برمجة المتحكمات المصغرة

التجارب العملية

الجلسة الخامسة

Programming

Embedded Systems Microcontroller

You Can Practice Microcontroller Programming Easily Now!

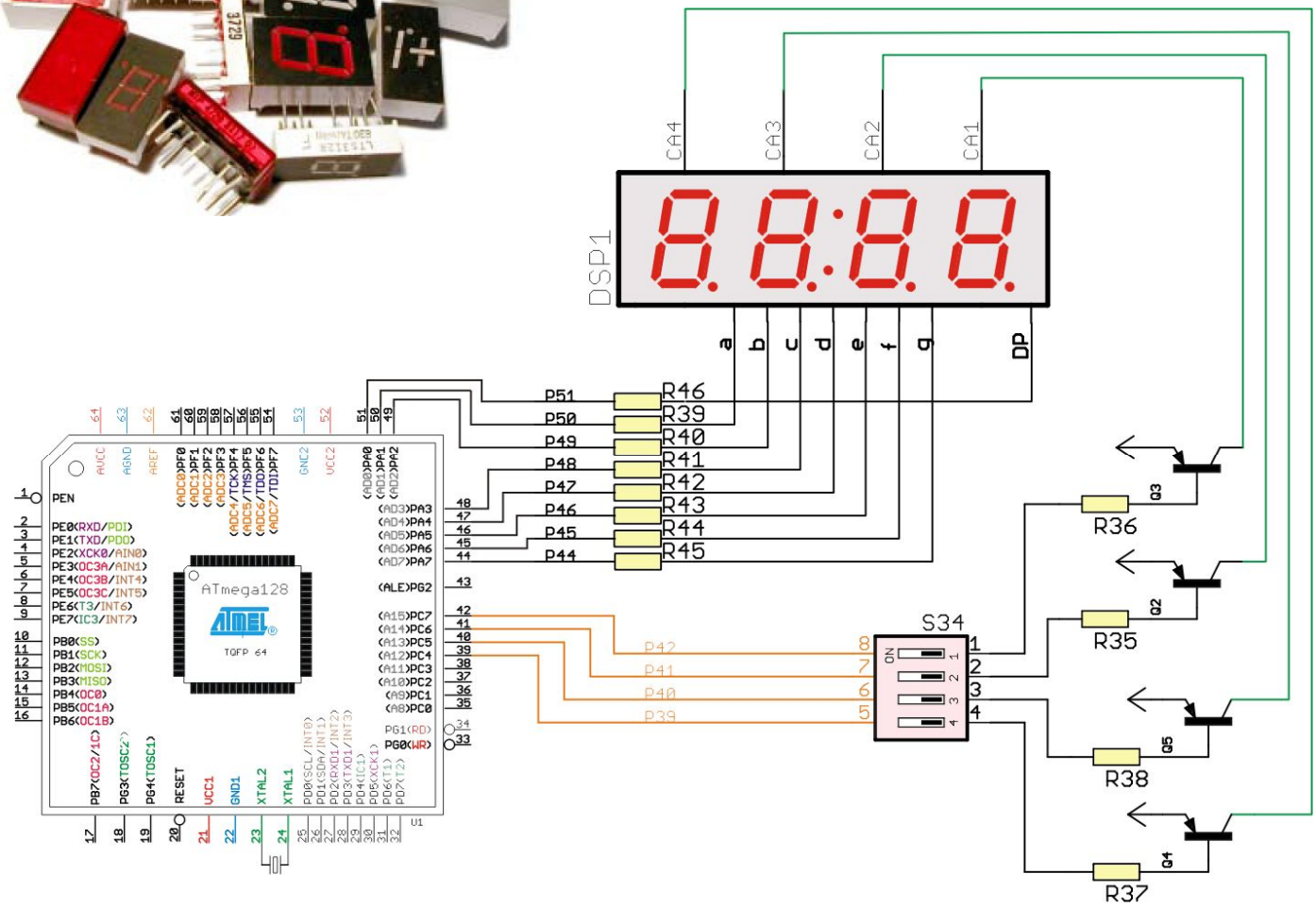
WALID BALID, Tuesday, December 15, 2009

التجربة الرابعة عشرة: لوحات الإظهار ذات السبع قطع
Exp.14: 7-Segment Single Display

الغاية من التجربة:

توصيل وبرمجة لوحة إظهار سباعية (7-Segment).

مخطط التوصيل:



شرح عمل الدارة:

سوف نقوم بكتابة برنامج التشغيل للوحات الإظهار السباعية على ثلاث مراحل:

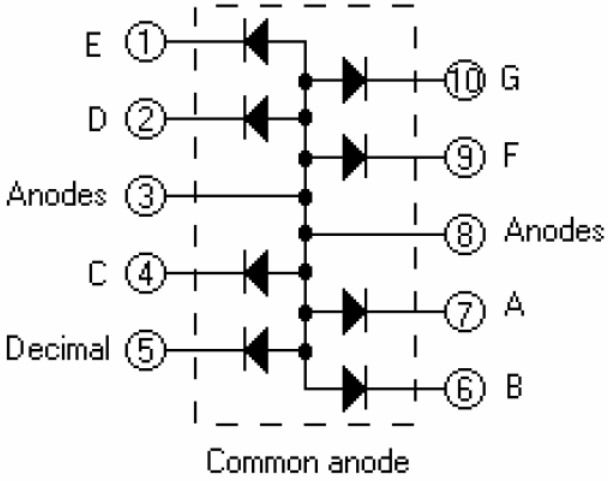
1- تشغيل لوحة إظهار فردية كعداد عشري 9 – 0.

2- تشغيل لوحة إظهار ثنائية كعداد عشري 99 – 0.

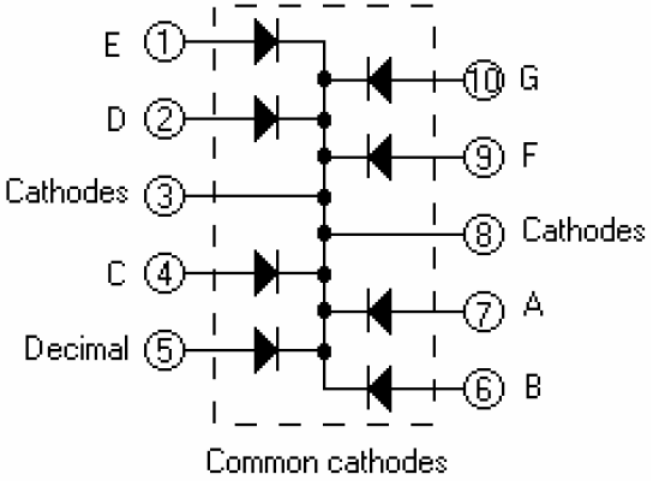
3- تشغيل لوحة إظهار رباعية كعداد عشري 9999 – 0.

إن عملية تشغيل لوحة الإظهار ذات السبع قطع بما في ذلك إظهار الأرقام على اللوحة، يتعلق مباشرة بقيمة تشفير كل رقم من أرقم لوحة الإظهار، ومن أجل معرفة تشفير كل رقم من الأرقم فإنه يجب معرفة بنية لوحة الإظهار ذات السبع قطع.

الشكل التالي يبين المخطط الداخلي للوحات الإظهار ذات السبع قطع.



Common anode



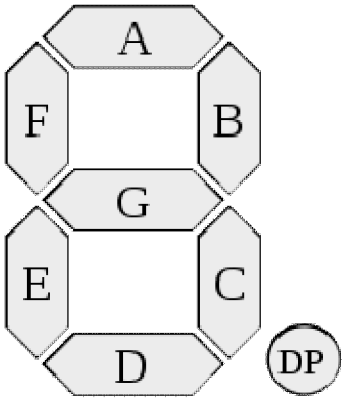
Common cathodes

كما هو واضح من الشكل أعلاه أن لوحة الإظهار تحوي على سبعة ثنائيات ضوئية (ثنائي ضوئي لكل قطعة)، بالإضافة إلى ثنائي ضوئي ثامن للفاصلة العشرية.

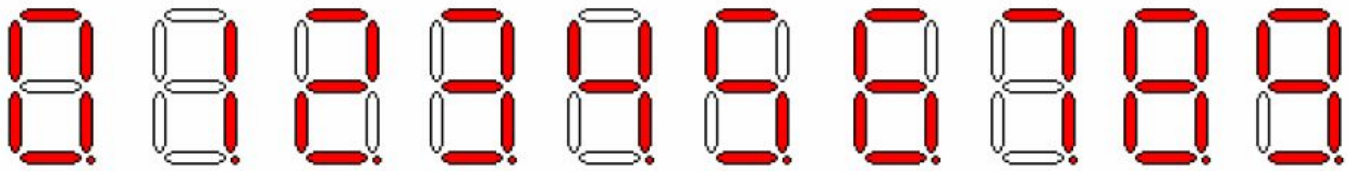
إن جميع الثنائيات الضوئية تكون موصولة بإحدى طريقتين، إما مصعد مشترك (CA) أو مهبط مشترك (CC).

الشكل اليميني يوضح وصلة المهبط المشترك، وأما اليساري فيوضح وصلة المصعد المشترك.

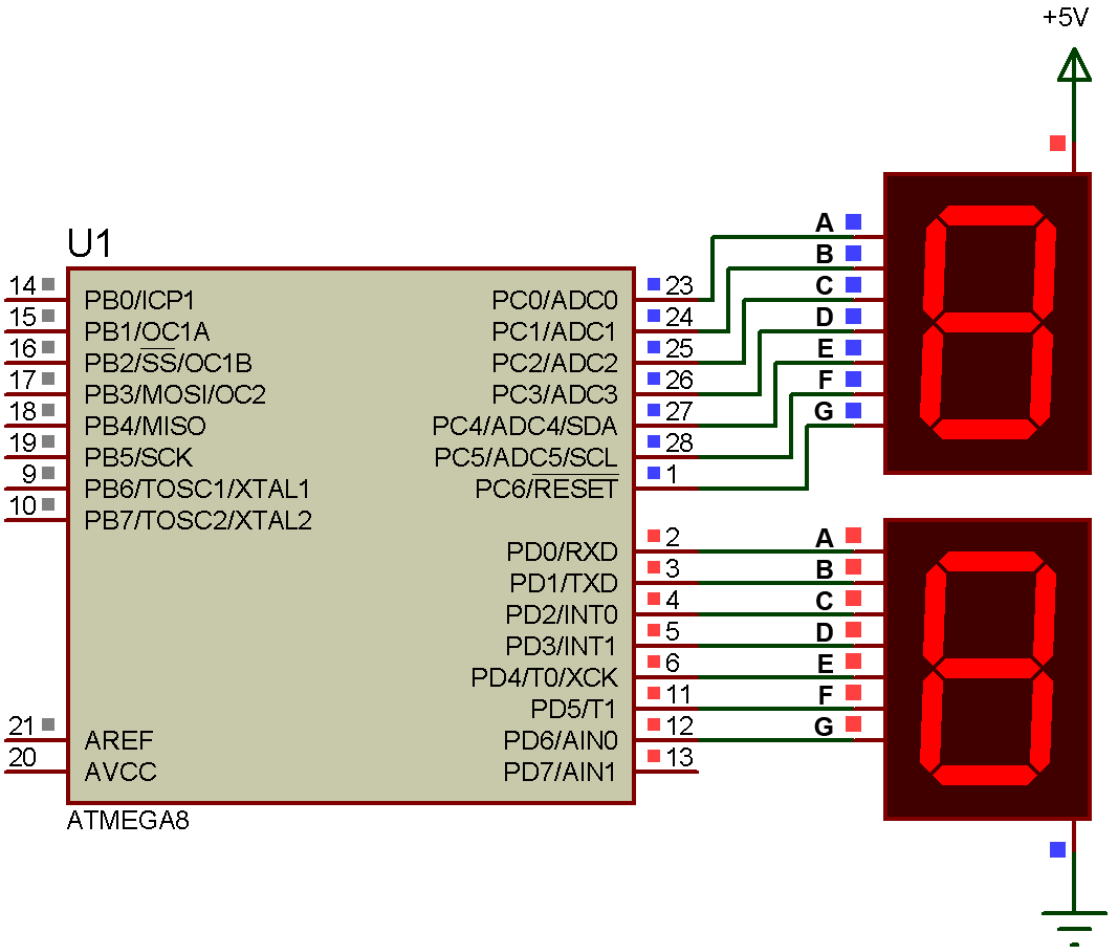
إن القطع السبعة في لوحة الإظهار بالإضافة إلى نقطة الفاصلة العشرية يتم الإشارة إلى كل واحدة منها بحرف (A,.....,H)، إن هذا الترميز لكل قطعة هو ترميز عالمي معتمد من قبل كل الشركات كما في الشكل:



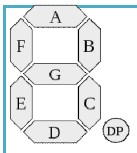
وبالتالي من أجل إظهار الرقم (1) على لوحة الإظهار السباعية، فإنه يجب تشغيل القطعتين B,C وهكذا بالنسبة لباقي الأرقام.



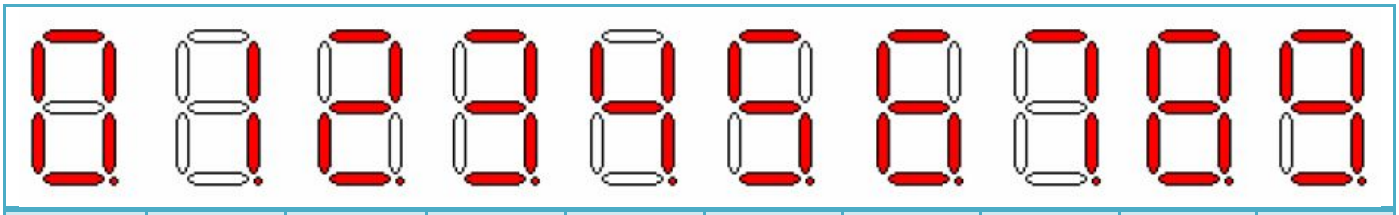
بفرض إنه لدينا شاشة إظهار سباعية ذات مصعد مشترك موصولة مع البوابة PortC لمتحكم مصغر وأخرى ذات مهبط مشترك موصولة مع البوابة PortD كما في الشكل التالي:



وبالتالي فإنه من أجل تشغيل أي قطعة في اللوحة العلوية (ذات المهبط المشترك)، ينبغي وضع صفر منطقي على قطب المتحكم الموصول مع القطعة.

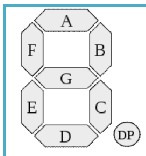
الرقم المطلوب إظهاره	القيمة على البوابة PortC								القطع التي هي في حالة عمل: 
	0-A	1-B	2-C	3-D	4-E	5-F	6-G	7-H	
0	0	0	0	0	0	0	1	1	A.B.C.D.E.F
1	1	0	0	1	1	1	1	1	B.C
2	0	0	1	0	0	1	0	1	A.B.D.E.G
3	0	0	0	0	1	1	0	1	A.B.C.D.G
4	1	0	0	1	1	0	0	1	B.C.F.G
5	0	1	0	0	1	0	0	1	A.C.D.F.G
6	0	1	0	0	0	0	0	1	A.C.D.E.F.G
7	0	0	0	1	1	1	1	1	A.B.C
8	0	0	0	0	0	0	0	1	A.B.C.D.E.F.G
9	0	0	0	0	1	0	0	1	A.B.C.D.F.G

وبالتالي بتحويل القيم الثنائي إلى عشرية أو ست عشرية يمكن أن نحدد مايلي:

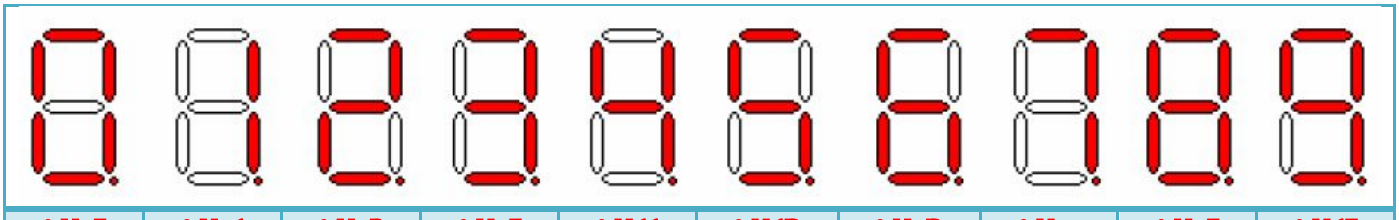


&HC0	&HF9	&HA4	&HB0	&H99	&H92	&H82	&HF8	&H80	&H90

وبالمثل فإنه يمكن حساب القيم من أجل تشغيل اللوحة السفلية (ذات المصعد المشترك)، ولكن الحالة معكوسة تماماً، لأنه ينبغي وضع واحد منطقي على قطب المتحكم الموصول مع القطعة المراد تشغيلها.

القطع التي هي في حالة عمل: 	القيمة على البوابة PortC								الرقم المطلوب إظهاره
	7-H	6-G	5-F	4-E	3-D	2-C	1-B	0-A	
A.B.C.D.E.F	0	0	1	1	1	1	1	1	0
B.C	0	0	0	0	0	1	1	0	1
A.B.D.E.G	0	1	0	1	1	0	1	1	2
A.B.C.D.G	0	1	0	0	1	1	1	1	3
B.C.F.G	0	1	1	0	0	1	1	0	4
A.C.D.F.G	0	1	1	0	1	1	0	1	5
A.C.D.E.F.G	0	1	1	1	1	1	0	1	6
A.B.C	0	0	0	0	0	1	1	1	7
A.B.C.D.E.F.G	0	1	1	1	1	1	1	1	8
A.B.C.D.F.G	0	1	1	0	1	1	1	1	9

وبالتالي بتحويل القيم الثنائي إلى عشرية أو ست عشرية يمكن أن نحدد مايلي:



&H3F	&H06	&H5B	&H4F	&H66	&H6D	&H7D	&H07	&H7F	&H6F

ملاحظة: عند تحويل قيم التشفير بين شاشة إظهار من نوع مصعد مشترك إلى شاشة إظهار من نوع مهبط مشترك فإنه يكفي طرح القيمة من القيمة &HFF (255 DEC).

التعليمات الجديدة:

إن قيم التشفير يمكن تخزينها في البرنامج ضمن ما يسمى بجدول بيانات (LUT: Look-Up Table)، وهناك تعليميتين أساسيتين وبعض التعليمات الأخرى للتخزين والقراءة من جداول البيانات.

التعليمة البرمجية	شرح التعليمة
`Data var1 , … , varn`	تخزين بيانات (رقمية، محرفية) في ذاكرة البرنامج عند لافتة معينة، حيث أن العنصر الأول له الدليل 0.
`var = Lookup(Index , label)`	جلب قيمة من LUT مخزن في **Data** عند اللافتة `label` وله الدليل `Index`.
`Var = Lookupstr(Index , Label)`	جلب محرف من LUT مخزن في **Data** عند اللافتة `label` وله الدليل `Index`.
`Var = Lookdown(value,Label,Entries)`	البحث عن دليل (`Index`) قيمة (`value`) مخزن في **Data** عند اللافتة `label` على أن يبتم البحث ضمن عدد محدد من العناصر (`Entries`).
`Restore Label`	تقوم بوضع مؤشر (عداد) بيانات إلى أول عنصر موجود في LUT مخزن في **Data** عند اللافتة `label` من أجل قراءة البيانات باستخدام تعليمة: **Read** `Var`
`Read Var`	قراءة القيم الموجودة في LUT مخزن في **Data** والتي تم الإشارة إليها باستخدام التعليمة **Restore** وإسنادهم إلى المتحول `Var`.

برنامج تشغيل الدارة (لوحة إظهار فردية كعداد عشري 9-0):

متطلبات التوصيل: يجب إغلاق نقطة الوصل 4 من المفتاح SW34.

```
$regfile = "m128def.dat"
$crystal = 8000000
'---------------------

Config Porta = Output
Segment Alias Porta
Porta = &H81

Config Portc.4 = Output
Ctr1 Alias Portc.4
Reset Ctr1
'---------------------
Dim I As Byte
'---------------------
Do
   For I = 0 To 9
      Segment = Lookup(i , Seg)
      Waitms 1000
   Next I
Loop

End
'---------------------

Seg:
Data &H81 , &HF3 , &H49 , &H61 , &H33 ,
&H25 , &H05 , &HF1 , &H01 , &H21
```

التوجيهات.

تعريف البوابة الموصل معها شاشة الإظهار.
تعيين القيمة الأولية على البوابة (&H81) وهي تظهر الرقم صفر على شاشة الإظهار

تعريف قطب التحكم بالمصعد المشترك للوحة الإظهار

تعريف المتحولات

حلقة البرنامج الرئيسي يتم فيها:
قراءة القيم المخزنة في (Data) عند اللافتة (Seg) باستخدام التعليمة (Lookup)، قيمة تلو لأخرى.

لافتة جدول تخزين البيانات.

برنامج تشغيل الدارة (تشغيل لوحة إظهار ثنائية كعداد عشري 99 – 0):

متطلبات التوصيل: يجب إغلاق نقطتي الوصل 4,5 من المفتاح SW34.

```
$regfile = "m128def.dat"
$crystal = 8000000
'--------------------
Config Porta = Output
Segment Alias Porta
Porta = &H81

Config Portc.4 = Output
Display1 Alias Portc.4

Config Portc.5 = Output
Display2 Alias Portc.5

Set Display1 : Set Display2
'--------------------
Dim I As Byte , J As Byte
'--------------------

Do
   For I = 0 To 9
      For J = 0 To 9
         Segment = Lookup(i , Seg)
         Reset Display1 : Waitms 20
         Set Display1

         Segment = Lookup(j , Seg)
         Reset Display2 : Waitms 20
         Set Display2
      Next J
   Next I
Loop

End
'--------------------
Seg:
Data &H81 , &HF3 , &H49 , &H61 , &H33 ,
&H25 , &H05 , &HF1 , &H01 , &H21
```

التوجيهات.

تعريف البوابة الموصل معها خطوط البيانات لشاشات الإظهار.

تعيين القيمة الأولية على البوابة (&H81) وهي تظهر الرقم صفر على شاشة الإظهار

تعريف قطب التحكم بالمصعد المشترك للوحة الإظهار الأولى والثانية.

إطفاء كلا لوحتي الإظهار.

تعريف المتحولات

حلقة البرنامج الرئيسي يتم فيها:
قراءة القيم المخزنة في (Data) عند اللافتة (Seg) باستخدام التعليمة (Lookup)، قيمة تلو لأخرى.
تشغيل لوحة الإظهار الأولى.
إيقاف لوحة الإظهار الأولى وتشغيل الثانية.
تكرار العملية...

لافتة جدول تخزين البيانات.

برنامج تشغيل الدارة (تشغيل لوحة إظهار رباعية كعداد عشري 9999 – 0):

```
$regfile = "m128def.dat"
$crystal = 8000000
'---------------------
Config Porta = Output
Segment Alias Porta
Porta = &H81

Config Portc = &B11110000
Display1 Alias Portc.4
Display2 Alias Portc.5
Display3 Alias Portc.6
Display4 Alias Portc.7

Set Display1 : Set Display2
Set Display3 : Set Display4
'---------------------
Dim I As Byte , J As Byte
Dim K As Byte , L As Byte
'---------------------
Do
   For I = 0 To 9
      For J = 0 To 9
         For K = 0 To 9
            For L = 0 To 9
               Segment = Lookup(i , Seg)
               Reset Display1 : Waitms 7
               Set Display1

               Segment = Lookup(j , Seg)
               Reset Display2 : Waitms 7
               Set Display2

               Segment = Lookup(k , Seg)
               Reset Display3 : Waitms 7
               Set Display3

               Segment = Lookup(l , Seg)
               Reset Display4 : Waitms 7
               Set Display4
            Next L
         Next K
      Next J
   Next I
Loop

End
'---------------------
Seg:
Data &H81 , &HF3 , &H49 , &H61 , &H33 ,
&H25 , &H05 , &HF1 , &H01 , &H21
```

التوجيهات.

تعريف البوابة الموصل معها خطوط البيانات لشاشات الإظهار.

تعيين القيمة الأولية على البوابة (&H81) وهي تظهر الرقم صفر على شاشة الإظهار

تعريف أقطاب التحكم بالمصعد المشترك للوحات الإظهار الأربعة.

إطفاء كل لوحات الإظهار.

تعريف المتحولات

حلقة البرنامج الرئيسي يتم فيها:

قراءة القيم المخزنة في (Data) عند اللافتة (Seg) باستخدام التعليمة (Lookup)، قيمة تلو لأخرى.

تشغيل لوحة الإظهار الأولى.

إيقاف لوحة الإظهار الأولى وتشغيل الثانية.

وهكذا....

تكرار العملية...

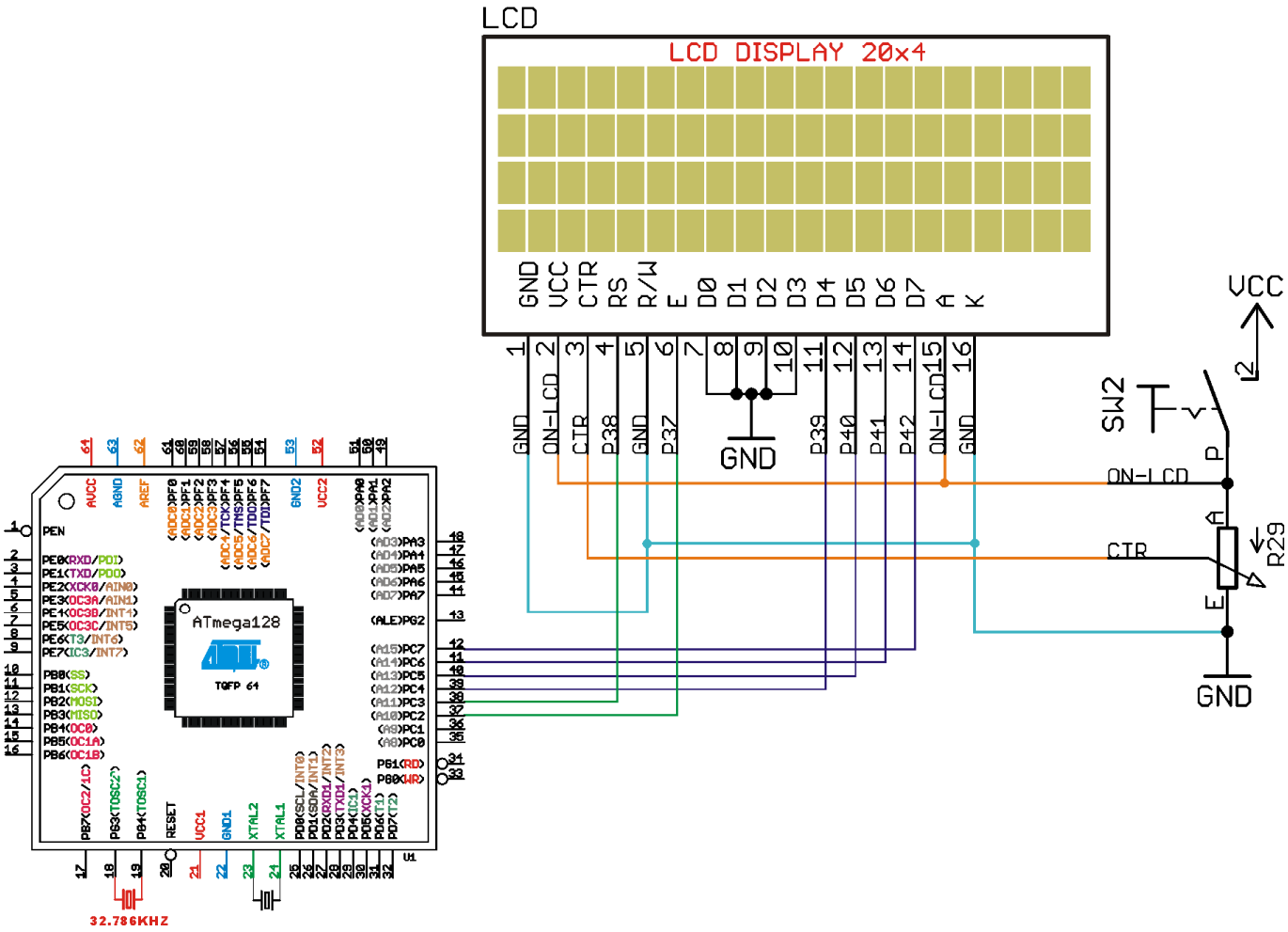
لافتة جدول تخزين البيانات.

التجربة الخامسة عشرة: ساعة توقيت الزمن الحقيقي
Exp.15: Real Time Clock

الغاية من التجربة:

برمجة وتشغيل ساعة توقيت وتاريخ في الزمن الحقيقي.

مخطط التوصيل:



متطلبات توصيل:

يجب إغلاق المفتاح SW2.

شرح عمل الدارة:

سوف نقوم بكتابة برنامج لتشغيل ساعة توقيت وعرض قيمة التوقيت والتاريخ على شاشة الإظهار الكريستالية.

إنه من أجل تشغيل وحدة التوقيت في الزمن الحقيقي (RTC) الموجودة داخل لمعالج فإنه يجب وصل هزاز كريستالي قيمته 32.786 KHZ على الأقطاب TOSC1,2.

التعليمات الجديدة:

الجدول التالي يوضح التعليمات الجديدة المستخدمة في ساعة الزمن الحقيقي الداخلية.

التعليمة البرمجية	شرح التعليمة
`Config Clock = Soft | User`	تعريف ساعة الزمن الحقيقي (داخلية | خارجي).
`Config Date = Dmy , Separator = Char`	تعيين شكل عرض التاريخ والفواصل بين اليوم والشهر والسنة
`Time$ = "hh:mm:ss"`	تعيين قيمة افتراضية للتوقيت يبدأ منها التوقيت.
`Date$ = "mm/dd/yy"`	تعيين قيمة افتراضية للتاريخ يبدأ منها التاريخ.
`Var = Time$`	قراءة القيمة الحالية للوقت
`Var = Date$`	قراءة القيمة الحالية للتاريخ

برنامج تشغيل الدارة:

```
$regfile = "m128def.dat"
$crystal = 8000000
'----------------------
Config Lcdpin = Pin , Db4 = Portc.4 , Db5 =
Portc.5 , Db6 = Portc.6 , Db7 = Portc.7 , E
= Portc.2 , Rs = Portc.3
Config Lcd = 20 * 4
'----------------------
Config Clock = Soft
Config Date = Dmy , Separator = /
Enable Interrupts

Time$ = "16:59:00"
Date$ = "14/04/08"
----------------------
Do
   Print Time$
   Print Date$

   Lcd Time$ : Locate 2 , 1 : Lcd Date$

   Wait 1
   Cls
Loop
```

Directives

Configure the LCD display

تعريف ساعة التوقيت الحقيقي الداخلية للمعالج.

تعريف نمط إظهار اليوم/الشهر/السنة

تعريف القيم الابتدائية (معايرة) للتاريخ والتوقيت

حلقة البرنامج الرئيسي يتم فيها:

إظهار التوقيت والتاريخ على النافذة التسلسلية للمعالج

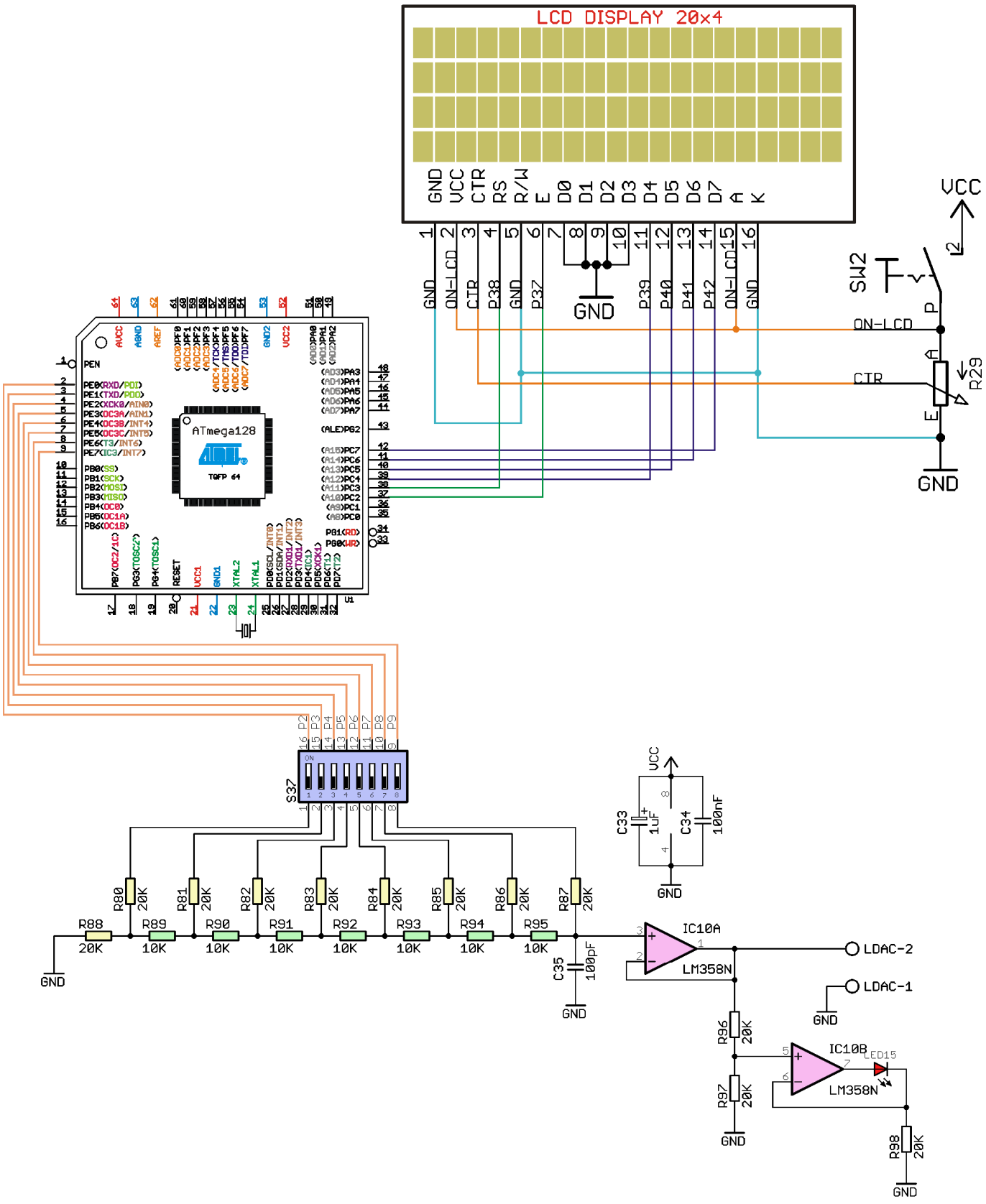
عرض التوقيت والتاريخ على شاشة الإظهار الكريستالية.

التجربة السادسة عشرة: التحويل الرقمي التشابهي
Exp.16: DAC using Ladder R-2R

الغاية من التجربة:

استخدام شبكة لادر R-2R لأغراض التحويل الرقمي التشابهي.

مخطط التوصيل:



متطلبات توصيل:

يجب إغلاق جميع نقاط المفتاح SW37 لوصل شبكة لادر مع المعالج.

يجب إغلاق المفتاح SW2 لتشغيل شاشة الإظهار الكريستالية.

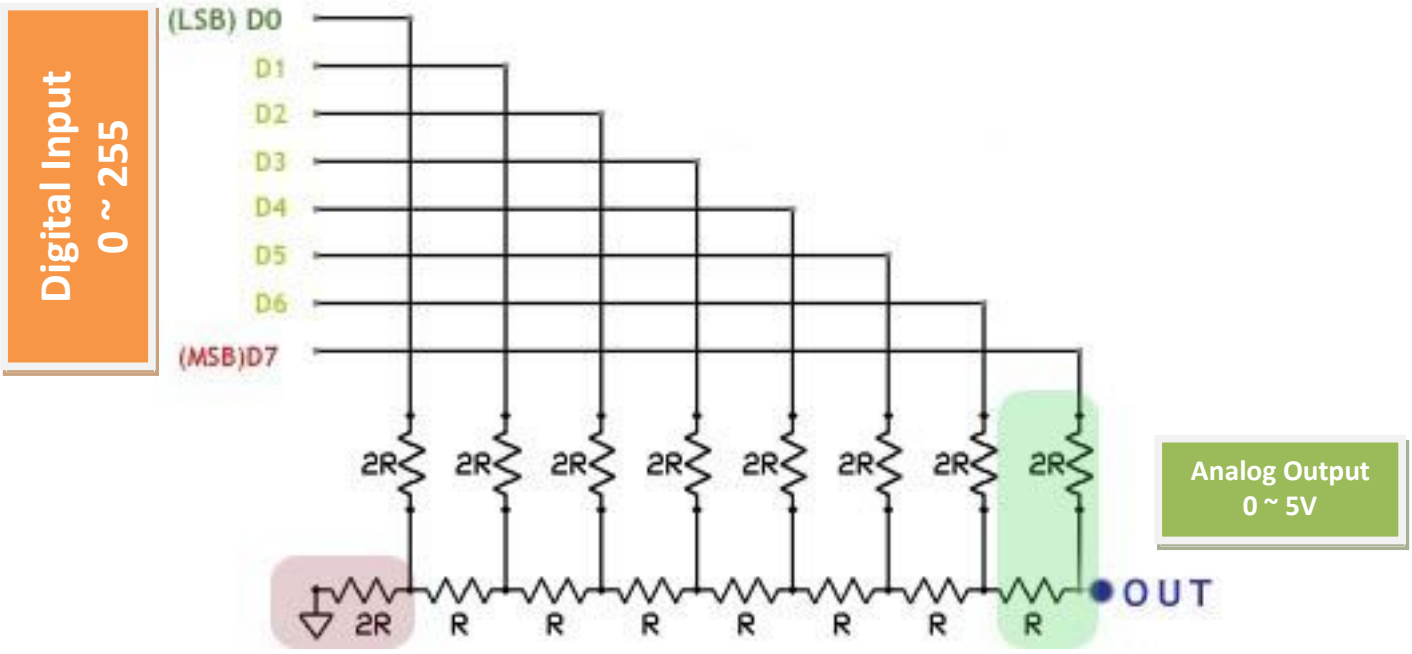
شرح عمل الدارة:

سوف نقوم بكتابة برنامج يقوم بتوليد قيمة رقمية تعبر عن جهد تشابهي يظهر على خرج شبكة لادر، وسوف نقوم بحساب قيمة هذا الجهد وعرضه على شاشة الإظهار الكريستالية.

إن الفكرة الأساسية لشبكة لادر هو إدخال قيم رقمية (0 or 1) على كل مدخل من مداخل الشبكة، والحصول في خرجها على قيمة تشابهية لها علاقة مباشرة بالقيم الرقمية المطبقة على مداخلها.

بشكل عام من أجل شبكة لادر بدقة 8-bit، فإن القيم الرقمية على مداخل الشبكة ستتراوح من 0-255 (00000000>>11111111) والذي بدوره ينتج جهداً تشابهياً خطياً في خرج الشبكة $0\text{-}5V_{DC}$.

$$V_{OUT} = V_{REF} \times VAL / 2^N$$



وبالتالي فإنه من أجل شبكة لادر بدقة 8-bit، فإن قيمة كل درجة يمكن حسابه بالعلاقة التالية:

$$V_{step} = V_M / 2^N = 5 / 255 = 0.019 \text{ Volt}$$

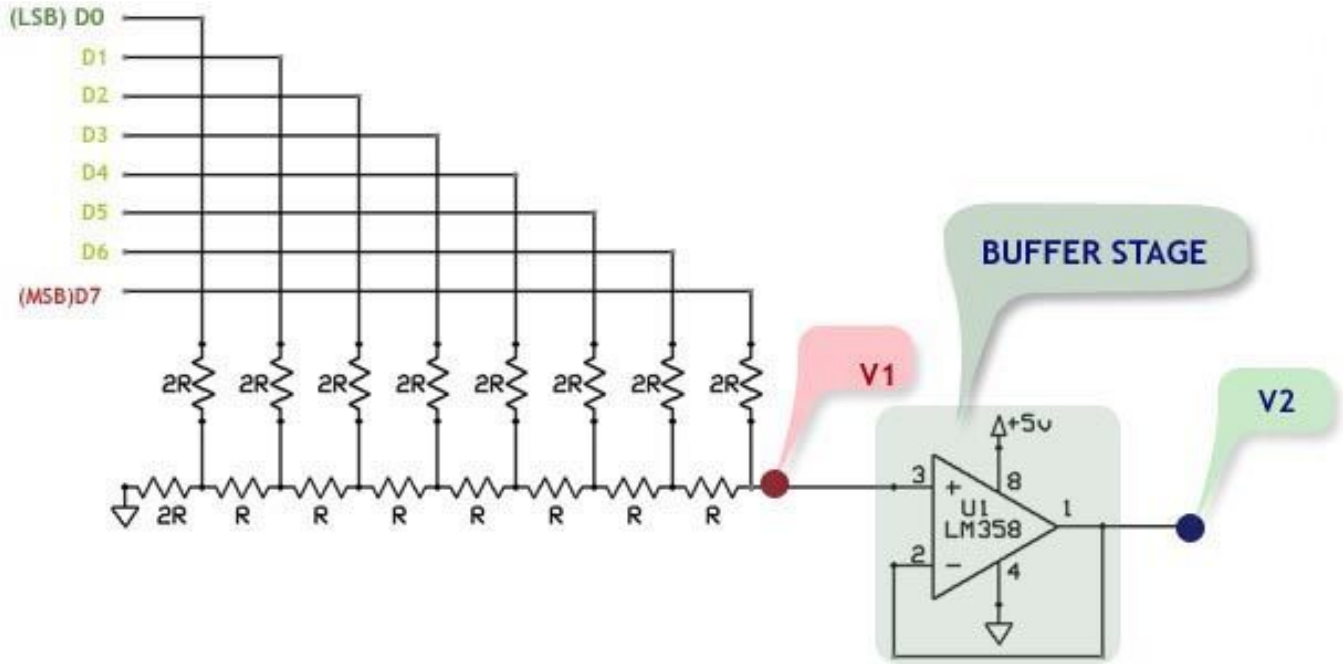
وبالتالي فإن جهد خرج الشبكة سيكون عبارة عن القيمة الرقمية على دخل الشبكة مضروباً بجهد الدرجة الواحد المحسوب أعلاه.

$$V_{out} = Dig_{val} \times V_{step}$$

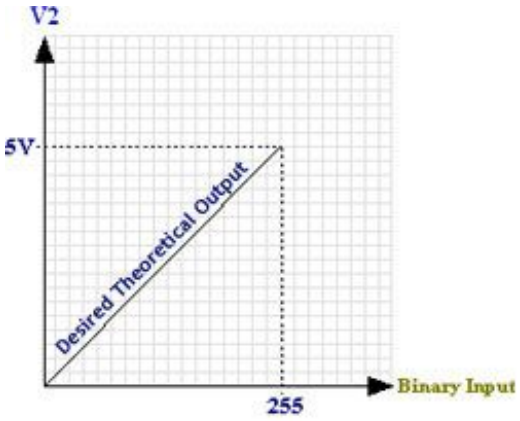
وبالتالي من أجل قيمة رقمية على دخل الشبكة ولتكن 129=0001 1000 فإن جهد الخرج يجب بالشكل:

$$129 \times 0.019 = 2.451V$$

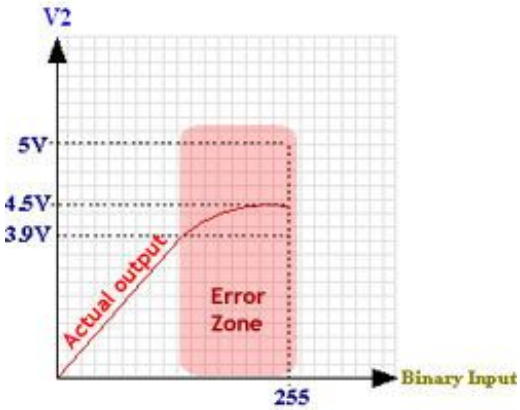
من أجل توصيل خرج شبكة لادر إلى تطبيقات حقيقية فإنه يجب إضافة دارة عزل في خرج الشبكة كما في الشكل التالي:



إن الغاية من دارة العزل هو الفصل بين الجهد في خرج الشبة (V1) والجهد الموصول مع الحمل (V2) بحيث أن استجرار التيار يكون من العازل ولا يكون من الشبكة كي لا يؤثر على عملها. ولكن سيظهر لدينا مشكلة حقيقية (عدم الخطية) نتيجة استخدام المضخم كعازل في الدارة السابقة.



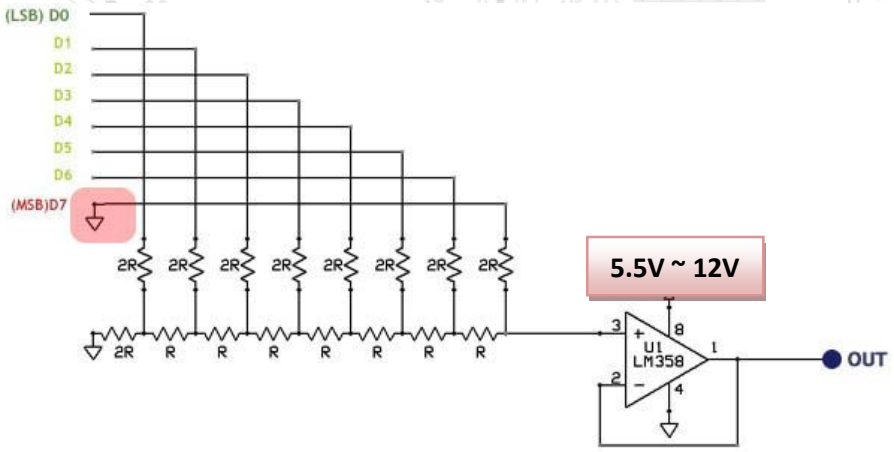
1- في حال كون جهد تغذية المضخم مساوٍ لمجال الجهد الأعظمي على دخله، فإن جهد خرج المضخم لن يكون على كامل مجال جهد الدخل للمضخم (0-to-Vcc)، وإنما سيكون أقل بقليل (Vcc - 0.5V) أي حوالي 4.5V.



2- كما أن خرج المضخم غير خطي على كامل مجال جهد الخرج للمضخم، وإنما سيكون خطياً حتى عتبة معينة تبعاً لمواصفات المضخم نفسه (في مثالنا هذا 3.9V).

يمكن التخلص من المشكلة الأولى بزيادة جهد التغذية للمضخم إلى 6.5V بدلاً من 5V.

أما المشكلة الثانية فيمكن حلها بوصل النقطة D7 إلى نقطة صفرية وبالتالي يصبح مجال الخرج 0 to 2.5V فقط.



التعليمات الجديدة:

ليس هناك تعليمات جديدة في هذه التجربة ولكن من الجيد ذكر بعض تعليمات التقريب الرياضية.

التعليمة البرمجية	شرح التعليمة

`Var = Round(x)`	إرجاع القيمة (x) مقربة إلى أقرب قيمة صحيحة. مثال:
	`Round(2.3) = 2 , Round(2.8) = 3`
	`Round(-2.3) = -2 , Round(-2.8) = -3`

`Var = Int(source)` Ex: Int(134.567)=134	إرجاع القسم الصحيح لعدد كسري
`Var = Frac(source)` Ex: Frac(134.567)=567	إرجاع القسم الكسري لعدد كسري
`Var = Fix(x)` EX: Fix(12.98)=12	تقريب عدد كسري (x) إلى القيمة الصحيحة الأصغر.
`Var = Sgn(x)`	تحديد إشارة العدد (x). إذا كانت **x<0** فإن **Var=-1** إذا كانت **x>0** فإن **Var=1**

برنامج تشغيل الدارة:

```
$regfile = "m128def.dat"
$crystal = 8000000
'-----------------------
Config Lcdpin = Pin , Db4 = Portc.4 , Db5 =
Portc.5 , Db6 = Portc.6 , Db7 = Portc.7 , E
= Portc.2 , Rs = Portc.3
Config Lcd = 20 * 4
'-----------------------
Config Porte = Output
Digital Alias Porte
'-----------------------
Dim I As Byte , V_out As Single
Const Vlot_step = 5 / 255
'-----------------------
Cls
Do
   For I = 0 To 255
      Digital = I
      V_out = I * Vlot_step
      Lcd "DAC OUT= " ; V_out
      Wait 1 : Cls
   Next I
Loop
End
```

التوجيهات.

تعريف أقطاب شاشة الإظهار الموصلة مع المتحكم.
تعريف موديل شاشة الظهار

تعريف البوابة الموصولة مع شبكة لادر.

تعريف الثوابت والمتحولات.

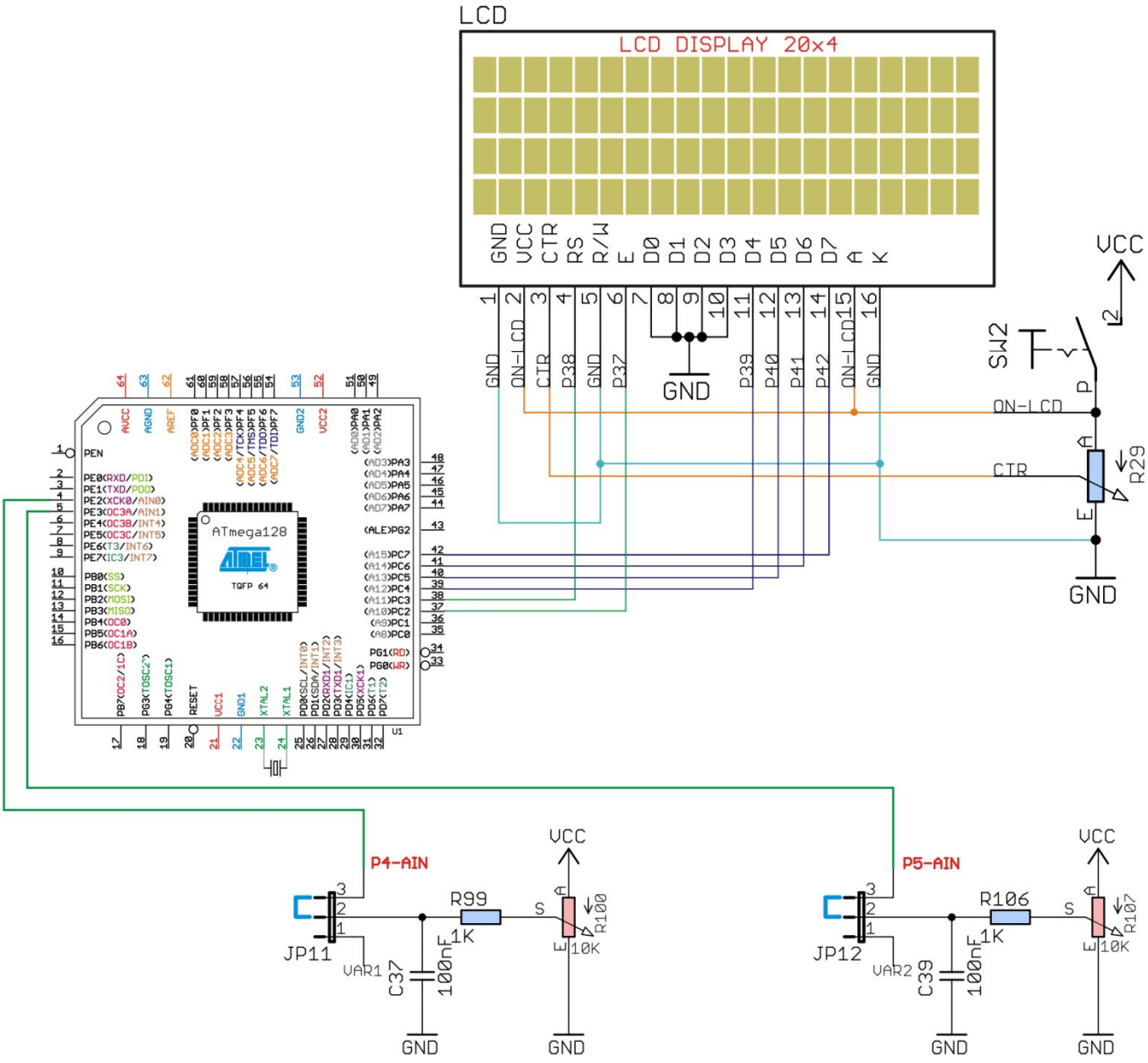
حلقة البرنامج الرئيسي يتم فيها:
توليد قيم رقمية على خرج البوابة (دخل شبكة لادر)
والتي ينشأ عنها جهد تشابهي على خرج الشبكة.
حساب قيمة جهد الخرج التشابهي وعرضها على شاشة
الإظهار الكريستالية.

التجربة السابعة عشرة: المقارن التشابهي Exp.17: Analog Comparator

الغاية من التجربة:

استثمار المقارن التشابهي الداخلي للمعالج في أغراض مقارنة الإشارة التشابهية.

مخطط التوصيل:



متطلبات توصيل:

يجب إغلاق المفتاح SW2.

يجب وضع نقطتي الوصل (Jumpers) JP11, JP12 ليت وصلها مع أقطاب المقارن التشابهي.

شرح عمل الدارة:

سوف نقوم بكتابة برنامج لتشغيل المقارن التشابهي للمعالج من أجل مقارنة إشارتين تشابهيتين مختلفتين.

في حال كون الجهد التشابهي المطبق على القطب **ANI0** (المدخل الموجب للمقارن) أكبر من الجهد المطبق على القطب **AIN1** (المدخل السالب للمقارن) فإن خرج المقارن يصبح في المستوى **High**، وأما في حال كون الجهد على القطب **ANI1** أكبر من الجهد المطبق على القطب **AIN0** فإن خرج المقارن يصبح في المستوى **Low**.

يمكن استخدام هذا المقارن في تطبيقات قياس ارتفاع أو انخفاض مستوى السوائل في خزان من أجل تشغيل أو إيقاف مضخة المياه.

إن خرج المقارن يمكن أن يستخدم لقدح مقاطعة حادثة المسك للمؤقت/عداد1 كما أن المقارن التشابهي يملك مقاطعة خاصة يمكن إعدادها لتكون على الجبهة الصاعدة أو الهابطة أو تغير الجبهة أثناء المقارنة.

التعليمات الجديدة:

الجدول التالي يوضح التعليمات الجديدة المستخدمة في ساعة الزمن الحقيقي الداخلية.

التعليمة البرمجية	شرح التعليمة
`Config Aci= On|off , Compare= On|off , Trigger = TOGGLE|RISING|FALLING`	(**Aci=On|off**): وصل | فصل المقارن التشابهي عن أقطاب نافذة المعالج. (**Compare=On|off**): من أجل "" فإن المؤقت1 في نمط حادثة المسك سيقوم بقدح المقارن التشابهي. (**Trigger**): يوجد ثلاث حالات تتفعل عندها مقاطعة المقارن: `Rising: (Ain0 > Ain1)` سوف تحدث مقاطعة عندما يصبح الجهد على القطب "`Ain0`" أكبر من الجهد على القطب "`Ain1`". `Falling: (Ain0 < Ain1)` سوف تحدث مقاطعة عندما يصبح الجهد على القطب "`Ain0`" أصغر من الجهد على القطب "`Ain1`". `Toggle: (Ain0 > Ain1 and Ain0 < Ain1)`
`Enable Aci`	تفعيل مقاطعة المقارن التشابهي.
`On Aci label`	في حال حصلت مقاطعة المقارن التشابهي إقفز إلى برنامج المقاطعة الموجود عن اللافتة "`label`".
`Start Ac`	تغذية المقارن التشابهي وبدأ عمله.
`Stop Ac`	إيقاف المقارن التشابهي وفصله عن التغذية.
`Enable Interrupts`	تفعيل علم المقاطعات العام الذي لابد منه عند استخدام أي مقاطعة لأي عنصر في المعالج.
`nop`	هذه التعليمة من تعليمات لغة التجميع "Assembly" وهي تأخير زمني بمقدار دورة آلة واحدة.

برنامج تشغيل الدارة:

التوجيهات.	`$regfile = "m128def.dat"` `$crystal = 8000000`

تعريف أقطاب شاشة الإظهار الموصلة مع المتحكم.
تعريف موديل شاشة الظهار
	`'-----------------------`
	`Config Lcdpin = Pin , Db4 = Portc.4 , Db5 = Portc.5 , Db6 = Portc.6 , Db7 = Portc.7 , E = Portc.2 , Rs = Portc.3`
	`Config Lcd = 20 * 4`

إعداد المقارن التشابهي بحيث يقدح عن تغير الجهد على مدخليه.
القفز إلى برنامج خدمة المقاطعة للمقارن عند حصول المقاطعة.
تشغيل المقارن التشابهي
	`Config Aci = On , Compare = Off , Trigger = Rising`
	`On Aci Isr_aci`
	`Enable Aci`
	`Start Ac`
	`Enable Interrupts`

	`'-----------------------` `Dim I As Byte`

حلقة البرنامج الرئيسي وهي حلقة فارغة يمكن كتابة أي روتين مرغوب.	`'-----------------------`
	`Cls`
	`Do`
	`  nop`
	`Loop`
	`End`

	`'-----------------------`

برنامج خدمة مقاطعة المقارن التشابهي
يعد مقاطعات المقارن التشابهي ويعرضها على شاشة الإظهار
	`Isr_aci:`
	`  Cls`
	`  I = I + 1`
	`  Lcd "change: " ; I`
	`  Waitms 100`
	`Return`