

الباب الرابع (4) CHAPTER

التحكم بشاشات الـ LCD باستخدام الميكروكونترولر

Controlling LCD Screens Using PIC Microcontrollers

عرفنا فى الجزئية السابقة أنه يمكن إستخدام الـ LEDs أو حتى الـ 7-segments مع الـ Microcontroller بسهولة وذلك لعرض أرقام أو حتى حروف. ولكن فى السنين الأخيرة زادت شعبية شاشات الكريستال السائل Liquid Crystal Screen أو LCD لهذه الأسباب:

1. هبوط أسعار شاشات الـ LCD.
2. إمكانية عرض أرقام Numbers وحروف Characters بل ورسومات Pictures أيضاً على هذه الشاشة وهذه ميزة يصعب الحصول عليها بإستخدام الـ LEDs.
3. التنوع الكبير فى شاشات الـ LCD من حيث الحجم وجودة الصور ونوع الشاشة من حيث كونها ملونة Colored أو أحادية اللون Single color .
4. إحتواء شاشات الـ LCD على متحكمات داخلية ذكية Intelligent controllers تتيح التعامل مع هذه الشاشة بسهولة عن طريق مجموعة من الأوامر الجاهزة وبطريقة سهلة.
5. لا تحتاج إلى أطراف كثيرة من الـ Microcontroller على عكس الـ LEDs أو الـ 7-segments.

يوجد العديد من شاشات الـ LCD فى السوق. وتختلف أطراف الشاشة وطريقة التامل معها تبعاً لنوع الـ Controller الداخلى للشاشة. لذلك فى هذه الجزئية سوف نتعلم كيفية توصيل وبرمجة أشهر أنواع شاشات الـ LCD مع الـ Microcontroller وهى شاشة الـ HD44780 والتى تم تصنيعها عن طريق شركة HITACHI اليابانية. وهى شاشة من Alphanumeric Dot Matrix أى أنها يمكن أن تطبع حروف وأرقام أو رموز فقط (أى أنها لا تستخدم لطباعة صور). وتسمى Dot matrix لأن كل حرف عبارة عن نقاط صغيرة جداً يتم التحكم بها عن طريق الـ Controller الذى يوجد بداخل الشاشة. وتتميز هذه الشاشة برخصها وسهولة توافرها فى الأسواق وأنواعها المتعددة كما أنها تتوافر بأحجام

متعددة (عدد حروف × عدد السطور). ولكننا في هذا الكتاب سوف نتعامل مع شاشة $2 \text{ lines} \times 16$ characters) أي أنها سوف تكون شاشة مكونة من سطرين فقط وكل سطر يمكنه طباعة 16 حرف بحد أقصى.

وشكل (4-1) يوضح أطراف الشاشة HD44780 والتي تنقسم إلى ثلاثة مجموعات وهم:

1. مجموعة أطراف التغذية Power supply pins :

Notes	Function	Name	pin
يتم توصيل هذا الطرف بالأرضى Gnd	Ground pin	V _{SS}	1
يتم توصيل هذا الطرف بالـ 5V	+ve supply (5V) pin	V _{DD}	2
للتحكم بمستوى تباين الحروف على الشاشة ويتم توصيله بمقاومة متغيرة	Contrast pin	V _{EE}	3

2. مجموعة أطراف التحكم Control pins :

Note	Function	Name	pin
إذا كانت قيمة RS=0 فهذا يعنى أنه مطلوب من الشاشة تعمل على تنفيذ أمر ما مثل أمر المسح Clear. أما إذا كانت RS=1 فهذا يعنى أن الشاشة سوف تطبع حروف أو Data	Register select pin	RS	4
يستخدم هذا الطرف للتحديد إمكانية إرسال أو إستقبال بيانات من وإلى الشاشة. فإذا كانت RW=1 فهذا يعنى أن المستخدم يريد قراءة شيء مخزن بداخل الشاشة وأما إذا كانت RW=0 فهذا يعنى إرسال بيانات إلى الشاشة	Read/Write pin	RW	5
إذا تحولت قيمة هذا الطرف من High إلى Low أو من 1 إلى 0 فإن الشاشة سوف تعمل تسجيل القيمة الموجودة على الـ Data pins	Enable pin	EN	6

3. مجموعة أطراف البيانات Data pins :

Note	Function	Name	pin
تستخدم هذه الأطراف لنقل البيانات من وإلى الشاشة سواء أكانت حروف سوف تُطبع الشاشة أو أوامر يتم تنفيذها	Data pins	D0:D7	7:14

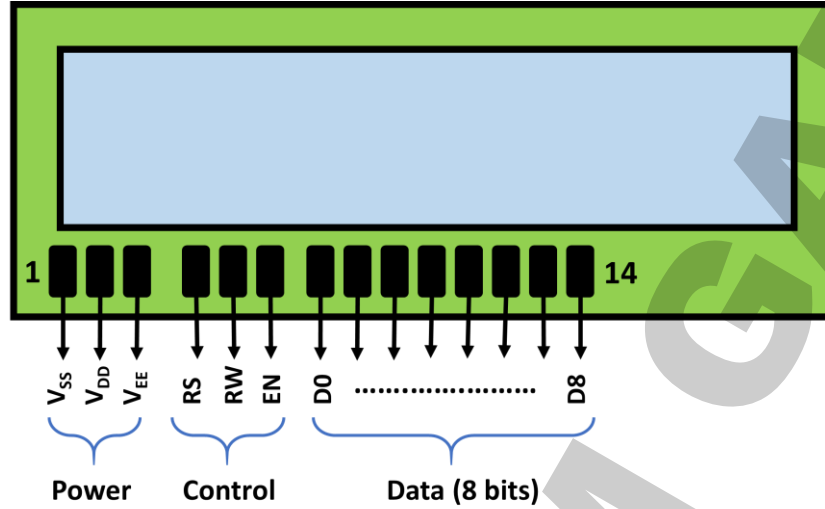


Fig. 4-1: LCD pins.

أطراف شاشة الـ LCD

وعند توصيل الشاشة بالـ Microcontroller فإننا نحتاج إلى توصيل أطراف الـ Control والـ Data. ولكن عند توصيل هذه الأطراف سوف نحتاج إلى 11 طرف pin من أطراف الـ Microcontroller وهذا سوف يؤدي إلى إستخدام أكثر من PORT وذلك بسبب أن كل PORT لا يزيد عن ثمانية أطراف pins 8. لذلك يجب أن نعرف ما هي الأطراف التي يمكننا الإستغناء عنها. فعلى سبيل المثال يمكننا الإستغناء عن الطرف RW وذلك لأننا لا نريد القراءة من شاشة الـ LCD بل نريد الكتابة عليها لذلك سوف نجعل الطرف RW=0 أى أننا سوف نوصله بالأرضى. ويمكننا أيضاً الإستغناء عن أربعة أطراف أخرى وهى الأطراف D0:D3 حيث أن الشركة المصنعة لهذه الشاشة أعطت الإمكانية لتوصيل هذه الشاشة بالميكروكونترولر إما على 8-bits mode أو على 4-bits mode. أى أنه يمكن توصيل أطراف الـ Data كلها أو توصيل أربعة أطراف منها فقط. وبذلك يمكن توصيل أطراف الشاشة بالـ Microcontroller بأطراف الـ PORTB (على سبيل المثال) كما هو موضح بشكل (2-37). ويلاحظ فى هذا الشكل أننا نحتاج فقط إلى 6 أطراف من الـ PORT.

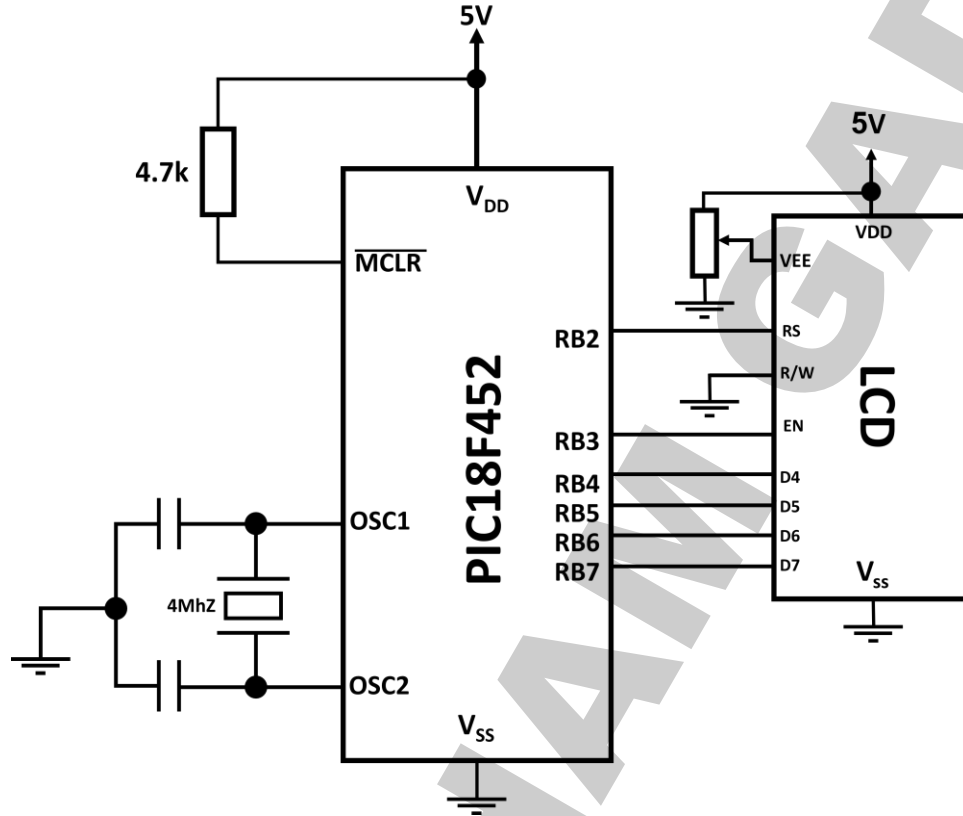


Fig. (4-2): Connecting LCD to PORTB pins.

توصيل أطراف الشاشة بأطراف الـ PORTB

الشيء الأخير الذي يجب معرفته هو كيفية برمجة الشاشة مع الـ Microcontroller وذلك باستخدام برنامج الـ mikroC. لذلك يتم أولاً تعريف أطراف الشاشة قبل بداية الـ main program كالآتي:

```
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;
```

توصيلات الشاشة إلى أطراف الـ
PORTB

```
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;
```

تحديد اتجاه الأطراف الخاصة
بتوصيلات الشاشة

ثانياً: يتم تشغيل الشاشة وإعدادها عن طريق الأمر التالى ويكتب داخل الـ main():

Command	Function
lcd_init()	تشغيل وإعداد الشاشة (يُكتب فى بداية البرنامج)

ولكى تُعطى أمر للشاشة يتم كتابة الآتى:

Command	Function
lcd_cmd(الأمـر المطلوب)	<p>فإذا كان المطلوب:</p> <ol style="list-style-type: none"> 1. مسح أى حروف مكتوبة على الشاشة lcd_cmd(_lcd_clear) 2. إزالة مؤشر الكتابة Cursor lcd_cmd(_lcd_cursor_off) 3. إزاحة الكلام المكتوب على الشاشة إلى اليسار lcd_cmd(_lcd_shift_left) 4. إزاحة الكلام المكتوب على الشاشة إلى اليمين lcd_cmd(_lcd_shift_right) 5. جعل مؤشر الكتابة فى السطر الأول lcd_cmd(_lcd_first_row) 6. جعل مؤشر الكتابة فى السطر الثانى lcd_cmd(_lcd_second_row) 7. تشغيل الشاشة lcd_cmd(_lcd_turn_on) 8. إطفاء الشاشة lcd_cmd(_lcd_turn_off)

كما يمكن إستعمال الأوامر الخاصة بالطباعة على الشاشة بداخل كود الـ main ومنها الآتى:

Command	Function
lcd_out(”الرسالة”، رقم الخانة، رقم السطر)	يستخدم هذا الأمر لكتابة رسالة على الشاشة عند سطر محدد وخانة محددة
lcd_out_cp(”الرسالة”)	يستخدم هذا الأمر لكتابة رسالة على الشاشة من آخر موضع للمؤشر

Example (1)

Draw a circuit and write a microcontroller program by using PIC18F452 to print two messages as shown in Fig. (2-38). The delay between each message is 1 second.

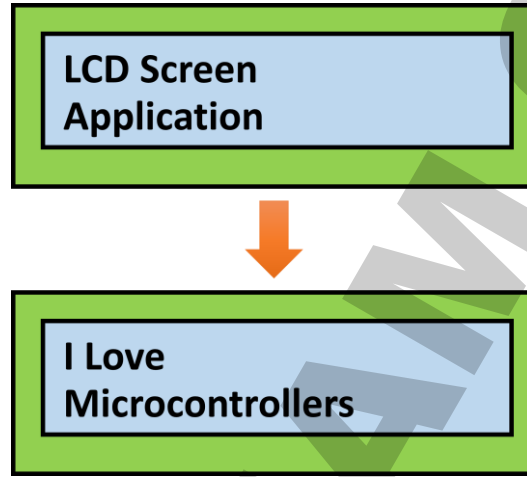


Fig. 4-3.

Solution:

في هذا المثال يريد طباعة رسالتين على شاشة الـ LCD والفرق الزمني بين الرسالتين 1 ثانية” كما هو موضح في الشكل السابق. لذلك يتم توصيل الشاشة بالـ Microcontroller كما هو موضح بالشكل التالي.

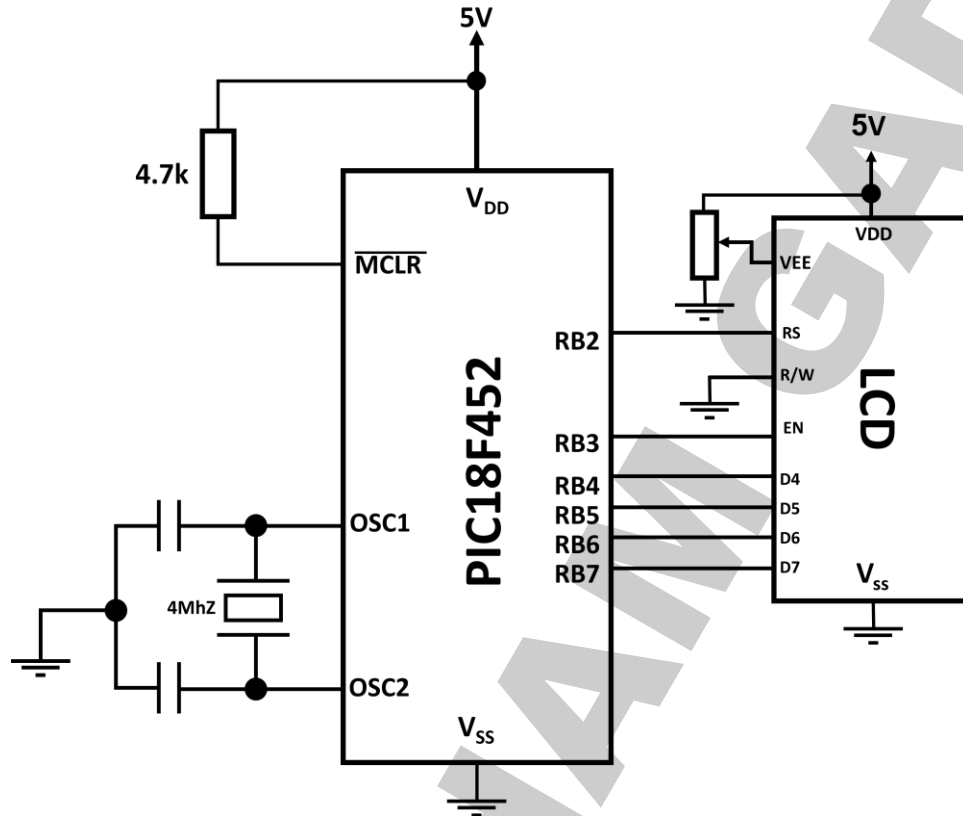


Fig. 4-4.

Program:

```

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;

```

توصيلات الشاشة إلى أطراف الـ
PORTB

```

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;

```

تحديد اتجاه الأطراف الخاصه
بتوصيلات الشاشة

```

void main ()

```

```

{

```

```

    lcd_Init();

```

```

    // إعداد الشاشة

```

```

lcd_cmd(_lcd_clear);           // مسح الشاشة (إختياري)
lcd_cmd(_lcd_cursor_off);      // إطفاء المؤشر الذى على الشاشة
while(1)
{
    lcd_out(1,1,"LCD Screen");
    lcd_out(2,1,"Application");
    delay_ms(1000);             // توقف لمدة 1 ثانية
    lcd_cmd(_lcd_clear);        // مسح الشاشة
    lcd_out(1,1,"I Love");
    lcd_out(2,1,"Microcontrollers");
    delay_ms(1000);             // توقف لمدة 1 ثانية
    lcd_cmd(_lcd_clear);        // مسح الشاشة
}
}

```

طباعة الرسالة الأولى

طباعة الرسالة الثانية

Example (2)

Draw a circuit and write a microcontroller program by using PIC18F452 to print the numbers from 0 to 255. The delay between each number is 0.5 second.

Solution:

فى هذا المثال يريد طباعة الأرقام من 0 إلى 255 على شاشة الـ LCD مع وجود فارق زمنى 0.5 ثانية بين كل رقم والرقم الذى يليه. ويمكن فعل ذلك عن طريق عداد يعمل بالـ FOR Loop كما فعلنا سابقاً مع دائرة الـ 7-segments. المشكلة هنا أن الشاشة لا تستطيع التعامل مع الأرقام بشكل مباشر لأنها تفهم فقط حروف الـ ASCII. لذلك يتم إستخدام أوامر خاصة تستخدم لتحويل الأرقام Digits إلى حروف ASCII strings. والجدول التالى يبين بعض الأوامر الخاصة بتحويل الأرقام إلى حروف. ويجب أن تعرف أن الأمر يختلف على حسب نوع المخزن الرقمى الذى سوف يتم تحويله. وحيث أننا سوف نتعامل مع قيم صحيحة لن تزيد عن 255 إذا يمكننا إستعمال مخزن نوعية char أو int.

Command	Function
ShortToStr(المخزن الحرفي, المخزن الرقمي)	تحويل مخزن رقمي مكون من 8-bits (0 إلى 255) مثل char إلى حروف ASCII. ويجب أن لا يقل المخزن الحرفي عن 5 حرف على الأقل
IntToStr(المخزن الحرفي, المخزن الرقمي)	تحويل مخزن رقمي مكون من 16-bits (0 إلى 65535) مثل int إلى حروف ASCII. ويجب أن لا يقل المخزن الحرفي عن 7 حرف على الأقل
floatToStr(المخزن الحرفي, المخزن الرقمي)	تحويل مخزن رقمي كسري مثل float إلى حروف ASCII. ويجب أن لا يقل المخزن الحرفي عن 15 حرف على الأقل

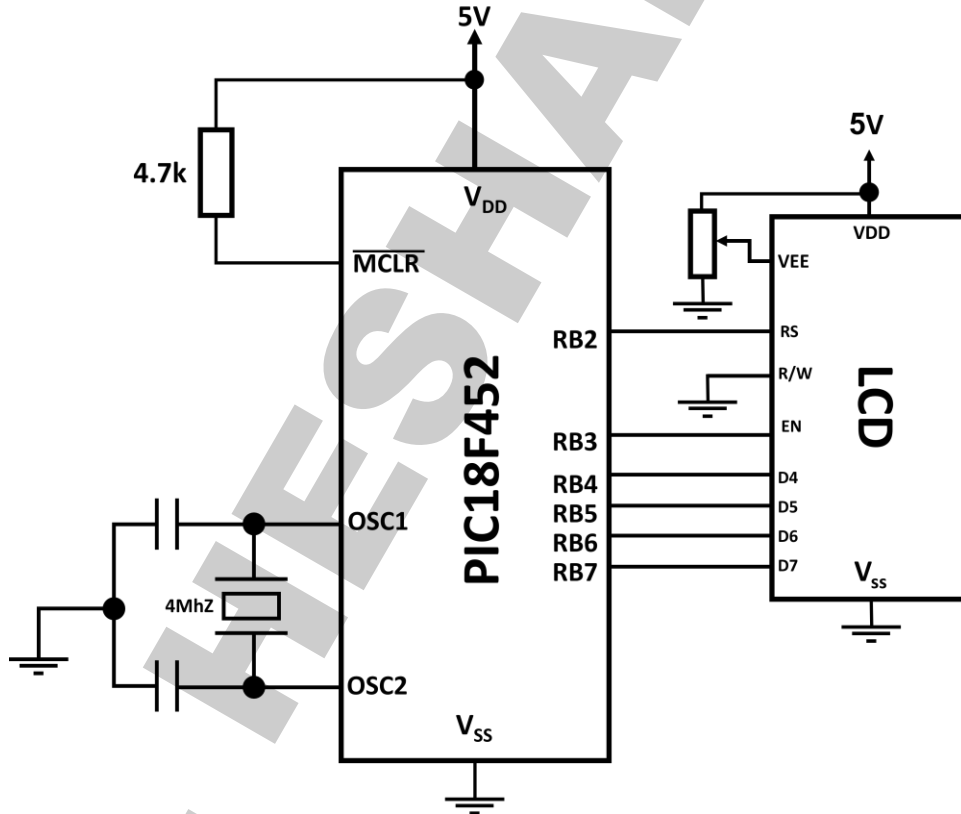


Fig. 4-5.

Program:

```
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;
```

توصيلات الشاشة إلى أطراف الـ
PORTB

```
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;
```

تحديد إتجاه الأطراف الخاصه
بتوصيلات الشاشة

```
void main ()
```

```
{
```

```
int i;
```

```
char txt[7];
```

```
lcd_Init();
```

// إعداد الشاشة

```
lcd_cmd(_lcd_clear);
```

// مسح الشاشة (إختياري)

```
lcd_cmd(_lcd_cursor_off);
```

// إطفاء المؤشر الذى على الشاشة

```
while(1)
```

```
{
```

```
for(i=0;i<=255;i++)
```

```
{
```

```
IntToStr(i,txt);
```

```
lcd_out(1,1,txt);
```

```
delay_ms(500);
```

```
}
```

```
}
```

- تحويل الرقم إلى حروف
- طباعة الحروف على الشاشة
- الإنتظار لمدة نصف ثانية