برمجة المتحكمات المصغرة

التجارب العملية

الجلسة الثانية

Programming

Embedded Systems Microcontroller

You Can Practice Microcontroller Programming Easily Now!

WALID BALID, Tuesday, December 15, 2009

- تعليمات التوجيهات الأساسية:

وظيفة التعليمة	شكل التعليمة
تحديد اسم المعالج المستخدم (ATmega128)	`$regfile = "m128def.dat"`
تحديد تردد الهزاز الكريستالي الذي يعمل عليه المعالج	`$crystal = 1000000`
تحديد معدل بود النقل لنافذة الاتصال التسلسلي	`$baud = 9600`

- تعليمات التأخير الزمني:

وظيفة التعليمة	شكل التعليمة
تأخير زمني (قيمة التأخير Value تعطى بالثانية)	`Wait value`
تأخير زمني (قيمة التأخير Value تعطى بالميلي ثانية)	`Waitms value`
تأخير زمني (قيمة التأخير Value تعطى بالميكرو ثانية)	`Waitus value`

- تعليمات تعريف الأقطاب (دخل/خرج) ومقاومات الرفع الداخلية:

وظيفة التعليمة	شكل التعليمة
تعريف البوابة C كبوابة خرج	`Config Portc = Output`
تعريف القطب رقم 5 من البوابة C كقطب خرج	`Config Portc.5 = Output`
تعريف البوابة C كبوابة دخل	`Config Portc = Input`
تعريف القطب رقم 5 من البوابة C كقطب دخل	`Config Pinc.5 = Input`
تفعيل مقاومات الرفع الداخلية للبوابة C	`Portc = 255`
تفعيل مقاومة الرفع الداخلية للقطب رقم 5 من البوابة C	`Pinc.5 = 1`
إلغاء تفعيل مقاومة الرفع الداخلية للقطب رقم 5 من البوابة C	`Pinc.5 = 0`
تفعيل بعض مقاومات الرفع الداخلية للبوابة C	`Portc = &B11110000`
يمكن استخدام هذا الشكل لتعريف الأقطاب من البوابة كدخل/خرج حيث أن (0) تعني قطب دخل، والـ (1) تعني قطب خرج.	`Config Portc = &B11110000`
يصرح إلى أن البوابة (D) سوف يشار إليها أثناء البرنامج بالاسم (Leds)	`Leds Alias Portd`
يصرح إلى أن القطب (5) سوف يشار إليه أثناء البرنامج بالاسم (Led)	`Leds Alias Portd.5`

ملاحظة: عند تعريف بوابة أو قطب منها كدخل، فإنه يجب تحديد الحالة الابتدائية للقيمة على البوابة وإلا ستكون حالة ممانعة عالية على قطب البوابة. إن القيمة الابتدائية يمكن تحديدها من خلال تفعيل مقاومة الرفع الداخلية لقطب الدخل أو إلغائها.

- تعليمات التعامل على مستوى البت (Set/Reset):

وظيفة التعليمة	شكل التعليمة
جعل قيمة (البت/بت من متحول المتحول) واحد منطقي	`Set bit`
جعل قيمة (البت/بت من متحول المتحول) صفر منطقي	`Reset bit`
تغيير قيمة (البت/بت من متحول المتحول) إلى الحالة المعاكسة	`Toggle bit`

- التعليمات الشرطية:

وظيفة التعليمة	شكل التعليمة

اختبار **حالة أو قيمة** متحول وتنفيذ تعليمات معينة تبعاً لنتيجة شروط الاختبار.
إذا تحقق الشرط1 فنفذ التعليمات1
وإلا إذا تحقق الشرط2 فنفذ التعليمات2
وغير ذلك نفذ التعليمات3
	`If Expression1 Then`
	`  Statements1`
	`  ...`
	`Elseif Expression2 Then`
	`  Statements2`
	`  ...`
	`Else`
	`  Statements3`
	`  ...`
	`End If`

اختبار **حالة أو قيمة** متحول وتنفيذ تعليمات معينة تبعاً لنتيجة شرط الاختبار المتحقق.
إذا كان var = Test1 فنفذ التعليمات1
إذا كان var = Test2 فنفذ التعليمات2
وغير ذلك نفذ التعليمات3
	`SELECT CASE var`
	`  Case Test1 : Statements1`
	`  Case Test2 : Statements2`
	`  Case Else : Statements3`
	`END SELECT`

- تعليمات الحلقات:

وظيفة التعليمة	شكل التعليمة

يستمر بالدوران في الحلقة وتنفيذ التعليمات الموجودة في جسم الحلقة حتى تحقق الشرط أو الخروج القسري من الحلقة.	`Do`
	`  Statements`
	`Loop [until Expression]`

تنفيذ جملة من التعليمات طالما أن الشرط محقق.	`While Condition`
	`  Statements`
	`Wend`

تنفيذ جملة من التعليمات عدداً من المرات يبدأ من القيمة Start وينتهي عند القيمة End. يمكن تحديد خطوة العد بالمتحول step.	`For Var = Start To End [step Value]`
	`  Statements`
	`Next Var`

خروج قسري من الحلقة For
خروج قسري من الحلقة Do
خروج قسري من الحلقة While
	`Exit For`
	`Exit Do`
	`Exit While`

- تعليمات تعريف المتحولات في الذاكرة SRAM:

وظيفة التعليمة	شكل التعليمة
تعريف متحول عددي نوع بت (0 or 1).	`Dim Var1 As Bit`
تعريف متحول عددي نوع بايت (0 to 255).	`Dim Var2 As Byte`
تعريف متحول عددي صحيح (-32,768 to +32,767).	`Dim Var3 As Integer`
تعريف متحول عددي نوع وورد (0 to 65535).	`Dim Var4 As Word`
تعريف متحول عددي طويل (-2147483648 to 2147483647).	`Dim Var5 As Long`
تعريف متحول عددي مؤشر (1.5 x 10^-45 to 3.4 x 10^38).	`Dim Var6 As Single`
تعريف متحول عددي مؤشر مضاعف.	`Dim Var7 As Double`
تعريف متحول نوع محرف محدد المحارف بـ (chr_num *).	`Dim Var8 As String * 1`
تعريف مصفوفة بثمان بايتات.	`Dim Array(8) As Byte`
تعريف متحول رقمي ثابت.	`Const Symbol = Numconst` `Ex. Const Pi = 3.14159265358979`
تعريف متحول محرف ثابت.	`Const Symbol = Stringconst` `Ex. Const S = "TEST"`
تعريف تعبير رياضي ثابت.	`Const Symbol = Expression` `Ex. Const E =(b1 * 3) + 2`
تعريف متحول محلي في برنامج فرعي أو برنامج فرعي وظيفي.	`Local Var As Type`

- تعليمات قراءة حالة المداخل:

وظيفة التعليمة	شكل التعليمة
يراقب حالة القطب المحدد في $P_{x.y}$ كلما مر عليه وعندما تصبح حالته موافقة للحالة المحددة في **state**، سوف يقفز إلى البرنامج الفرعي عند اللافتة **label** وينفذ البرنامج ويعود.	`Debounce Px.y , state , label , Sub` `Ex. Debounce Key1 , 0 , Sw1 , Sub`
تهيئة زمن تأخير (ميلي ثانية) عن استعمال تعليمة **Debounce** للتخلص من العطالة الميكانيكية للمفتاح.	`Config Debounce = time`
سوف يقف البرنامج عند هذه التعليمة وينتظر أن تصبح حالة البت (القطب) صفر أو واحد منطقي عندها يكمل البرنامج.	`Bitwait x , Set/reset` `Ex. Bitwait Pinb.7 , reset`

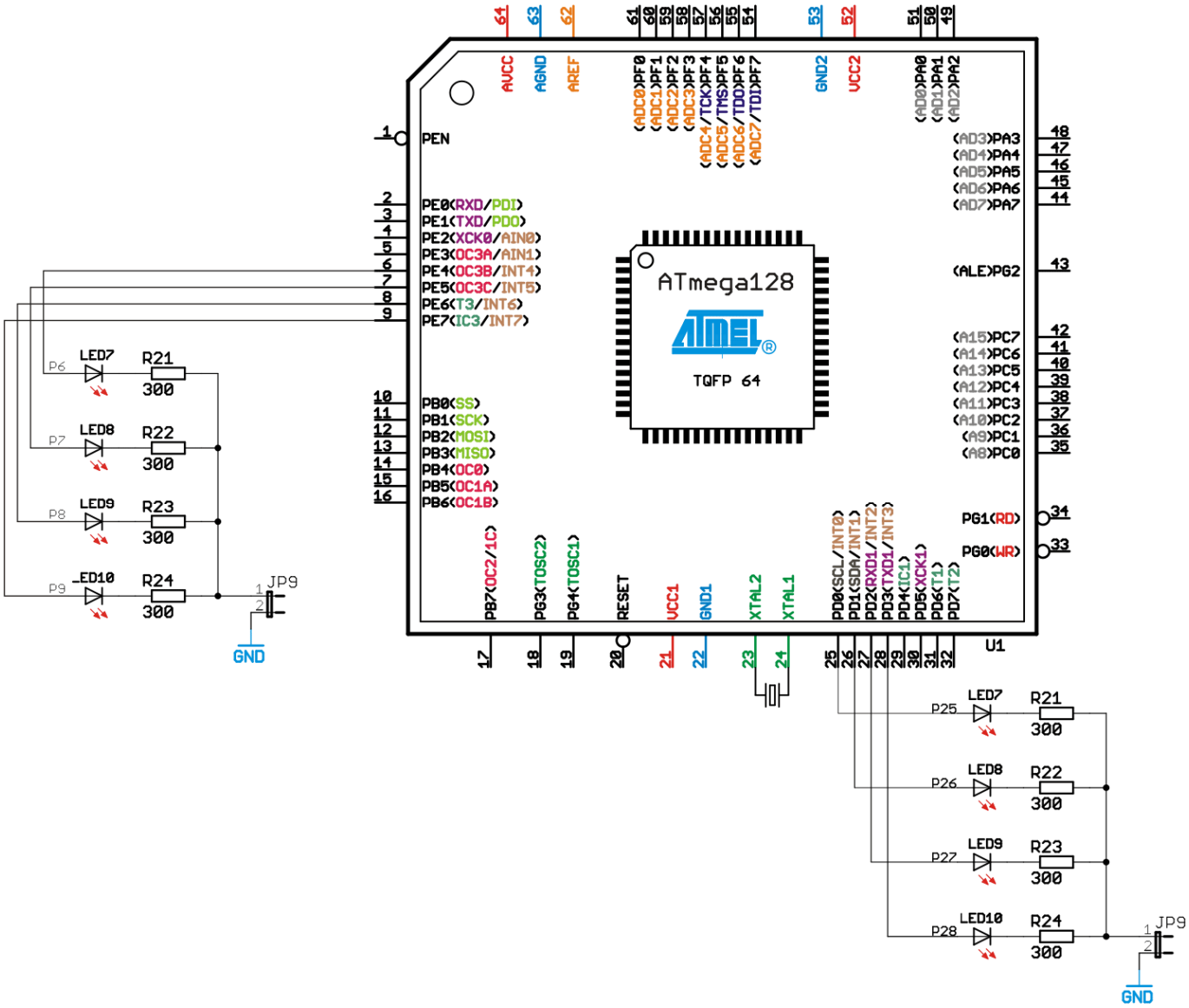
Exp.1: Scrolling LEDs

التجربة الأولى: الثنائيات الضوئية المتحركة

الغاية من التجربة:

استثمار بوابات المتحكم كبوابات خرج رقمي من أجل تشكيل حركة ضوئية باستخدام الثنائيات ضوئية.

مخطط التوصيل:



متطلبات توصيل:

يجب قصر الوصلة JP9, JP10.

شرح عمل الدارة:

سوف نقوم بكتابة برنامج من أجل تشكيل حركة للثنائيات الضوئية بحيث أنه سوف يتم تشغيلها كما يلي:

1- تشغيل LEDs7.
2- انتظار مدة تأخير زمني ثانية.
3- تشغيل الثنائي التالي... وهكذا حتى يتم تشغيل جميع الثنائيات.

4- بعدها يتم إطفاء الثنائيات الضوئية بشكل معاكس.

برنامج تشغيل الدارة:

التوجيهات.	`$regfile = "m128def.dat"` `$crystal = 1000000`

تعريف الأقطاب.	`'-----------------------`
	`Config Portd.0 = Output`
	`Config Portd.1 = Output`
	`Config Portd.2 = Output`
	`Config Portd.3 = Output`
	
	`Config Porte.4 = Output`
	`Config Porte.5 = Output`
	`Config Porte.6 = Output`
	`Config Porte.7 = Output`

الإشارة إلى الأسماء الفيزيائية للبوابات أثناء البرنامج بأسماء حقيقية.	`'-----------------------`
	`Leds1 Alias Portd`
	`Leds2 Alias Porte`

تعريف المتحولات.	`'-----------------------` `Dim I As Byte , J As Byte`

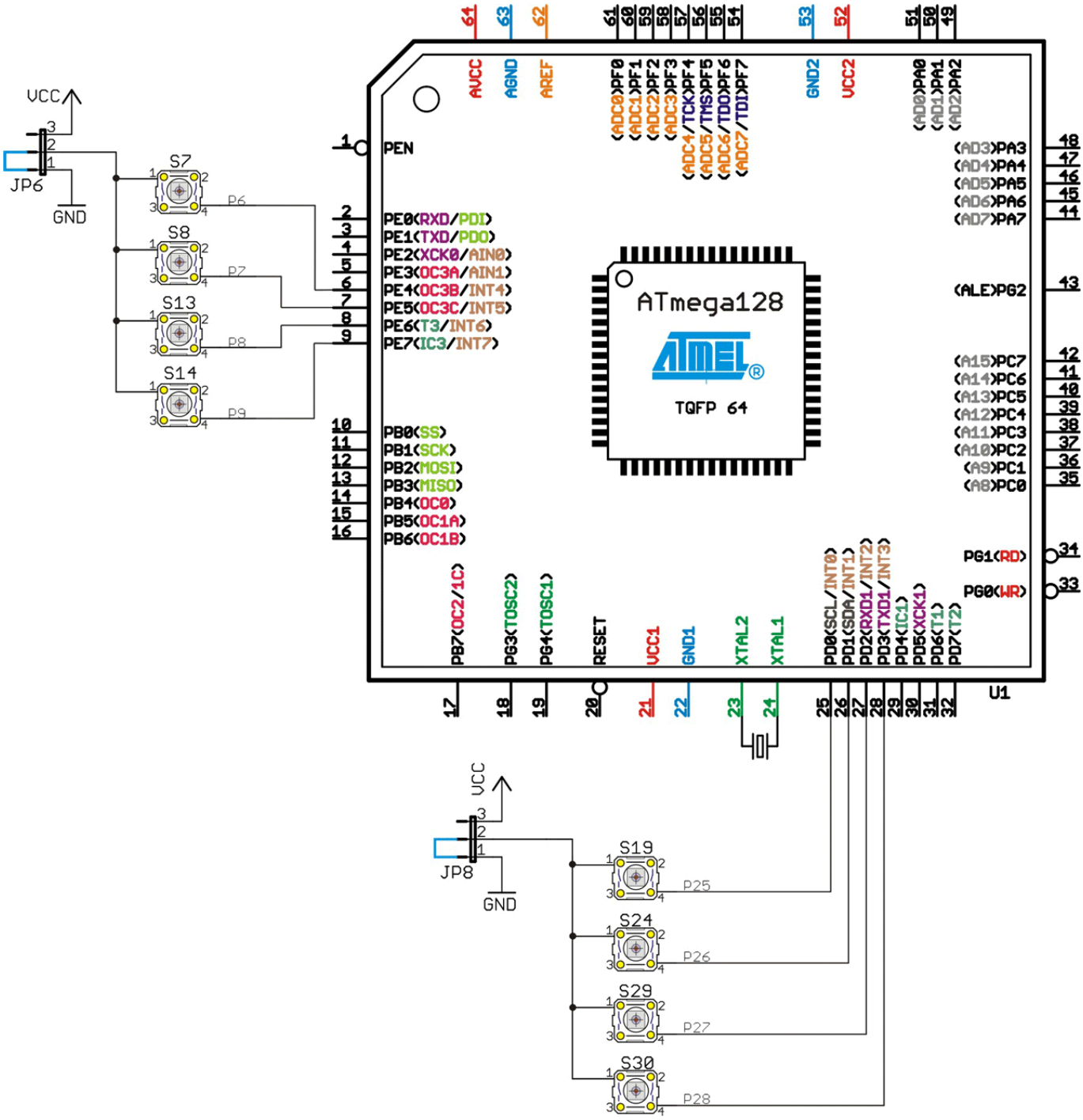
حلقة تشغيل الثنائيات الضوئية بالتتالي وبفواصل زمنية واحد ثانية.	`'-----------------------`
	`Do`
	`  For I = 0 To 3`
	`  J = I + 4`
	`  Set Leds1.i`
	`  Set Leds2.j`
	`  Wait 1`
	`  Next I`

حلقة إطفاء الثنائيات الضوئية عكسياً وبالتتالي وبفواصل زمنية واحد ثانية.	`  For I = 3 To 0 Step -1`
	`  J = I + 4`
	`  Reset Leds1.i`
	`  Reset Leds2.j`
	`  Wait 1`
	`  Next I`
	`Loop`
	`End`

الغاية من التجربة:

استثمار أقطاب المتحكم كأقطاب دخل رقمي من أجل قراءة حالة مفاتيح لحظية.

مخطط التوصيل:



متطلبات توصيل:

يجب قصر الوصلة JP6, JP8.

شرح عمل الدارة:

سوف نقوم بكتابة برنامج من أجل قراءة حالة المفاتيح اللحظية في حالتين:

1- الوصلات JP6, JP8 موصلة إلى GND (قراءة الجبهة الهابطة على القطب).

2- الوصلات JP8 موصلة إلى GND, JP6 موصلة إلى VCC.

برنامج تشغيل الدارة في حالة JP6, JP8 موصلة إلى GND:

`$regfile = "m128def.dat"`
`$crystal = 1000000`
`$baud = 4800`
	التوجيهات.

`'----------------------` `Config Debounce = 150`	تحديد التأخير الزمني للقراءة الجديدة لحالة المفتاح

`Config Pind.0 = Input`
`Config Pind.1 = Input`
`Config Pind.2 = Input`
`Config Pind.3 = Input`
 	
`Config Pine.4 = Input`
`Config Pine.5 = Input`
`Config Pine.6 = Input`
`Config Pine.7 = Input`
	تعريف الأقطاب.

`Portd.0 = 1 : Portd.1 = 1`
`Portd.2 = 1 : Portd.3 = 1`
`Porte.4 = 1 : Porte.5 = 1`
`Porte.6 = 1 : Porte.7 = 1`
	تفعيل المقاومات الداخلية

`'----------------------`
`Key1 Alias Pind.0`
`Key2 Alias Pind.1`
`Key3 Alias Pind.2`
`Key4 Alias Pind.3`
`Key5 Alias Pine.4`
`Key6 Alias Pine.5`
`Key7 Alias Pine.6`
`Key8 Alias Pine.7`
`'----------------------`
	الإشارة إلى الأسماء الفيزيائية للبوابات أثناء البرنامج بأسماء حقيقية.

`Do`
`Debounce Key1 , 0 , Sw1 , Sub`
`Debounce Key2 , 0 , Sw2 , Sub`
`Debounce Key3 , 0 , Sw3 , Sub`
`Debounce Key4 , 0 , Sw4 , Sub`
`Debounce Key5 , 0 , Sw5 , Sub`
`Debounce Key6 , 0 , Sw6 , Sub`
`Debounce Key7 , 0 , Sw7 , Sub`
`Debounce Key8 , 0 , Sw8 , Sub`
 	
`Waitms 200`
	حلقة البرنامج الرئيسي
	قراءة حالة المفاتيح الثمانية

`Loop`
`End`
`'----------------------`	

```
Sw1:
   Print "Switch(1) is pressed!"
Return
'/------------
Sw2:
   Print "Switch(2) is pressed!"
Return
'/------------
Sw3:
   Print "Switch(3) is pressed!"
Return
'/------------
Sw4:
   Print "Switch(4) is pressed!"
Return
'/------------
Sw5:
   Print "Switch(5) is pressed!"
Return
'/------------
Sw6:
   Print "Switch(6) is pressed!"
Return
'/------------
Sw7:
   Print "Switch(7) is pressed!"
Return
'/------------
Sw8:
   Print "Switch(8) is pressed!"
Return
```

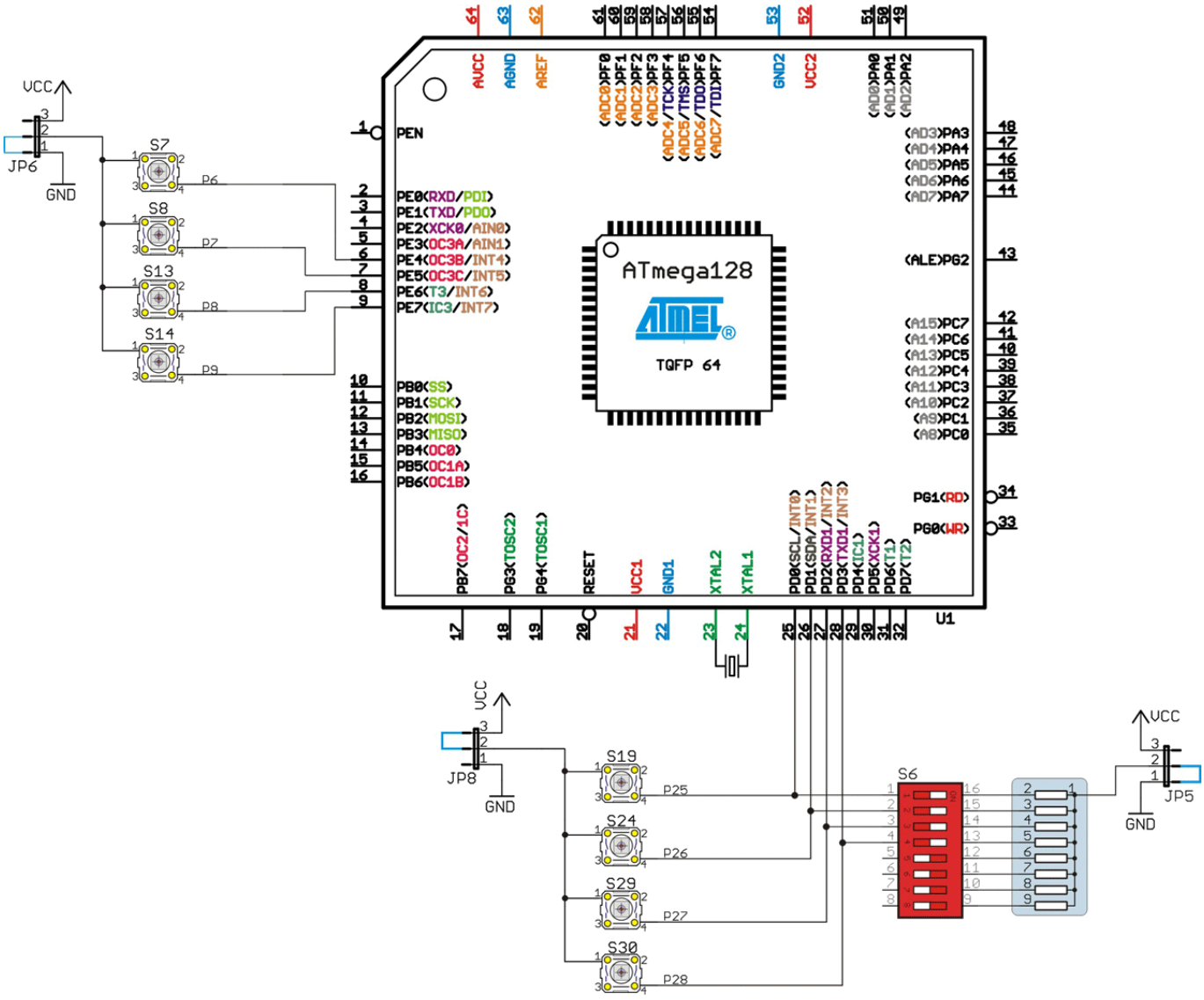
البرامج الفرعية

طباعة اسم المفتاح المضغوط على النافذة التسلسلية

والعودة إلى حلقة البرنامج الرئيسية

برنامج تشغيل الدارة في حالة JP6, موصلة إلى GND وJP8 موصلة إلى VCC:

في هذه الحالة فإنه يجب إلغاء تفعيل مقاومات الرفع الداخلية للبوابة PortD وتفعيل مقاومات سحب إلى النقطة الأرضية باستخدام مفاتيح توصيل المقاومات الخارجية S6.




```
$regfile = "m128def.dat"
$crystal = 1000000
$baud = 4800
'-----------------------
Config Debounce = 150

Config Pind.0 = Input
Config Pind.1 = Input
Config Pind.2 = Input
Config Pind.3 = Input

Config Pine.4 = Input
Config Pine.5 = Input
Config Pine.6 = Input
Config Pine.7 = Input
```

التوجيهات.

تحديد التأخير الزمني للقراءة الجديدة لحالة المفتاح

تعريف الأقطاب.

```
Portd.0 = 0 : Portd.1 = 0
Portd.2 = 0 : Portd.3 = 0
Porte.4 = 1 : Porte.5 = 1
Porte.6 = 1 : Porte.7 = 1
'-----------------------
Key1 Alias Pind.0
Key2 Alias Pind.1
Key3 Alias Pind.2
Key4 Alias Pind.3
Key5 Alias Pine.4
Key6 Alias Pine.5
Key7 Alias Pine.6
Key8 Alias Pine.7
'-----------------------
Do
   Debounce Key1 , 0 , Sw1 , Sub
   Debounce Key2 , 0 , Sw2 , Sub
   Debounce Key3 , 0 , Sw3 , Sub
   Debounce Key4 , 0 , Sw4 , Sub
   Debounce Key5 , 0 , Sw5 , Sub
   Debounce Key6 , 0 , Sw6 , Sub
   Debounce Key7 , 0 , Sw7 , Sub
   Debounce Key8 , 0 , Sw8 , Sub
Waitms 200
Loop
End
'-----------------------
Sw1:
   Print "Switch(1) is pressed!"
Return
'/------------
Sw2:
   Print "Switch(2) is pressed!"
Return
'/------------
Sw3:
   Print "Switch(3) is pressed!"
Return
'/------------
Sw4:
   Print "Switch(4) is pressed!"
Return
'/------------
Sw5:
   Print "Switch(5) is pressed!"
Return
'/------------
Sw6:
   Print "Switch(6) is pressed!"
Return
'/------------
Sw7:
   Print "Switch(7) is pressed!"
Return
'/------------
Sw8:
   Print "Switch(8) is pressed!"
Return
```

إلغاء تفعيل المقاومات الداخلية للبوابة Portd

تفعيل المقاومات الداخلية للبوابة PortE

الإشارة إلى الأسماء الفيزيائية للبوابات أثناء البرنامج بأسماء حقيقية.

حلقة البرنامج الرئيسي

قراءة حالة المفاتيح الثمانية

البرامج الفرعية

طباعة اسم المفتاح المضغوط على النافذة التسلسلية

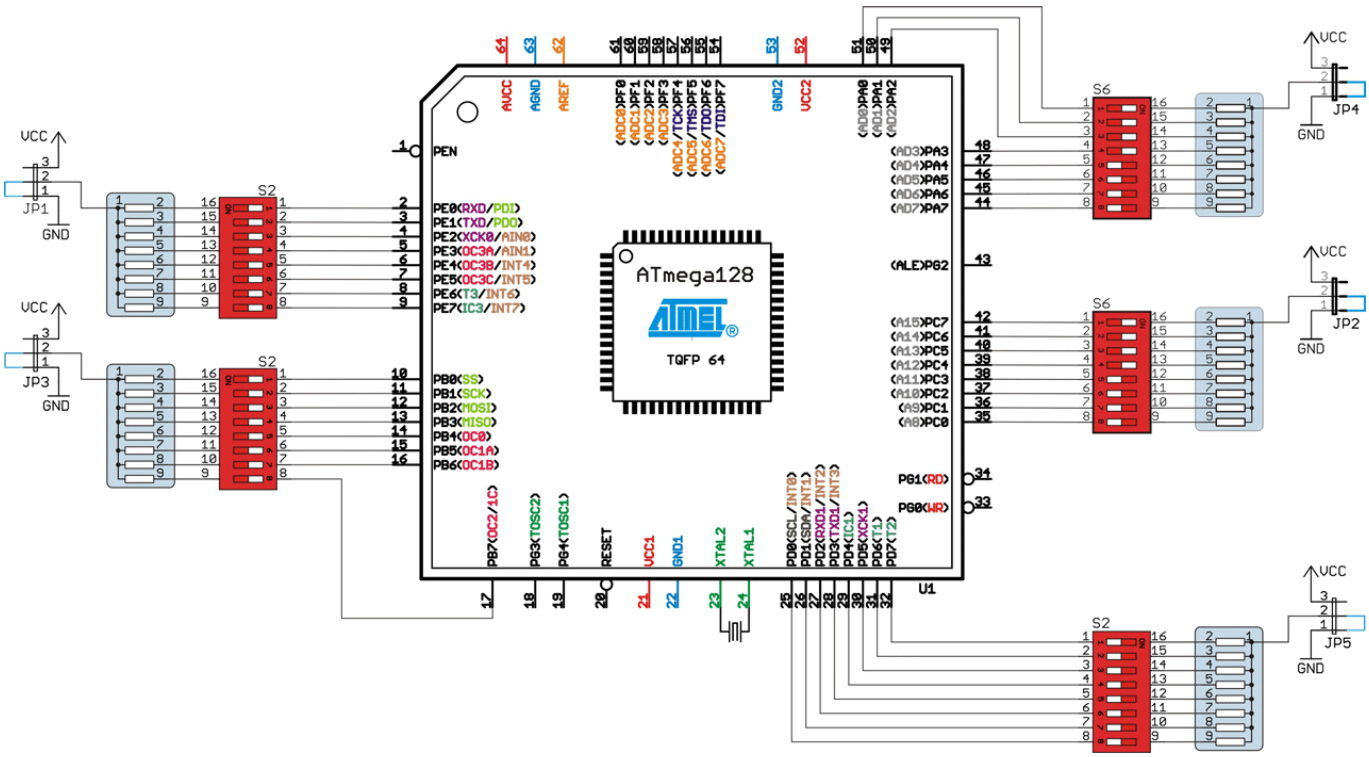
والعودة إلى حلقة البرنامج الرئيسية

التجربة الثالثة: استثمار توسعات البوابات (دخل/خرج) Exp.3: Expansion Connectors

الغاية من التجربة:

استثمار توسعات البوابات من أجل أغراض الدخل – الخرج الرقمي وتفعيل/إلغاء مقاومات الرفع خارجية.

مخطط التوصيل:



متطلبات توصيل:

يجب قصر الوصلات JP1~ JP6 حسب حالة المقاومة الخارجية للبوابة (رفع – سحب).

شرح عمل الدارة:

سوف نقوم بكتابة برنامج نعرف فيه جميع البوابات أعلاه كبوابات دخل رقمي ومن ثم سنقوم بقراءة القيم على مداخل البوابات عند تغيير قيمة الدخل باستخدام مفاتيح توصيل المقاومات الخارجية (S) في حالتين:

1- المقاومات الخارجية موصلة إلى النقطة VCC (مقاومات رفع للبوابات).

2- المقاومات الخارجية موصلة إلى النقطة GND (مقاومات سحب للبوابات).

- برنامج تشغيل الدارة في حالة المقاومات الخارجية موصلة إلى النقطة GND (مقاومات سحب للبوابات):

`$regfile = "m128def.dat"`
`$crystal = 1000000`
`$baud = 4800`
	التوجيهات.

`Config Porte = Input`
`Config Portb = Input`
`Config Portd = Input`
`Config Portc = Input`
`Config Porta = Input`
	تعريف الأقطاب.

`Porta = 255`
`Portb = 255`
`Portc = 255`
`Portd = 255`
`Porte = 255`
	تفعيل المقاومات الداخلية للبوابات

`Pa_value Alias Pina`
`Pb_value Alias Pinb`
`Pc_value Alias Pinc`
`Pd_value Alias Pind`
`Pe_value Alias Pine`
	الإشارة إلى الأسماء الفيزيائية للبوابات أثناء البرنامج بأسماء حقيقية.

`Do`
`Print "PA= " ; Pa_value`
`Print "PB= " ; Pb_value`
`Print "PC= " ; Pc_value`
`Print "PD= " ; Pd_value`
`Print "PE= " ; Pe_value`
`Print "------------"`
`Wait 1`
`Loop`
`End`
	حلقة البرنامج الرئيسي
	قراءة قيمة الدخل على البوابات وطباعتها.

- برنامج تشغيل الدارة في حالة المقاومات الخارجية موصلة إلى النقطة VCC (مقاومات رفع للبوابات):

التوجيهات.	`$regfile = "m128def.dat"`
	`$crystal = 1000000`
	`$baud1 = 4800`

تعريف الأقطاب.	`'-----------------------`
	`Config Porte = Input`
	`Config Portb = Input`
	`Config Portd = Input`
	`Config Portc = Input`
	`Config Porta = Input`

إلغاء تفعيل المقاومات الداخلية للبوابات	`'-----------------------`
	`Porta = 0`
	`Portb = 0`
	`Portc = 0`
	`Portd = 0`
	`Porte = 0`

الإشارة إلى الأسماء الفيزيائية للبوابات أثناء البرنامج بأسماء حقيقية.	`'-----------------------`
	`Pa_value Alias Pina`
	`Pb_value Alias Pinb`
	`Pc_value Alias Pinc`
	`Pd_value Alias Pind`
	`Pe_value Alias Pine`

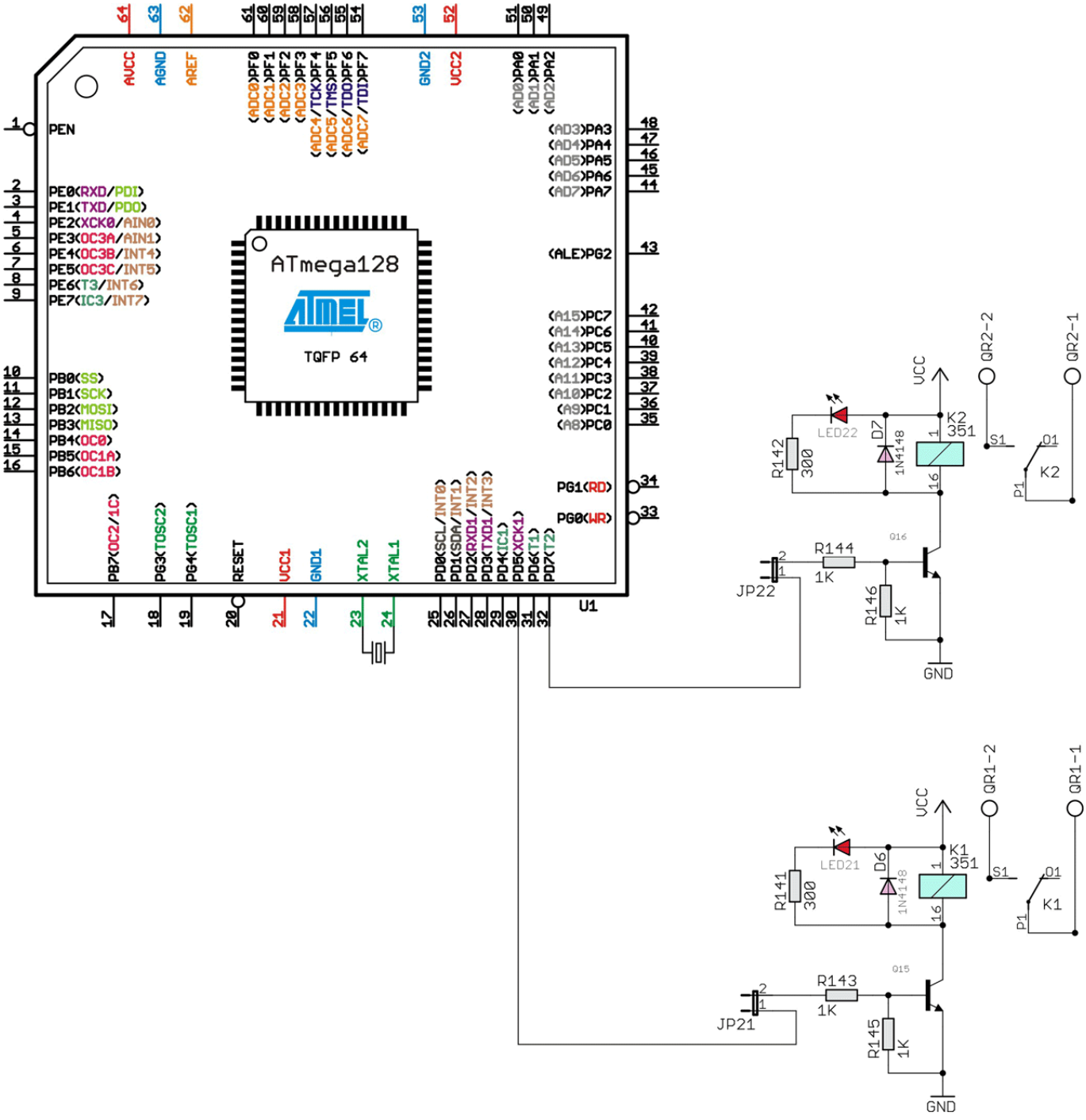
حلقة البرنامج الرئيسي
قراءة قيمة الدخل على البوابات وطباعتها.
	`'-----------------------`
	`Open "com2:" For Binary As #1`
	
	`Do`
	`  Print #1 , "PA= " ; Pa_value`
	`  Print #1 , "PB= " ; Pb_value`
	`  Print #1 , "PC= " ; Pc_value`
	`  Print #1 , "PD= " ; Pd_value`
	`  Print #1 , "PE= " ; Pe_value`
	`  Print #1 , "-----------"`
	`  Wait 1`
	`Loop`
	
	`Close #1`
	`End`
	`'-----------------------`

Exp.4: Controlling Relay (On/Of) التجربة الرابعة: تشغيل زمني لمخارج تحكم ريليه

الغاية من التجربة:

استثمار مخارج بوابات المتحكم الرقمي للتحكم بعمل ريليه استطاعية.

مخطط التوصيل:



متطلبات توصيل:

يجب قصر الوصلات JP21 ~JP22.

شرح عمل الدارة:

سوف نقوم بكتابة برنامج نعرف فيه البوابة Portd.5 والبوابة Portd.7 كبوابات خرج رقمي للتحكم بعمل الريليه K1 والريليه K2 على التوالي.

برنامج تشغيل الدارة:

التوجيهات.	`$regfile = "m128def.dat"` `$crystal = 1000000`

تعريف الأقطاب.	`'-----------------------`
	`Config Portd.5 = Output`
	`Config Portd.7 = Output`

الإشارة إلى الأسماء الفيزيائية للبوابات أثناء البرنامج بأسماء حقيقية.	`'-----------------------`
	`Relay1 Alias Portd.5`
	`Relay2 Alias Portd.7`

تعريف المتحولات	`'-----------------------` `Dim I As Byte`

حلقة البرنامج الرئيسي.	`'-----------------------`
	`For I = 0 To 9`
	`Set Relay1`
	`Wait 2`
	`Set Relay2`
	
	`Wait 2`
	
	`Reset Relay1`
	`Wait 2`
	`Reset Relay2`
	`Next I`
	`End`
	`'-----------------------`