الأوامر أننا نتعامل مع نفس الأرجل التي تعاملنا معها في الستة أوامر السابقة وهذا مهم جدا



المجموعة الثانية من المتغيرات

```
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;
```

لاحظ

```
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;
```

... وعندئذ تصبح الأوامر كلها بالشكل الآتي:

التوابع

الآن حان الوقت لتتعرف على التوابع التي تستخدم مع الشاشة :

```
Lcd_Init ();
```

التابع الأول:

ويتم كتابتها داخل التابع الرئيسي وقبل الـ while وفائدتها تجهيز الموديول الذي يتعامل مع الشاشة داخل الميكرو ...

التابع الثاني:

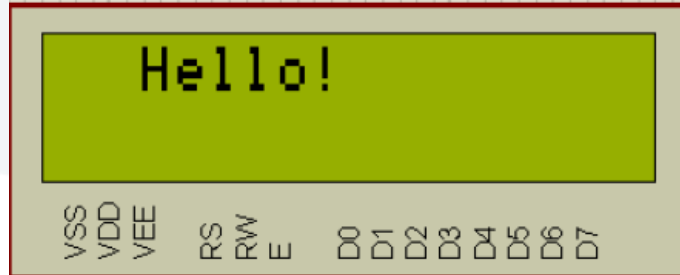
وهو تابع يستخدم لإخراج سلسلة حرفية أو جملة على الشاشة وهي كالآتي:

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

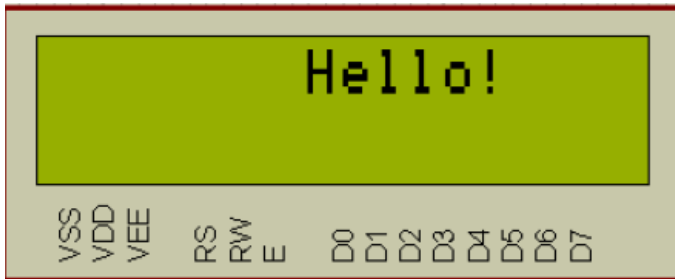
رقم الصف

رقم العمود

الكلمة التي ستظهر



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



```
Lcd_Out(1,8,"Hello !");
```



```
Lcd_Out(2,8,"Hello !");
```

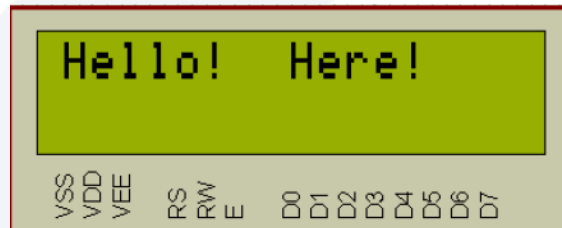
التابع الثالث:

لفهم هذا التابع نذكر مثال بسيط نفهمه من خلاله ... في برنامج الورد (word) عندما كنا نكتب مجموعة كلمات تجد انه بعد آخر حرف يوجد Cursor يظهر ويختفى ... فاذا أردت أن تكمل الكتابة فسيظهر ما ستكتبه بعده، لكن ماذا إذا كنت تريد أن تكتب في مكان آخر غير هذا السطر الذي تقف عنده لابد أن تذهب بالماوس وتضغط في المكان الذي تريد أن تكتب به وعندها يظهر ال Cursor في هذا المكان ومن ثم تبدأ الكتابة ... وبالمثل فإن هذا التابع يستخدم للكتابة عند آخر حرف انتهينا منه وبالتالي فلن نحتاج أن نقوم بتحديد الصف والعمود لها:

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

وبالتالي لو كتبنا الأمرين الآتيين

```
Lcd_Out (1,1,"Hello! ");  
Lcd_Out_Cp ("Here!");
```



سيكون الخرج كالاتي



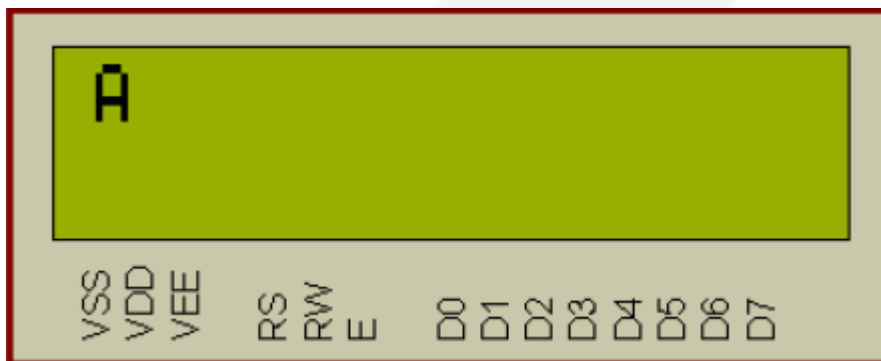
جامعة
المنارة
MANARA UNIVERSITY

التتابع

التابع الرابع:

وهو تابع يستخدم لإظهار حرف واحد على الشاشة، ويأخذ نفس صيغة التابع الذي يخرج السلسلة كآتي:

```
Lcd_Chr (1, 1, "A") ;
```



سيكون الخرج كآتي

وبالطبع لو غيرنا الصف أو العمود في هذا التابع سيتغير مكان ظهور الحرف.

التابع الخامس:

وهذا هو التابع الأخير الذي سنتعرض له وهو التابع الذي يستخدم لنقل أوامر للشاشة لكي تقوم بتنفيذها ... ويكون بالشكل الآتي:

```
Lcd_Cmd ( يكتب الأمر هنا ) ;
```

حيث نقوم بكتابة الأوامر بين أقواس هذا التابع، فمثلا لمسح الشاشة نكتب الآتي:

```
Lcd_Cmd ( _LCD_CLEAR ) ;
```

أيضا الأمر الذي يستخدم في إلغاء ال Cursor

```
Lcd_Cmd ( _LCD_CURSOR_OFF ) ;
```





جامعة
المنارة
MANARA UNIVERSITY

شكل أي برنامج يتعامل مع الشاشة

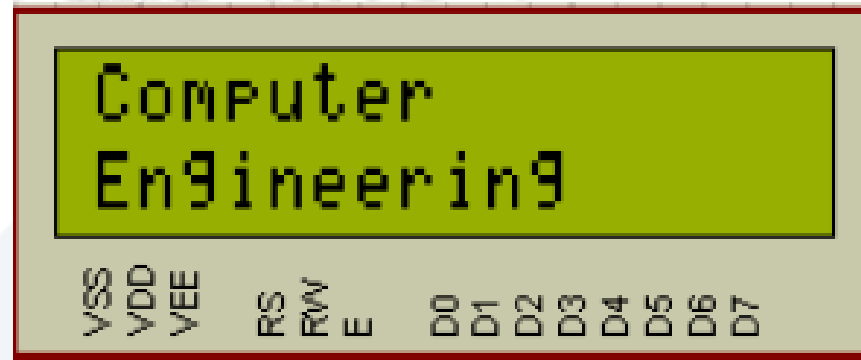
```
sbit LCD_RS at RD2_bit;  
sbit LCD_EN at RD3_bit;  
sbit LCD_D4 at RD4_bit;  
sbit LCD_D5 at RD5_bit;  
sbit LCD_D6 at RD6_bit;  
sbit LCD_D7 at RD7_bit;
```

```
sbit LCD_RS_Direction at TRISD2_bit;  
sbit LCD_EN_Direction at TRISD3_bit;  
sbit LCD_D4_Direction at TRISD4_bit;  
sbit LCD_D5_Direction at TRISD5_bit;  
sbit LCD_D6_Direction at TRISD6_bit;  
sbit LCD_D7_Direction at TRISD7_bit;
```

```
void main()  
{  
    Lcd_Init();  
    Lcd_Cmd(_LCD_CURSOR_OFF);  
  
    while(1)  
    {  
  
    }  
}
```

المثال التطبيقي الأول

نريد أن نقوم بتنفيذ مشروع يعرض كلمة Computer على السطر الأول من الشاشة ويعرض كلمة Engineering على السطر الثاني كما يظهر في هذه الصورة.





جامعة
المنارة

MANARA UNIVERSITY

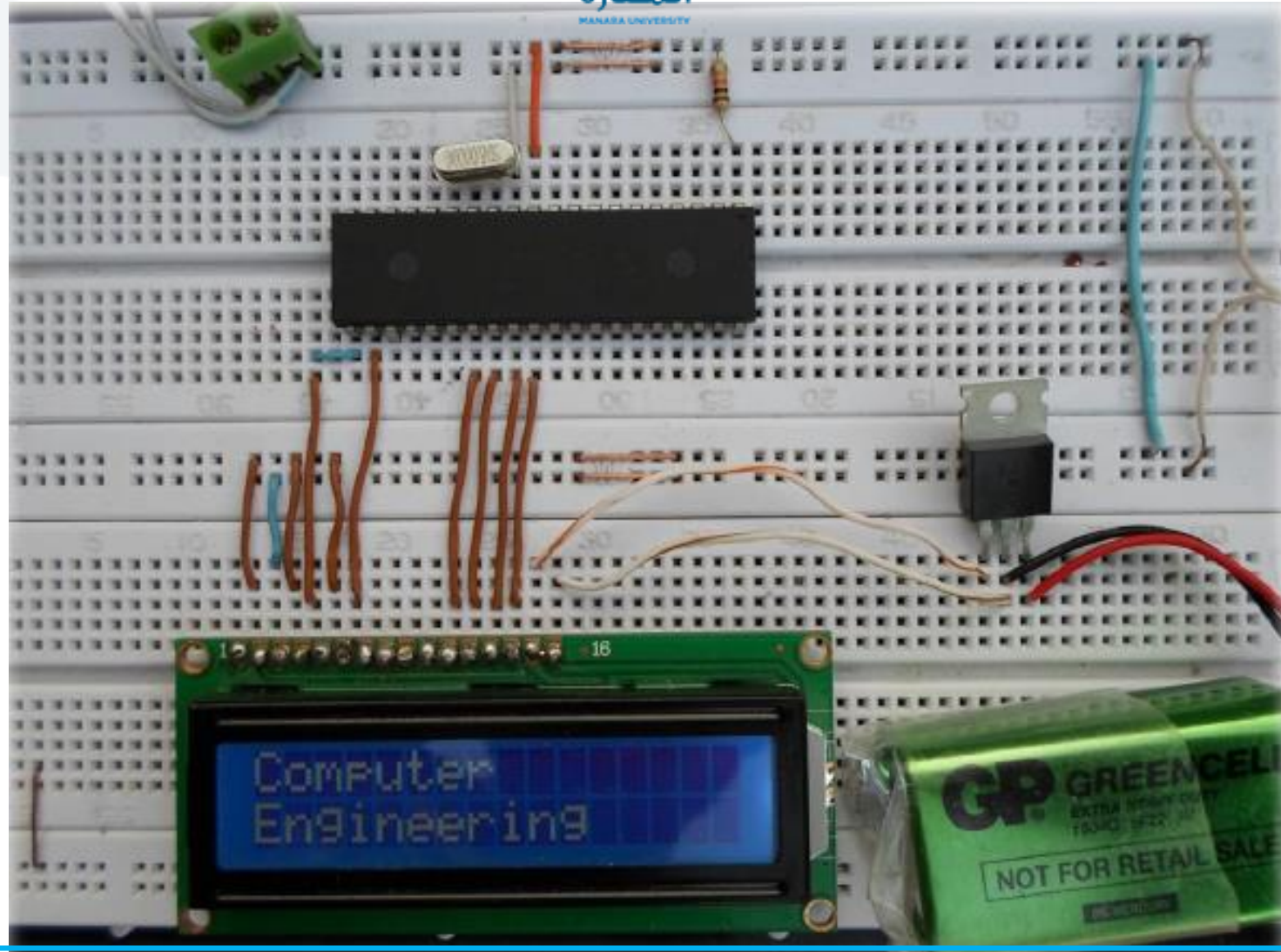
لاحظ أنني قد اخترت
PORTD في الميكرو لتوصيل

المثال التطبيقي الأول : البرنامج

```
sbit LCD_RS at RD2_bit;  
sbit LCD_EN at RD3_bit;  
sbit LCD_D4 at RD4_bit;  
sbit LCD_D5 at RD5_bit;  
sbit LCD_D6 at RD6_bit;  
sbit LCD_D7 at RD7_bit;
```

```
sbit LCD_RS_Direction at TRISD2_bit;  
sbit LCD_EN_Direction at TRISD3_bit;  
sbit LCD_D4_Direction at TRISD4_bit;  
sbit LCD_D5_Direction at TRISD5_bit;  
sbit LCD_D6_Direction at TRISD6_bit;  
sbit LCD_D7_Direction at TRISD7_bit;
```

```
void main()  
{  
  
    Lcd_Init();  
    Lcd_Cmd(_LCD_CURSOR_OFF);  
  
    while(1)  
    {  
        Lcd_Out(1,1,"Computer ");  
        Lcd_Out(2,1,"Engineering");  
    }  
}
```

المثال التطبيقي الثاني

المطلوب: عرض كلمة DISPLAY على الشاشة وجعلها تسير من اليسار إلى اليمين.
فكرة المشروع: هو أن نقوم بعرض الكلمة في أول السطر ثم نقوم بمسحها وعرضها بداية من الموضع الثاني ثم مسحها بعد زمن صغير وعرضها بداية من الموضع الثالث وهكذا حتى نصل إلى الموضع 16 فتظهر الكلمة وكأنها تتحرك ... ويتوقف مدى سرعة تحركها على الزمن الذي نحدده نحن عندما نعرض ونمسح الكلمة ...

لعرض الكلمة بداية من أول الصف نكتب الأمر كالاتي:

```
Lcd_Out (1,1,"DISPLAY");
```

ولعرضها بداية من الموضع الثاني نكتب الأمر الآتي:

```
Lcd_Out (1,2,"DISPLAY");
```

ولعرضها بداية من الموضع الثالث نكتب الأمر الآتي:

```
Lcd_Out (1,3,"DISPLAY");
```

```
for (i=1;i<=16;i++)
```

```
{
```

عدد مرات التكرار

الواامر المراد تكرارها

وهكذا حتى نصل إلى الموضع السادس عشر ...
وطبعاً لن نكتب الأمر 16 مرة فلابد من استخدام
FOR كما تعلمناها من قبل.

وليصبح البرنامج هكذا بدون ال 12 أمر الخاصين بالتوصيل والذين ينبغي علينا كتابتهم بالتأكيد:

```
int i;
```

هذا متغير صحيح نحتاجه للحلقة التكرارية FOR

```
void main()
```

```
{
```

```
  Lcd_Init();
```

```
  Lcd_Cmd(_LCD_CURSOR_OFF);
```

```
  while(1)
```

```
  {
```

```
    for(i=1;i<=16;i++)
```

```
    {
```

```
      Lcd_Out(1,i,"DISPLAY");
```

```
      delay_ms(50);
```

```
      Lcd_Cmd(_LCD_CLEAR);
```

```
      delay_ms(50);
```

```
    }
```

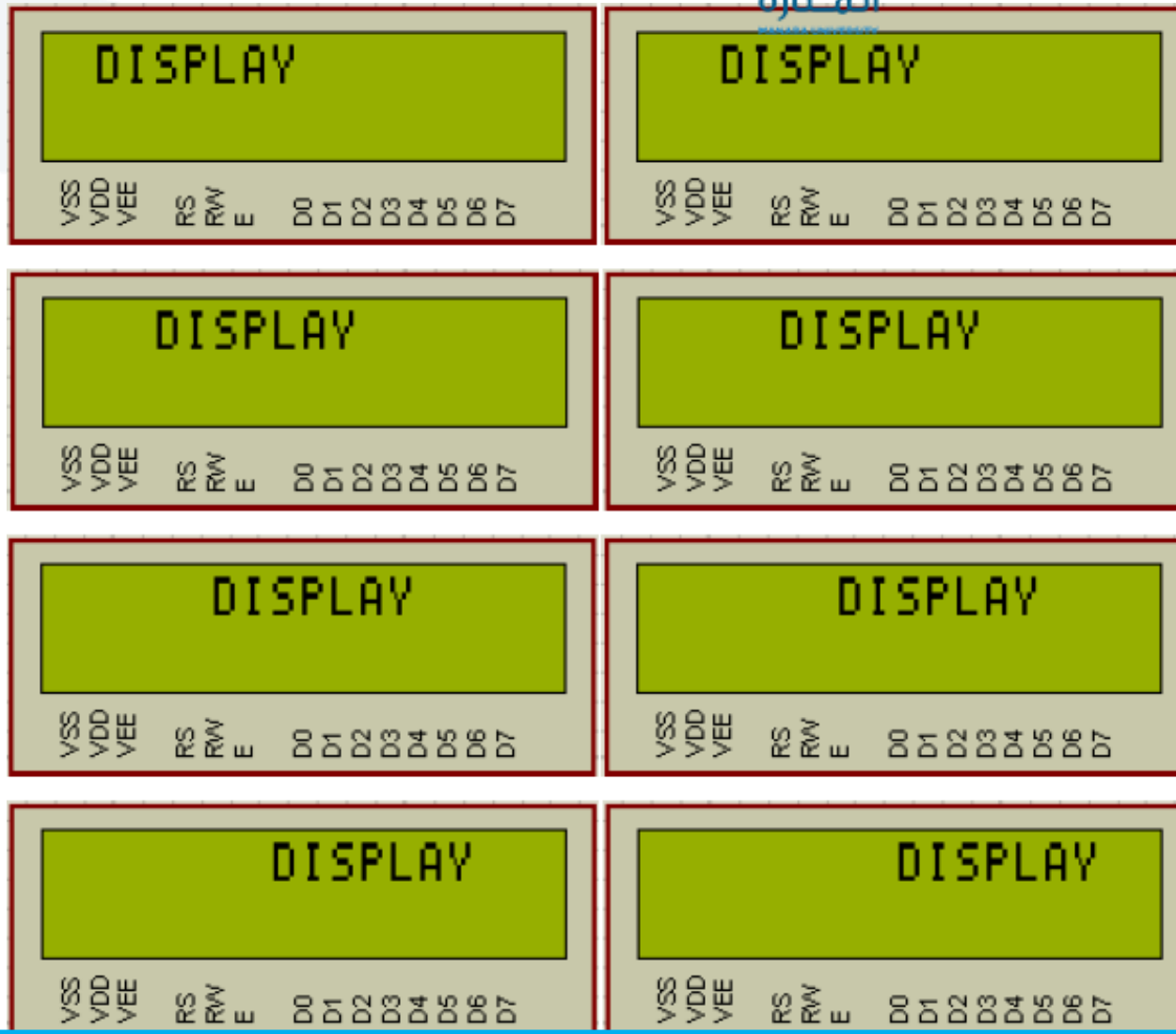
```
  }
```

```
}
```

لاظهار الكلمة على الشاشة كل مرة في مكان مختلف لان رقم العمود هو i وهو يتغير في كل عملية تكرار من واحد الى 16

لاظهار الكلمة لمدة 50 ميلي ثانية

مسح الشاشة والانتظار هكذا لمدة 50 ميلي ثانية



المطلوب: عرض القيم من واحد إلى عشرة على الشاشة .

فكرة المشروع: أول ما قد يتبادر إلى الذهن هو استخدام الحلقة التكرارية for وبالتالي سنحتاج إلى متغير صحيح نقوم بزيادته في كل تكرار ونقوم بإرساله إلى الشاشة، ولكن تظهر مشكلة بسيطة هنا وهي أن التتابع السابقة الخاصة بالشاشة لا تتعامل إلا مع السلاسل الحرفية أما المتغيرات الصحيحة فلا تتعامل معها، وبالتالي إذا قمنا بتعريف متغير صحيح وليكن

```
int x = 5;
```

ثم قمنا بكتابة التابع الآتي:

```
Lcd_Out(1,1, x );
```

فإنه لن يطبع القيمة خمسة، ولحل هذه المشكلة لكي نتمكن من طباعة القيمة خمسة الموجودة في المتغير x لابد من تحويله من شكل المتغير الصحيح إلى شكل السلسلة الحرفية (ولن تتغير قيمته فنحن نريد ان نقوم بتخزينه بصيغة السلسلة الحرفية ولكن الاختلاف أننا لن نتمكن من عمل العمليات الحسابية عليه) والذي يفعل ذلك هو التابع **IntToStr**



جامعة
المنارة
MANARA UNIVERSITY

المثال التطبيقي الثالث

يجب أن يكون متغير السلسلة الحرفية الذي سنحول فيه المتغير لا يقل طوله عن 7 حروف، وبالتالي لابد أن يتم تعريف متغير كالاتي:

```
Char str[7];
```

يتبقى الخطوات التي نريد بها تنفيذ المشروع كالاتي:

أولاً: سنستخدم الحلقة التكرارية for لأننا نريد إظهار الأرقام من واحد إلى 10 ولا نريد أن نكتب نفس الأوامر أكثر من مرة.

ثانياً: داخل الحلقة for نقوم أولاً بمسح الشاشة، وهذا أمر هام لابد منه لأن الشاشة سيكتب عليها في كل مرة من مرات التكرار، وبالتالي قبل الكتابة لابد من مسحها أولاً.

ثالثاً: نقوم بتحويل الرقم i الذي نستخدمه حلقة for إلى صيغة السلسلة الحرفية باستخدام التابع الذي أشرنا إليه سابقاً.

رابعاً: نقوم بإظهار الرقم.

خامساً: نقوم بعمل تأخير لمدة واحد ثانية حتى نستطيع رؤية الأرقام.

سادساً: لا ننسى كتابة ال 12 أمر الخاصين بتوصيل الأرجل.



وعندئذ يصبح البرنامج كالآتي:

```
int i;
char str[7];
void main()
{
    Lcd_Init();
    Lcd_Cmd(_LCD_CURSOR_OFF);

    while(1)
    {
        for(i=1;i<=10;i++)
        {
            Lcd_Cmd(_LCD_CLEAR);
            IntToStr(i, str);
            Lcd_Out(1, 1, str);
            delay_ms(250);
        }
    }
}
```


المثال التطبيقي الثالث

نلاحظ أن ما بداخل الحلقة for سيتم تنفيذه 10 مرات، في كل مرة سيكون فيها المتغير i بقيمة متزايدة، ففي المرة الأولى سيكون المتغير يساوي واحد وفي المرة الثانية سيكون المتغير يساوي 2 وهكذا تستمر حلقة التكرار حتى يصل إلى القيمة 10 مع ملاحظة أن متغير السلسلة المحرفية تم تحديد حجمه بسبع حروف ...

