التحكم الرقمي بواسطة المخرج التفرعي LPT

م . بشار دياب

هذا المشروع هو باكورة سلسلة من المشاريع التي ستنشر تباعاً في أعداد هذه المجلة بإذن الله وستهتم بمواضيع التحكم الآلي والأتمتة الصناعية باستخدام الحاسب (PC) أو المتحكمات (Microcontrollers) .

سنحرص على جعل هذه المشاريع متنوعة من حيث التطبيقات (عامة أو خاصة) ، كذلك ستكون متنوعة من حيث درجة التعقيد بحيث تغطي اهتمامات شرائح واسعة من الهواة والطلاب والباحثين على حد سواء .

لابد من التذكير ببعض المعلومات الاساسية في نظم العد (الثنائي والسداسي عشر والعشري) والبوابات المنطقية وبعض المعلومات البرمجية في إحدى اللغات (Pascal - C - Basic) أو اللغات المطورة عنها ضمن بيئة ويندوز .

سيتم طرح هذه المواضيع بشكل تدريجي مع صدور أعداد جديدة للمجلة .

يمكن تنفيذ التحكم من خلال الحاسب بواسطة كروت الربط (Expansion Cards) أو البوابة التفرعية (Parallel Port) أو البوابة التسلسلية (Serial Port) ولكل طريقة ميزاتها . فالتحكم عن طريق كروت الربط الداخلية يخدم تطبيقات معقدة ويستثمر إمكانيات الحاسب بشكل كبير لكن يتطلب تصميم دارات ذات سوية أعلى وكذلك الأمر بالنسبة للبرمجيات .

أما بالنسبة للبوابة التسلسلية فهي تخدم عالم الإتصالات وبعض الطرفيات والأجهزة مثل المودم والكاميرات الخ ....

وتعتمد المبدأ التسلسلي بإرسال المعلومات واستقبالها أي بشكل بتات متتالية ومتلاحقة .

وهذا يفيد في حال المسافات الطويلة ويخفض من تكاليف الوصل . وهذه البوابات تسمى (COM1 - COM2) وتعتمد بروتوكول الاتصال RS232 .

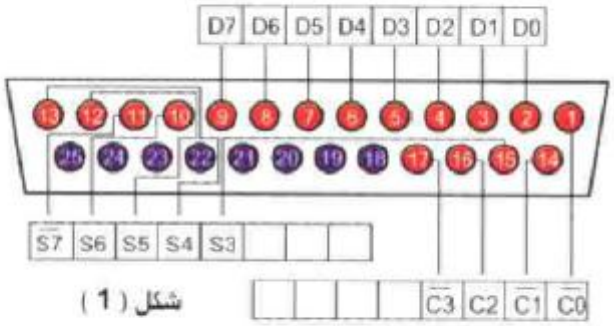
أما البوابة التفرعية فتتميز بوجود عدة خطوط لنقل البيانات ، بحيث يتم إرسال الـ Byte المطلوب دفعة واحدة وتتعامل بالسويات المنطقية الأساسية TTL . ويمكن توظيفها لمشاريع تطبيقية جيدة وسهلة التصميم ، وهي تختصر بعض الدارات المطلوبة من أجل تصميم كروت الحاسب إذا ما تم تنفيذ الدارة نفسها بشكل كرت حاسب .

وسيتم لاحقاً تنفيذ مشاريع مختلفة بحيث تشمل الطرق الثلاثة آنفة الذكر إضافة إلى موضوع المايكروكنترولر .

نطرح فيما يلي مشروعاً بسيطاً كنموذج للتخاطب مع بوابة الحاسب التفرعية (مرفق المخطط النظري للدارة إضافة للمخطط العملي PCB وكافة الملفات المطلوبة) . كذلك برنامج التحكم المنفذ بلغة Delphi والتي تؤمن إمكانيات عالية لهذا الغرض .

- خطوط البوابة التفرعية الموجودة خلف الحاسب موضحة بالشكل (1) .

وهو عبارة عن مخرج (25 pin Female) ، ويتألف من ثلاث بوابات تفرعية تم تجميع خطوطها ضمن مخرج واحد كما هو موضح بالشكل .



شكل (1)

البوابة الأولى : عبارة عن بوابة من ثمانية خطوط وهي بشكل مسجل (Latch) أي تقوم بمسك القيمة الأخيرة فيها والتي تم ارسالها إليها بحيث تبقى فعالة فيها حتى يتم استبدالها بقيمة جديدة ، وخطوطها هي (9....2) .

البوابة الثانية : عبارة عن بوابة من أربعة خطوط وهي أيضاً بشكل مسجل (Latch) كما في السابقة لكن يجب الانتباه إلى أن بعض هذه الخطوط معكوس (تحوي عاكس داخلي) حيث يظهر عليها عكس القيمة المرسلة إليها وهذه الخطوط تحوي خط فوق اسم التماس كما في الشكل (1) . أرقام تماساتها ضمن المخرج هو (1,14,16,17) . ويمكن تلافي ذلك بوضع عاكس خارجي ضمن الدارة أو عكس القيمة المرسلة ضمن البرنامج .

البوابة الثالثة : عبارة عن بوابة من خمسة خطوط فيها الخط الأخير فقط معكوس وأرقامها (10,11,12,13,15) . الخطوط من 18 إلى 25 تشكل أرضي الحاسب نفسه وهو الموصول إلى أرضي الدارة الخارجية .

يجب قبل التعامل مع مخرج الطابعة أو البوابة الثلاثة التي أتينا على ذكرها أن نتذكر دوماً أن لكل بوابة عنوان خاص يميزها عن غيرها ، هذا العنوان مشابه من حيث المبدأ للعنوان البريدي لشخص ما أو عنوان بريده الإلكتروني . هذا العنوان يصبح البوابة بحيث يتشابه مع عنوان بوابة أخرى وهو ثابت من أجل بوابة معينة .

إذا العنوان عبارة عن رقم يحدد البوابة التي سيتم إرسال البيانات إليها أو استقبالها منها . هذا الرقم يتم إصداره من قبل المعالج المركزي في الحاسب (CPU) والذي بدوره يقرر مع أي بوابة سيتم التعامل سواء في حالة القراءة منها أو الكتابة عليها (إرسال المعلومات إليها) .

تملك البوابات الثلاثة المذكورة في مخرج الطابعة العناوين التالية :

بوابة الخرج 8bit : العنوان 378 H

بوابة الخرج 4bit : العنوان 37A H

بوابة الدخل 5bit : العنوان 379 H

هذه القيم هي بالنظام السداسي عشر (Hex) .

التعامل مع البوابة :

سنقوم الآن بشرح الآلية البرمجية للتخاطب مع بوابة حاسب ضمن لغة باسكال (الأمر مشابه في لغة C) .

```
PORT[Address Num] := Data 8bit
```

حيث يمثل Address Num : رقم البوابة (عنوانها) ويمثل Data 8bit : القيمة المرسلة وهي بشكل Bit ويمكن كتابة القيمتين السابقتين بأي نظام عددي مع الانتباه إلى ان كتابة اي رقم في الباسكال بالنظام السداسي عشر يجب أن يسبق بالرمز ($) وبغير ذلك كتابته بالعشري .

أما في لغة C فيمكن الكتابة بالنظام الثنائي أو الثماني أو العشري أو السداسي عشر .

أما القراءة من البوابة التفرعية فيتم بالشكل :

```
X:= PORT[Address Num];
```

حيث X هو متحول برمجي (8bit) وهو يمثل القيمة الواردة من البوابة التي تم تحديد عنوانها .

أما في لغات Delphi و C ضمن الويندوز فيتم استخدام لغة الآلة (Assembly) بالشكل التالي :

```
im
ov      DX , Address Num
ov      AL , Data 8bit
t       DX , AL
d;
```

PDF created with pdfFactory trial version www.pdffactory.com

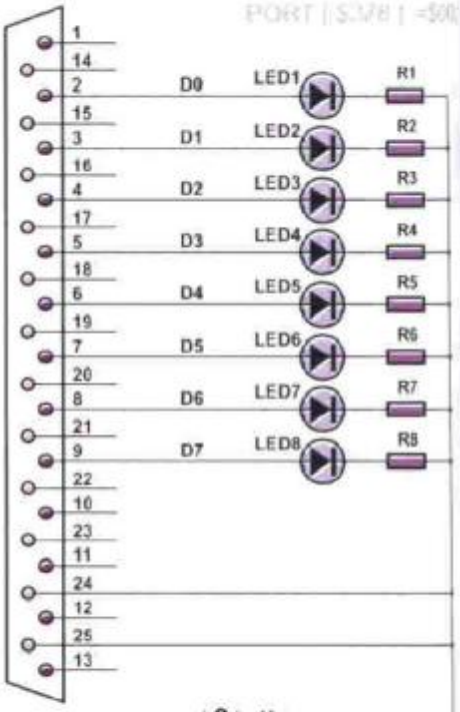
هذه الكتلة البرمجية تكتب كما هي تماماً بدون أي تعديل بعد تعويض قيم العنوان والـ DATA . حيث يتم في السطر الأول نقل قيمة العنوان إلى المسجل DX . وفي السطر الثاني نقل قيمة الـ DATA إلى المسجل AI ، أما التعليمة out فتقوم بإرسال القيمة إلى البوابة .
أما القراءة من البوابة فيتم بالشكل :

```
asm
mov     DX , Address Num
in      AI , DX
mov     Data8bitVar , AI
end;
```

هذه الكتلة البرمجية تكتب كما هي تماماً بعد تعويض قيم العنوان Address Num وتعويض اسم متحول 8bit بدلاً من (Data 8bit var) .

أمثلة تطبيقية بسيطة :

يمكن مراقبة تنفيذ البرنامج المذكور أدناه من خلال وصل ثماني ثنائيات ضوئية عبر مقاومات 820Ω إلى الأرجل من 2 إلى 9 في مخرج الطابعة . كما في الشكل (2) :



شكل (2)

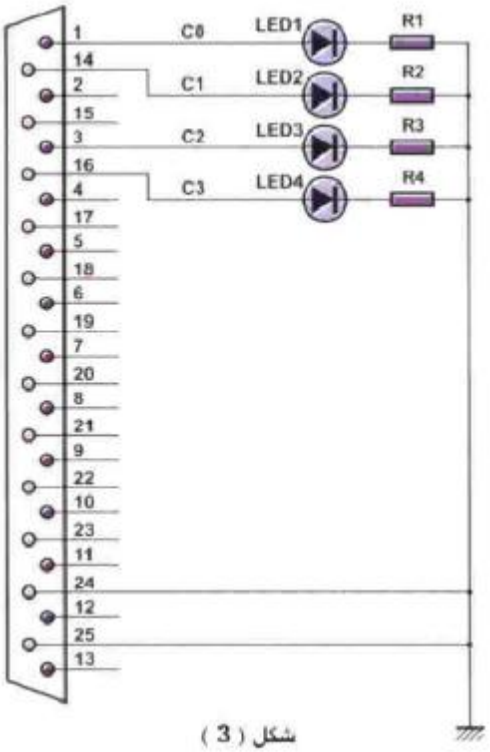
وفيما يلي (Assembly) :

```
asm
mov     DX , $378
mov     AI , $00
out     DX , AI
end;
```

في هذه الحالة سيتم إطفاء جميع الثنائيات في الخرج لأن القيمة المرسلة هي (00) .
وعند إرسال القيمة $FF :

والتي تقابل (11111111) بالنظام الثنائي ستضيء كل الثنائيات الضوئية .
وعند إرسال ($55) والتي تقابل (01010101) ستضيء الثنائيات المقابلة للخانات ذات القيمة (1) .
يمكن تحقيق هذه التجربة من خلال وصل كبل مع مخرج مناسب لمخرج الطابعة إلى Test Board وتثبيت الثنائيات الضوئية والمقاومات عليها .

كما في الشكل (3) . يتم تنفيذ السطر التالي :



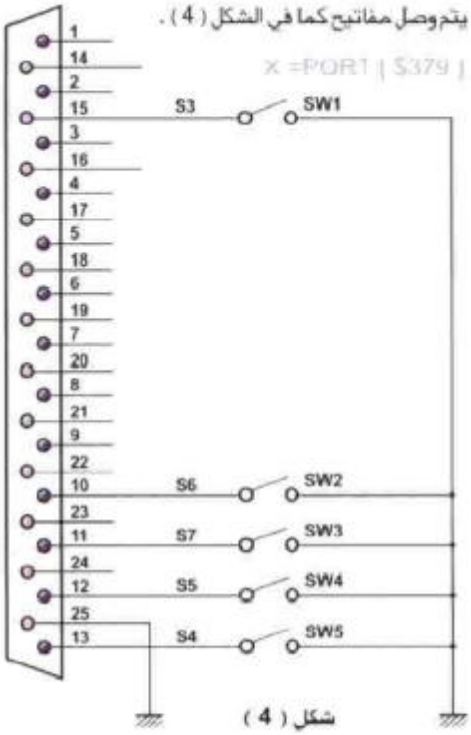
شكل (3)

هنا ستضيء الثنائيات (LED1 , LED2 , LED4) فقط وذلك بسبب أن قيمة هذه الخانات هي عكس مقابلاتها البرمجية .

عند إرسال القيمة :

سيتم إطفاء جميع الثنائيات الضوئية في هذه الحالة .
مع ملاحظة أننا قمنا برمجياً بإرسال قيمة Byte كاملة ($0B) والتي تقابل بالنظام الثنائي (00001011) بينما المربوط منها إلى مخرج الطابعة هو الخانات الأربعة الأولى فقط (1011) .

يتم وصل مفاتيح كما في الشكل (4) .



شكل (4)

سيتم استقبال Byte من بوابة الدخل وتخزينه ضمن المتحول X . لكن يجب الانتباه إلى أن الخانات الخمسة المستقبلة هي الخانات الخمسة العليا وأن الخانة الأخيرة معكوسة حسب الشكل (1) . لذلك ستكون الخانات الثلاثة الأولى غير مفيدة ضمن الـ Byte المقروء . ومن أجل إزاحة الخانات الخمسة العليا لتصبح الخمسة الدنيا نكتب :

X:= x SHR 3 ;

حيث التعليمة SHR هي إزاحة يمينية بمقدار (3) . طبعاً التعامل هنا هو بالنظام الثنائي وهذا يتم بشكل ضمني ولا حاجة لنا للقيام بعمليات تحويل بين الأنظمة من أجل إنجاز عمليات الإزاحة أو التخاطب مع البوابات .

من أجل معرفة حالة خانة معينة من خانات الدخل (0 أو 1) نكتب :

Y:= x And $ 08;

مثلاً من أجل فحص الخانة (S3) من بوابة الدخل حيث أن القيمة (08H) تقابل (00001000) وبالتالي إذا كانت قيمة المتحول (Y) هي صفر فإن الخانة (S3) هي صفر ، وإذا كانت (Y = $08) فهذه الخانة هي (1) .

يتم استخدام الدارة المتكاملة (74244) وهي عبارة عن مجموعتي Buffer ذات أربع خانات ، ولكل مجموعة خط تحكم خاص بها (يمكن استخدام شريحتين كل منها أربع خانات من دارات أخرى) .

PDF created with pdfFactory trial version www.pdffactory.com