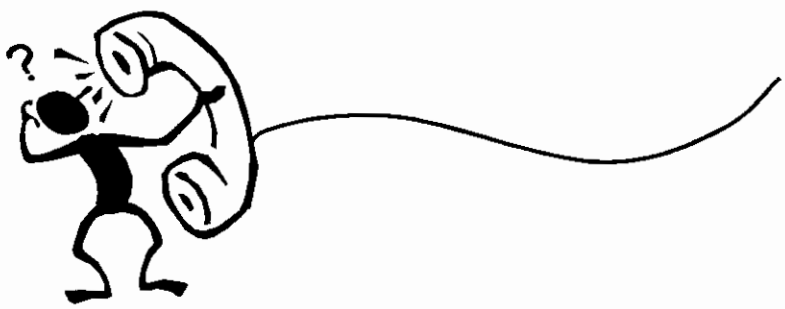
# الفصل التاسع ٩

# المواجهة على التوالى

## Serial Interfacing



## ٩-١ مقدمة

**يمكن** إرسال البيانات بين نقطتين إما فى الصورة الانسيابية analog أو فى الصورة الرقمية digital ، ولكن التراسل فى الصورة الرقمية له عدة مميزات منها ما يلى :

- أن الإشارات الرقمية فى أثناء إرسالها على شبكة الاتصالات تكون أقل عرضة للضوضاء المضافة من وسط الانتقال أو من أجهزة التراسل نفسها .
- يمكن إضافة بعض البتات إلى البيانات الرقمية المرسلة لاستخدامها فى تصحيح الأخطاء التى قد تطرأ على المعلومة ، هذه البتات تسمى بتات الباريتى parity bits أو بتات التصحيح .
- بهذا النظام يمكن استخدام الحاسبات من خلال برمجيات كثيرة لمعالجة الإشارة الرقمية والتى يكون من الصعب استخدامها مع الإشارات فى صورتها الانسيابية .

لقد انتشرت الاتصالات الرقمية الآن انتشارا واسعا وكادت تلغى وجود الاتصالات الانسيابية ، فهناك الآن السنترالات الرقمية ، وشبكات الحاسب ، والفاكس ، والبريد الإلكترونى ، والشبكة الدولية internet والكثير من التطبيقات التى لا يمكن حصرها هنا والتى يمكن استخدام المتحكم من خلالها .

إرسال البيانات الرقمية بين نقطتين يمكن أن يتم على صورتين مختلفتين :

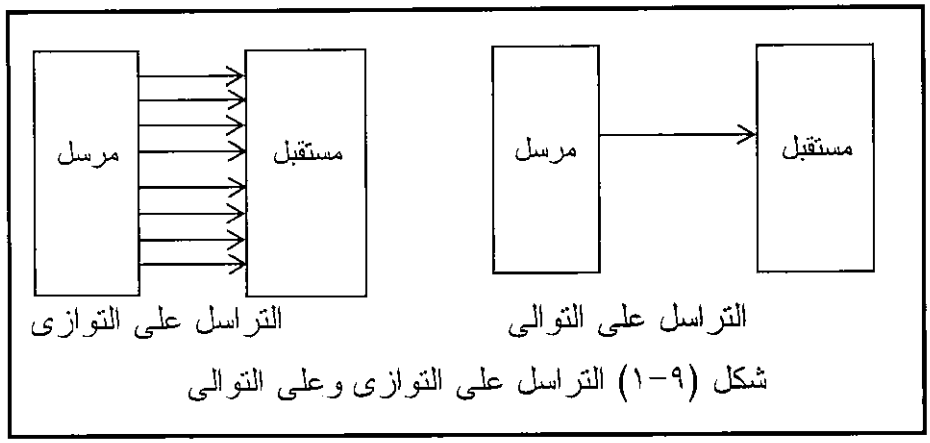
١- الإرسال على التوازى حيث يمكن إرسال بايت كاملة (أو ٢ بايت أو ٤ أو ٨ بايت على حسب عرض مسار البيانات فى الحاسب المستخدم) بين المرسل والمستقبل مرة واحدة على ٨ (أو ١٦ أو ٣٢ أو ٦٤ ) خطوط اتصال تربط بينهما .

٢- النظام الثانى هو إرسال البيانات على التوالى حيث يتم إرسال المعلومة بت بعد بت على خط اتصال واحد يربط المرسل والمستقبل .

شكل (٩-١) يبين رسما تخطيطيا للطريقتين حيث يمكن أن نستنتج من ذلك أن الإرسال على التوازى يكون أكثر تكلفة نتيجة توصيل ٨ خطوط بين المرسل والمستقبل ، ولذلك فإن هذا النوع من التراسل يستخدم فى نظم الاتصالات التى لا تمتد لمسافات طويلة . لذلك فإن هذا النظام يستخدم فى ربط الأجهزة المحيطة بالحاسب مثل الطابعات ، كما يستخدم فى توصيل الكروت المختلفة وتوصيل الشرائح داخل الكروت داخل نظام الحاسب أو على اللوحة الأم للحاسب . أما إذا كانت المسافة بين المرسل والمستقبل ستمتد أكثر من مترين فإنه فى هذه الحالة يتم استخدام نظام التراسل على التوالى لتقليل التكلفة وللمميزات الأخرى الموجودة فى هذا النظام والتى ستتضح فيما بعد .

إن الاتصالات على التوالى تكون فى العادة أصعب من الاتصالات المتوازية لأن البيانات تخرج فى العادة من أى معالج فى صورة متوازية ، لذلك لابد من تحويلها إلى الصورة المتوالية باستخدام شريحة تسمى شريحة الإرسال والاستقبال الغير متوافق ، Universal Asynchronous Receiver Transmitter, UART كما فى شكل

(٩-٢) . ولكن بالرغم من ذلك فإن الاتصالات المتوالية لها مميزات يمكن حصرها فيما يلى :




شكل (٩-١) التراسل على التوازى وعلى التوالى

١- يمكن استخدام تكنولوجيا الاتصالات على التوالى فى عمليات التراسل بين المسافات البعيدة . إن ذلك يرجع أساسا إلى أن المدى بين جهد الواحد المنطقى والصفر المنطقى كبير ، فالواحد المنطقى يمثله -٢٥ فولت بينما الصفر المنطقى فيمثله +٢٥ فولت . أى أن كل منهما يختلف عن الآخر بحوالى ٥٠ فولت مما يجعله أقل تأثرا بالضوضاء المضافة .

٢- عدم الحاجة لأسلاك كثيرة بين المرسل والمستقبل كما رأينا .

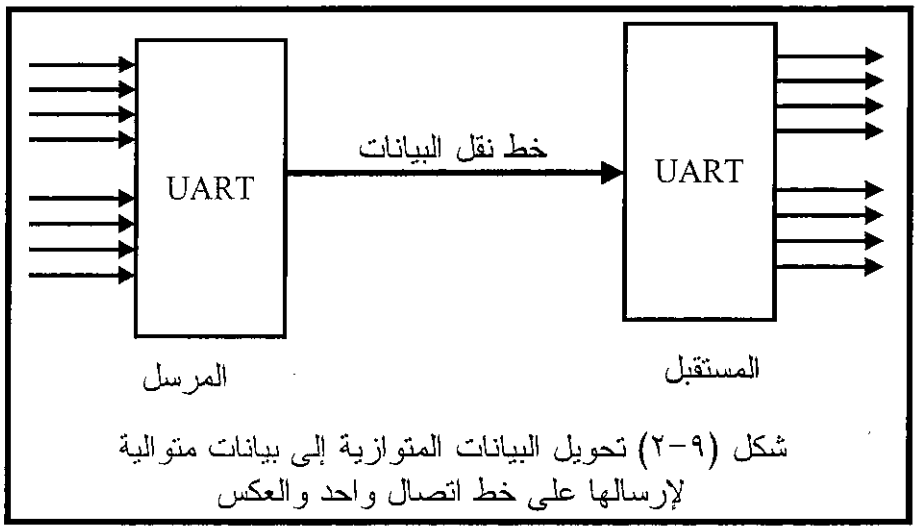
٣- مع انتشار تكنولوجيا التراسل باستخدام الليزر والأشعة فوق الحمراء فإنها تكون أسهل بكثير جدا مع المخارج المتوالية عنها مع المخارج المتوازية .

تنقسم الأجهزة التى تستخدم الاتصالات على التوالى إلى قسمين :

١- أجهزة طرفية Data Terminal Equipment's, DTE وهذه يمثلها جهاز الحاسب الذى نعمل عليه .

٢- أجهزة الاتصالات Data Communication Equipment, DCE وهذه يمثلها المودیم مثلا أو الراسم plotter أو الطابعة أو أى جهاز يرسل ويستقبل على التوالى .

من أول المسارات القياسية التى استخدمت فى الاتصالات على التوالى المسار RS232 الذى تم استخدامه فى عام ١٩٦٢ كأول مسار قياسى يلتزم به كل المشتغلين فى هذا المجال . بعد ذلك ظهرت مسارات قياسية أخرى مثل المسار RS422 والمسار RS449 وكل منها يتميز عن سابقيه بمعدل البيانات التى يتم نقلها فى الثانية ، وطول المسافة التى يتم التراسل عليها . فمثلا المسار RS232 يمكنه أن ينقل البيانات بمعدل ١٩٦٠٠ بت فى الثانية على مسافات أقل من ٢٠ متر . بينما المسار RS422 فيمكن نقل بيانات عليه بمعدل نقل يصل إلى ١٠ ميجابايت فى الثانية لمسافة أقل من ميل باستخدام كابلات محورية .




شكل (٩-٢) تحويل البيانات المتوازية إلى بيانات متوالية
لإرسالها على خط اتصال واحد والعكس

من الشفرات التى تستخدم عالميا فى نظم التراسل على الكابلات RS232 نظام التشفير ASCII (American Standard Code for Information Interchange) الذى يشفر كل الحروف المطبوعة وغير المطبوعة فى ٧ بتات ويستخدم البت الثامنة للباريتى . البت الثامنة تكون واحد أو صفر على حسب نوع الباريتى المستخدمة . فإذا كان نظام الباريتى هو النظام الفردى فإن هذه البت تكون واحد أو صفر بحيث تجعل عدد الوحايد فى البايت كلها يكون فردى . أما إذا كان نظام الباريتى هو النظام الزوجى فإن هذه البت تكون واحد أو صفر بحيث تجعل عدد الوحايد فى البايت كلها عدد زوجى . البرنامج التالى بلغة C يطبع كل الأحرف المطبوعة وغير المطبوعة وبجانب كل حرف شفرته ASCII مكتوبة بالنظام العشرى :

```
#include <stdio.h>
int main(void)
{  int i;
   for(i=0;i<128;i++)
   {  printf("<%3d  %2c>" ,i, i);
      if(!(i%6))
        printf("\n");
   }
   return(0)
}
```

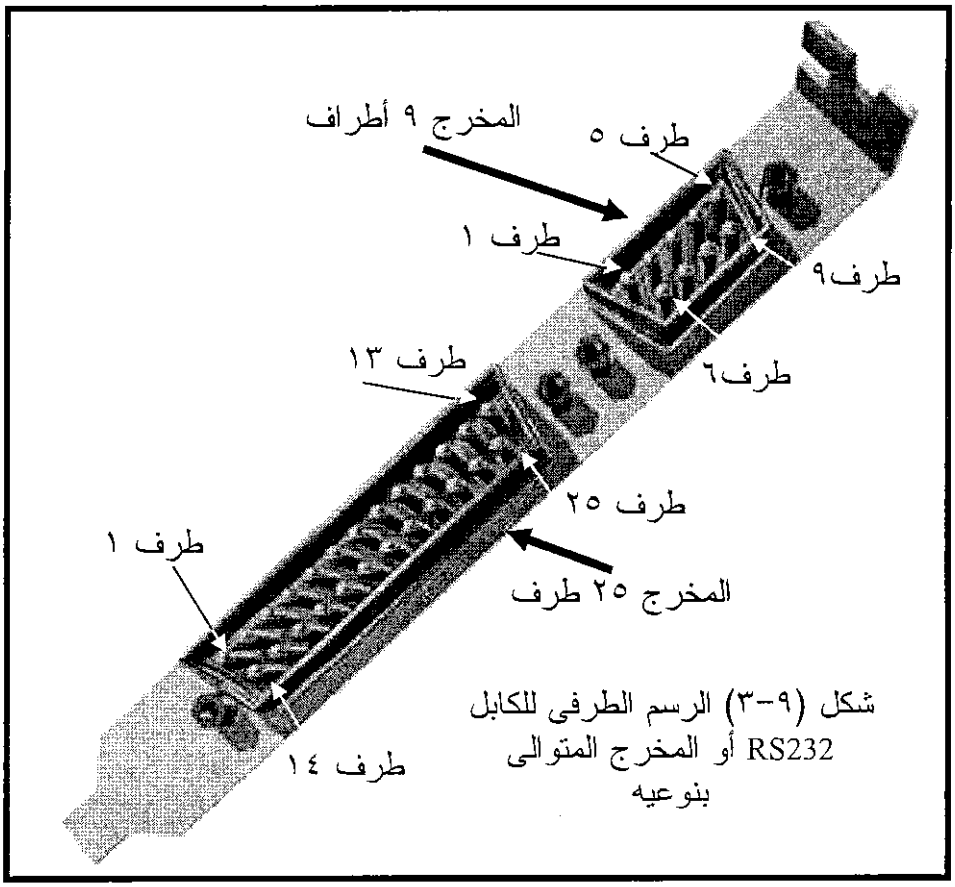
## ٩-٢ مواصفات المسار RS232

فى المواصفات القياسية للمسار RS232 يمثل الواحد المنطقى بأى جهد مقداره بين -٣ فولت و -٢٥ فولت ، كما يمثل الصفر المنطقى بأى جهد مقداره بين +٣ فولت و +٢٥ فولت ، وأى جهد بين -٣ فولت و +٣ فولت يعتبر حالة غير محددة . فى حالة عدم إرسال بيانات يكون المسار فى الحالة المنطقية واحد أى -٢٥ فولت . ولكن الكثير من التطبيقات الواقعية تستخدم +١٢ فولت و -١٢ فولت بدلا من +٢٥ و -٢٥ فولت . من ضمن المواصفات القياسية أيضا ألا يتعدى الجهد على أى طرف مفتوح القيمة ٢٥ فولت ولا يتعدى التيار فى أى طرف مقفول ٥٠٠ ميللى أمبير . فى الأصل كان تصميم المخارج المتوالية بغرض التعامل مع الموديم ، وسنرى فيما يلى أن وظيفة كل طرف من أطراف هذه المخارج يعكس هذه الوظيفة .

يحتوى أى جهاز حاسب على مخرجين على الأقل للاتصالات التتابعية إحداهما يسمى COM1 والاخر يسمى COM2 ، وفى الغالب توصل الفأرة على واحد منهما ويكون هذا المخرج فى الغالب من النوع ذو ٩ أطراف . توجد المخارج التتابعية على واحدة من صورتين ، الأولى هى المخرج ذو ٢٥ طرفا ، والأخرى هى المخرج ذو ٩ أطراف . شكل (٩-٣) يبين الرسم الطرفى لكل من الإصدارين . وظائف هذه الأطراف تكون إما بغرض التحكم فى البيانات ، أو تكون بيانات مرسلة أو مستقبلة ، أو تكون أرضى ، فهناك الكثير من هذه الأطراف موصلة على الأرضى . وظائف هذه الأطراف هى كما يلى :

- Frame Ground, FG الأرضى ، وهو الطرف رقم ١ ، ويتصل به الغلاف الخارجى (الأرضى) للكابل .
- Transmit Data, TD إرسال البيانات ، حيث تخرج البيانات المرسلة من الحاسب للموديم على هذا الطرف .
- Receive Data, RD استقبال البيانات ، حيث يتم استقبال البيانات المرسلة من الموديم على هذا الطرف .
- Ready (Request) To Send, RTS جاهز للإرسال ، حيث يتم تنشيط هذا الطرف بواسطة المرسل DTE أو الحاسب عند الاستعداد لإرسال بيانات ، أى أن هذه إشارة تحكم خارجة من النهاية الطرفية (الحاسب) تسأل الموديم إذا كان جاهز لأن يرسل له بيانات .
- Clear To Send, CTS جاهز للإرسال ، عندما يكون المستقبل DCE أو الموديم جاهزا لاستقبال البيانات يرد على المرسل (الحاسب) بتنشيط هذا الخط ، ومعناه أنه قام بتجهيز المسجلات (تصفيرها) لاستقبال البيانات الجديدة ، أى أن هذه إشارة تحكم داخلة للنهاية الطرفية (الحاسب) .
- Data Send(Set) Ready, DSR ينشط بواسطة الموديم DCE حينما يكون جاهز لإرسال بيانات إلى الحاسب .
- Signal Ground, SG الأرضى الذى تنسب إليه إشارة جميع الأطراف .

• Data Terminal Ready, DTR ينشط بواسطة الحاسب DTE حينما يكون جاهزا لاستقبال بيانات من المودیم . أى أن الخطين RTS و CTS يستخدمان للتحكم حينما يكون المرسل جهاز حاسب DTE والمستقبل موديم DCE . بينما الخطين DSR و DTR فلهما نفس الوظيفة حينما يكون المرسل موديم DCE والمستقبل هو الحاسب DTE .




شكل (٩-٣) الرسم الطرفى للكابل RS232 أو المخرج المتوالى بنوعيه

• Carrier Detect, CD ينشط عند استقبال موجة حاملة من الجانب الآخر ، وتعنى أن المودیم موصلا الآن على تليفون يعمل .

• Ring Indicator, RI ينشط لبيان وجود جرس تليفون من الطرف الآخر .

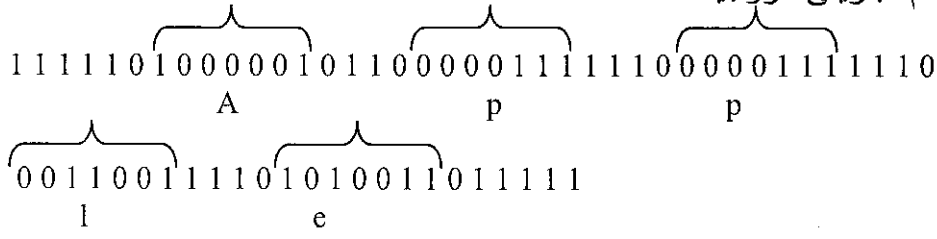
كل هذه الإشارات تكون فعالة عندما تكون صفر (١٢ فولت) ، وتكون غير فعالة عندما تكون واحد (-١٢ فولت) . جدول ٩-١ يبين كل طرف من هذه الأطراف على إصدارى المخرج RS232 . لاحظ أن كل الإشارات السابقة كانت تعتبر دائما أن التعامل هو بين حاسب ومودیم وهذا هو الأساس لأن الاتصالات على التوالى أول ما استخدمت كانت للاتصال بين الحاسب ومودیم .

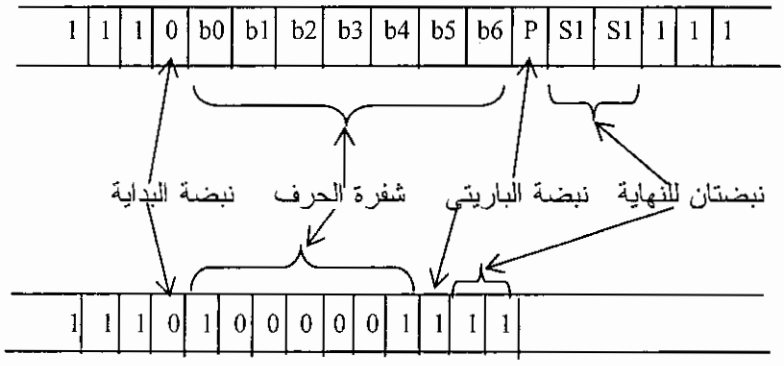
| المخرج ٢٥ طرف | المخرج ٩ أطراف | الاسم المختصر | الاسم بالكامل |
|---|---|---|---|
| 2 | 3 | TD | Transmit Data, إرسال البيانات |
| 3 | 2 | RD | Receive Data, استقبال البيانات |
| 4 | 7 | RTS | Request To Send, طلب إرسال |
| 5 | 8 | CTS | Clear To Send, جاهز فى انتظار الإرسال |
| 6 | 6 | DSR | Data Set Ready, بيانات جاهزة عند الموديم |
| 7 | 5 | SG | Signal Ground, أرضى الإشارة |
| 8 | 1 | CD | Carrier Detect, اكتشاف موجة حاملة |
| 20 | 4 | DTR | Data Terminal Ready, الجهاز الطرفى جاهز |
| 22 | 9 | RI | Ring Indicator, وجود جرس تليفون |

جدول ٩-١ بيان بأطراف وأسماء ووظيفة المخارج المتوازية

المسار RS232 يستخدم طريقة التراسل غير التوافقى Asynchronous Communication ، بمعنى أنه لا ضرورة لوجود نبضات التوافق (الساعة) بين المرسل والمستقبل . يتم تشكيل الأحرف على حسب نظام موحد يتم الاتفاق عليه بين المرسل والمستقبل وهذا النظام موضح فى شكل (٩-٤) . نلاحظ فى هذا الشكل أن الإرسال يبدأ بنبضة بداية Start bit وهى فى الغالب صفر منطقى (0) وبعد ذلك يبدأ إرسال الحرف المراد إرساله فى ٧ بتات ، ثم توضع البت (أو البتات) الخاصة بالباريتى وذلك أيضا على حسب نوع الباريتى الذى يتم اختياره ، ثم فى النهاية يتم وضع ٢ بت للدلالة على نهاية الإرسال وتكون فى الغالب واحد (1) . شكل (٩-٤) يبين تفاصيل إرسال الحرف A=1000001 حيث نلاحظ وجود صفر فى البداية للدلالة على بداية الحرف ، ثم أضيف واحد فى نهاية الحرف ليجعل عدد الوحايد فردى حسب نظام الباريتى الفردية ، ثم وضع 11 للدلالة على نهاية الحرف .

كمثال على الإرسال على المسار RS232 تتبع الرسالة التالية التى تمثل كلمة Apple فى نظام الباريتى الزوجية :






شكل (٩-٤) طريقة الإرسال على المسار RS232

كما رأينا فإن البيانات على المسار RS232 تبدأ عند الإرسال بنبضة البداية Start bit وتكون صفر منطقى ، بعدها تأتى بتات الحرف المراد إرساله بداية بالبت ذات القيمة الصغرى b0 ، وبعدها b1 وهكذا حتى آخر بت فى الحرف ، ثم يأتى بعد ذلك البت الممثلة للباريتى ، وبعدها البتات التى تحدد نهاية الحرف . أى أن العدد الممثل لأى حرف سيكون ١٠ بت (١+٧+١+١) فى حالة استخدام بت واحدة بدلا من ٢ للدلالة على نهاية الحرف . فإذا كان معدل إرسال الأحرف هو ٩٦٠ حرفا فى الثانية ، فإن معدل تدفق البتات على المسار سيكون ٩٦٠٠ بت فى الثانية ، وهذا ما يطلق عليه معدل بود أو Baud Rate . أى أن معدل بود هو معدل تدفق البتات الممثلة للأحرف المرسلة على المسار التتابعى فى الثانية . يمكن تحديد زمن البت الواحدة بحساب معدل تدفق البتات . فإذا كان معدل التدفق هو ٩٦٠٠ بت فى الثانية مثلا ، فإن زمن البت الواحدة سيكون ( 1/9600=104) ميكروثانية .

## ٩-٣ المصافحة Handshaking

عندما يقوم المرسل بإرسال حرفا على المسار RS232 ، فإن المستقبل عليه أن يقوم باستقبال هذا الحرف وتخزينه فى مخزن مؤقت buffer قبل إن يقوم المرسل بإرسال حرفا آخر . إذا لم يحدث ذلك وتأخر المستقبل فى عملية التخزين فإن المرسل يمكن أن يرسل حرفا آخر فيكتب فوق حرف سابق عند المستقبل وبالتالى فإن هذا الحرف السابق سيفقد تماما . هذه المشكلة تحدث نتيجة عدم وجود مصافحة أو عدم التفاهم بين المرسل والمستقبل . يمكن التغلب على ذلك إذا أمكن إيجاد نوع من المصافحة بين المرسل والمستقبل ، فالمرسل مثلا يمكن أن يخبر المستقبل أنه سيرسل حرف ، فيرد عليه المرسل أنه جاهز لاستقبال هذا الحرف فيقوم المرسل بعملية الإرسال ،

أما إذا كان المستقبل غير مستعد لاستقبال هذا الحرف فإن المرسل ينتظر حتى يستعد المستقبل لعملية الاستقبال . تتم عملية المصافحة هذه على خطوط التحكم الأخرى على المسار وهذا ما سنراه فى هذا الجزء . خطوط المصافحة على المسار RS232 كما رأيناها مسبقا هى :

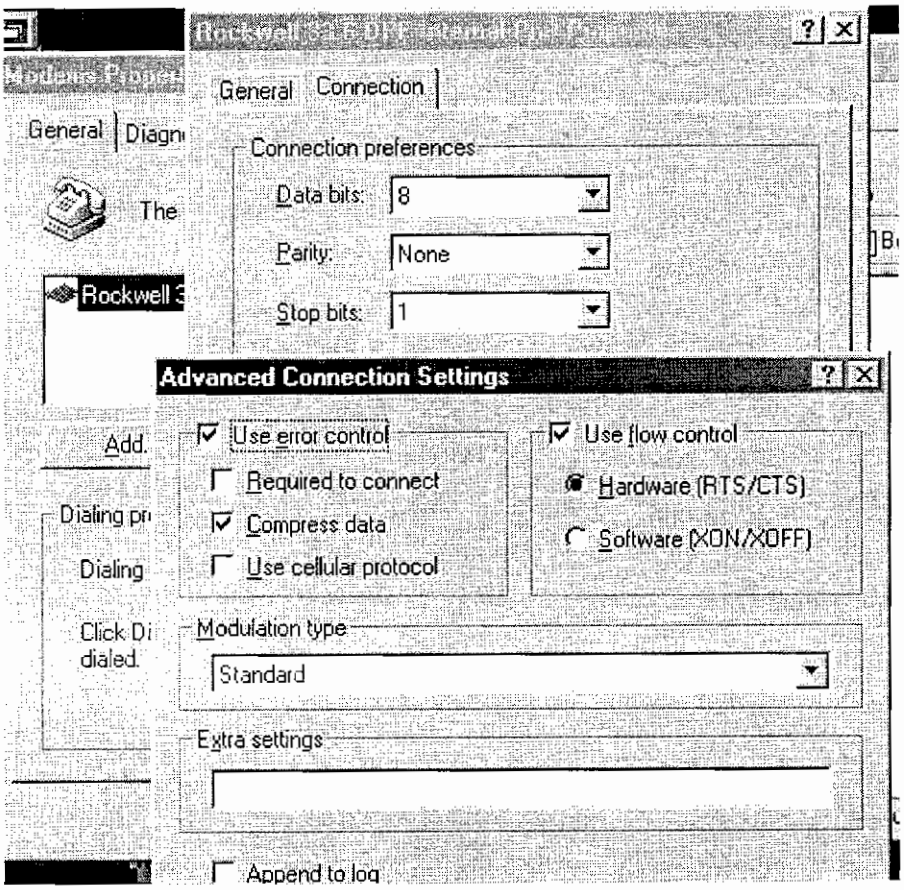
- CTS Clear To Send
- RTS Ready To Send
- DTR Data Terminal Ready
- DSR Data Send Ready

أى جهاز حاسب يحتوى على أكثر من مخرج للاتصالات على التوالى وفى العادة يطلق على هذه المخارج COM1 و COM2 و COM اختصار لكلمة communication . فى العادة يتم توصيل الفأرة على أحد هذين المخرجين والمخرج الآخر يتاح للاستخدامات الحرة عن طريق المستخدم أو بأحد كروت الاتصالات مثل كارت الفاكس أو كارت الشبكة . يمكن ضبط خواص هذه المخارج من حيث نوع الباريتى ومعدل بود وعدد بتات نهاية الحرف وغيرها من الخواص بالدخول فى برنامج النوافذ بالتتابعات التالية وكما فى شكل (٩-٥) :

Start→ Control Panel→ Modems→ General→ Advanced connection

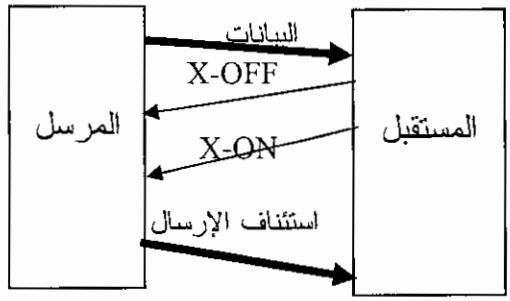
هناك نوعان من المصافحة ، المصافحة البرمجية software handshaking حيث فى هذه الحالة يتم استخدام الحرفان X-ON الذى شفرته الأسكى هى 17 والحرف X-OFF الذى شفرته الأسكى 19 فى عملية المصافحة . عندما يستقبل المرسل الحرف X-OFF من المستقبل فإنه يوقف عملية الإرسال إلى أن يقوم المستقبل بإرسال الحرف X-ON مرة ثانية للمرسل حيث نتيجة لذلك يستأنف المرسل عملية الإرسال . لاحظ أن الحرفان X-ON و X-OFF يستخدمهما المستقبل للسماح للمرسل بعملية الإرسال أو تعطيلها . شكل (٩-٦أ) يبين كيفية استخدام هذين الحرفين من قبل المستقبل للتحكم فى عملية الإرسال .

النوع الآخر من المصافحة هو المصافحة باستخدام أطراف التحكم على المسار RS232 . كل خطوط التحكم التى رأيناها على هذا المسار تكون فعالة عندما تكون فى حالة الواحد المنطقى high . شكل (٩-٦ب) يوضح ذلك . عندما يريد أحد الأطراف الاتصال بالطرف الآخر فإنه يقوم بتنشيط الطرف RTS أو جاهز للإرسال الخارج منه . هذا الطرف يكون متصلا بالطرف CTS أو جاهز للإرسال عند المستقبل (الموديم) . إذا كان المستقبل جاهزا لاستقبال بيانات ، أى أن المخزن المؤقت فيه فارغ فإنه يقوم بتنشيط الطرف RTS الخاص به والمتصل بالطرف CTS عند المرسل . لذلك فإن المرسل عليه مراقبة الطرف CTS عنده حتى إذا أصبح نشطا (1) فإن ذلك يعنى أن المستقبل مستعد لاستقبال حرفا جديدا ، فيقوم بعملية الإرسال على الطرف TD والمتصل بالطرف RD عند المستقبل .

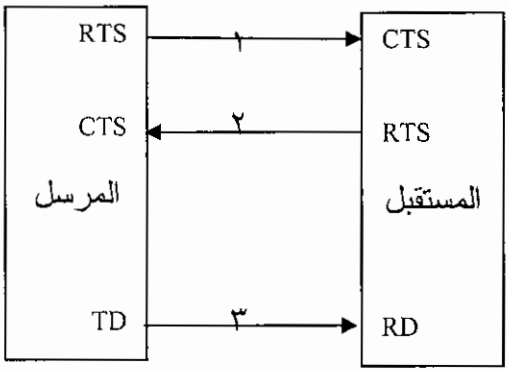



شكل (٩-٥) ضبط خواص المسار RS232 من خلال برنامج النوافذ

يمكن استخدام طريقة مبسطة لتوصيل جهازين من خلال مخرج التوالى بجعل كل جهاز يعتقد أن الجهاز الواصل عليه هو موديم وليس جهاز حاسب ، لذلك فإن هذه الطريقة تسمى طريقة الموديم الافتراضى . سنفترض فى هذه الطريقة أن المستقبل سيقوم بنقل الحرف الذى تم إرساله من مخزن الاستقبال قبل أن يقوم المرسل بإرسال الحرف التالى وإلا فإن هذا الحرف سيفقد حيث سيتم كتابة الحرف الجديد عليه . شكل (٩-٧) يبين طريقة التوصيل بين جهازين (مرسل ومستقبل) بدون استخدام المصافحة . لاحظ توصيل الطرف TD ، إرسال البيانات ، فى كل من الجهازين على الطرف RD قراءة البيانات ، فى الجهاز الآخر . لاحظ أيضا توصيل الطرف RTS الاستعداد للإرسال على الطرف CTS جاهز (clear) للإرسال فى كل جهاز وليس مع الجهاز الآخر ، كذلك تم توصيل الطرف DTR مع الطرف DSR فى نفس الجهاز والطرف CD .




أ- المصافحة البرمجية




ب- المصافحة باستخدام خطوط التحكم

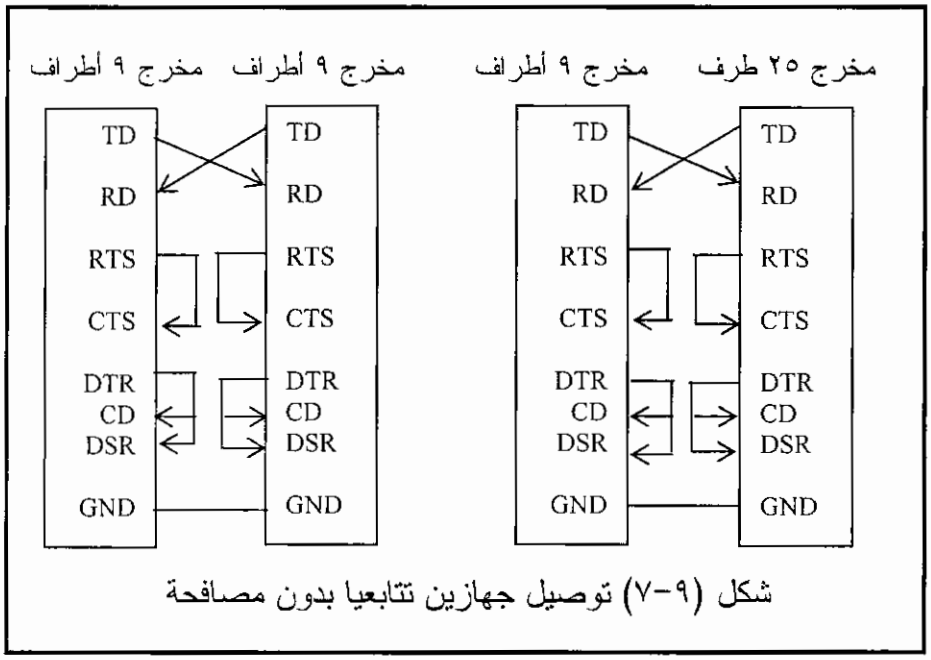
شكل (٩-٦) التراسل التتابعى باستخدام المصافحة

## ٩-٤ برمجة مخرج التوالى

أى مخرج توالى يكون له فى العادة ٣ مسجلات أساسية على الأقل وهى :

- مسجل مخزن الإرسال أو الاستقبال ويحمل العنوان الأساسى base address للمخرج . لاحظ أن كل من مخزن الإرسال والاستقبال لهما نفس العنوان . فى الغالب تكون العناوين المخصصة لمخارج التوالى كما فى جدول ٩-٢ .
- مسجل التحكم Line Control Register, LCR ويكون عنوانها هو العنوان الأساسى +٣ .
- مسجل الحالة Line Status Register, LSR ويكون عنوانه هو العنوان الأساسى +٥ .

مسجل الحالة LSR عبارة عن مسجل يمكن قراءته فقط ولا يمكن الكتابة فيه . بتات هذا المسجل تعكس حالة المخزن المؤقت عند المرسل والمستقبل . شكل (٩-٨) يبين الوظيفة التى تعكسها كل بت من بتات هذا المسجل .



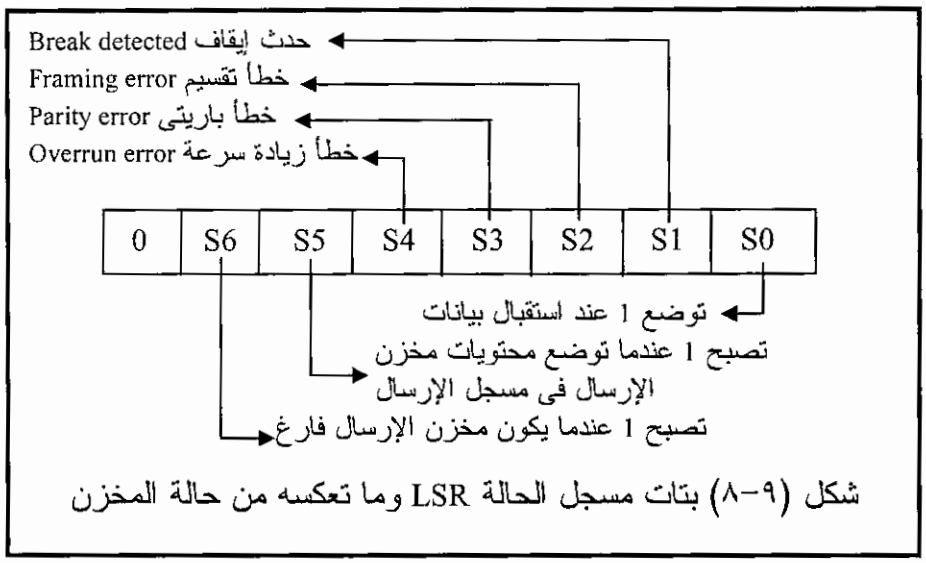

شكل (٩-٧) توصيل جهازين تتابعيا بدون مصافحة

مسجل التحكم LCR يمكن عن طريقه تحديد معاملات عملية الاتصالات مثل عدد البتات التى ستمثل كل حرف ، والباريتى ، وعدد بتات النهاية . هذا المسجل يمكن الكتابة فيه والقراءة منه . شكل (٩-٩) يبين بتات مسجل التحكم ووظيفة كل بت فى تحديد معاملات الاتصالات .

جدول ٩-٢

| اسم المخرج | عنوان البداية | رقم المقاطعةIRQ |
|---|---|---|
| COM1 | 3F8 | 4 |
| COM2 | 2F8 | 3 |
| COM3 | 3E8 | 4 |
| COM4 | 2E8 | 3 |

من الأمور المهمة التى لابد من الحذر منها هى أن يتم كتابة حرف فى مخزن المرسل transmitter buffer قبل تفريغ هذا المخزن من آخر حرف تم إرساله من قبل وإلا فإنه فى هذه الحالة فإن الحرف الجديد سيكتب فوق الحرف القديم ويضيع الحرف القديم . لتجنب ذلك نختبر البت S6 فى مسجل الحالة قبل عملية الإرسال بحيث إذا كانت S6=1 فإن ذلك يعنى وجود حرف فى المخزن ، وإذا كانت S6=0 فإن ذلك يعنى أن المخزن فاضى ويمكن تحميل مخزن الإرسال . هذه الأوامر تختبر هذه البت قبل عملية الإرسال :




شكل (٩-٨) بتات مسجل الحالة LSR وما تعكسه من حالة المخزن

```
do { status=inportb(LSR);
     status=status&0x40;
    }while (status= =0x40);
/*Send data*/
```

بنفس الطريقة عند المستقبل فإن البت S0=1 عندما يكون هناك حرف فى مخزن المستقبل ، لذلك لابد من اختبار هذه البت أولا لنرى إذا كان مخزن المستقبل فارغ أم لا . الأوامر التالية تبين ذلك :

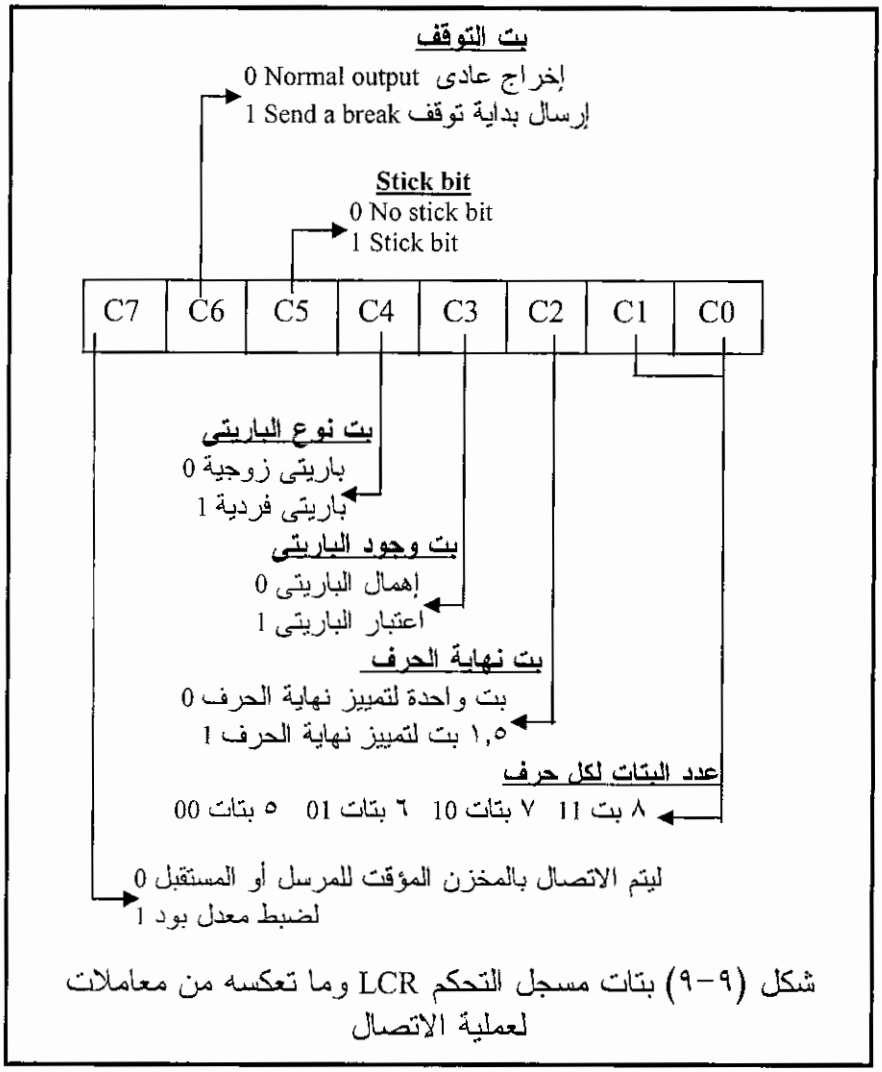
```
do { status=inportb(LSR);
     status=status&0x01;
    }while (status!=0x01);
/* Read data*/
```

لكى تتمكن من التعامل مع مخزن المرسل أو المستقبل لابد أن تكون البت C7=0 فى مسجل التحكم . عندما تكون C7=1 فإنه فى هذه الحالة يمكن ضبط معدل بود عن طريق تحميل رقم من ١٦ بت فى عنوان مخزن الإرسال أو الاستقبال والتالى له . هذا الرقم يعتمد على قيمة تردد شريحة الإرسال والاستقبال التتابعى 8250 . هذا الرقم N يمكن حسابه من المعادلة التالية :

Baud rate = Clock frequency/ (16*N)

جدول ٩-٣ يبين قيمة N المقابلة لبعض قيم معدل بود الشهيرة عندما كانت قيمة تردد الشريحة هى ١,٨٤٣٢ ميجاهرتز . تبعا لهذا الجدول فإنه لضبط معدل بود ٩٦٠٠ فإننا يجب أن نحمل العنوان 03F8H بالقيمة 0CH (البايت ذات القيمة الصغرى من القيمة 000CH كما فى الجدول) ، ثم نحمل العنوان 03F9H بالقيمة 00H . لاحظ أن العنوان 03F8H هو عنوان المخرج COM1 وهو أيضا عنوان

مخزن الاستقبال أو الإرسال على حسب تحديد هذا المخرج . فى أثناء عملية ضبط معدل بود لابد أن تكون البت C7 تساوى واحد .



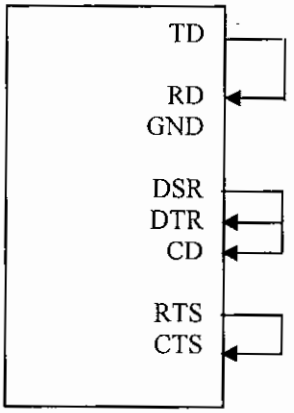

شكل (٩-٩) بتات مسجل التحكم LCR وما تعكسه من معاملات لعملية الاتصال

شكل (٩-١٠) يبين كيفية توصيل المخرج المتوالى مع نفسه لاختبار عملية الاتصال حيث سيوصل طرف الإرسال بطرف الاستقبال فى نفس المخرج ، بذلك فإن أى حرف يتم إرساله من خلال لوحة المفاتيح ، سيتم استقباله من نفس المخرج . تستمر عملية الإرسال والاستقبال هذه إلى أن نضرب الحرفين Ctrl+D والتى تمثل نهاية الإرسال EOT حيث يتم الخروج من البرنامج . شفرة الأسكى للحرف Ctrl+D هى 04H . البرنامج التالى سينفذ عملية الاختبار هذه .

```
/*This program transmits a character from COM1 and receives it through the
                                                          same port. */
#define COM1BASE  0x3F8H
#define TXDATA  COM1BASE
#define LCR  (COM1BASE+3)  /*  0x3FB  */
#define LSR  (COM1BASE+5)  /*  0x3FD  */
#include <conio.h>
#include <dos.h>
#include<stdio.h>
/*declaration of function prototypes */
void  setup_serial (void);
void send_character (int );
int  get_character(void);
```

جدول ٩-٣

| Baud rate | N |
|---|---|
| 110 | 0417H |
| 300 | 0180H |
| 600 | 00C0H |
| 1200 | 0060H |
| 1800 | 0040H |
| 2400 | 0030H |
| 4800 | 0018H |
| 9600 | 000CH |
| 19200 | 0006H |




شكل (٩-١٠) اختبار عملية الاتصال على التوالى

```
int main(void)
{  int inchar, outchar;
   setup_serial ();
   do { puts("Enter character to be transmitted , Ctrl+D to end");
        outchar=getch();
        send_character(outchar);
        inchar=get_character();
        printf("Character received was %c\n", inchar);
      } while (outchar!=4);
    return(0);
 }

void  setup_serial (void)
{  outportb(LCR, 0x80H);
   /* this will set bit 7 to 1 to to set baud rate */
```

```
  outportb(TXDATA, 0xCH);
  outportb(TXDATA+1, 0x00H);
  outportb(LCR, 0x0AH);
  /* access TD/TR buffer, normal output, no stick, even parity, 1 parity bit, 1
stop bit, 7 data bits */
 }

void send_character(int ch)
{  char status;
   do { status=inportb(LSR) &0x40H;
        } while (status !=0x40H);
    outportb(TXDATA, (char) ch);
}

int get_character(void)
{  int  status;
   do { status=inportb(LSR)&0x01H;
      } while (status!=0x01H);

   return ( (int) inportb(TXDATA));
}
```

البرنامج السابق اختبر عملية الاتصالات على نفس الجهاز ، أى أن نفس الجهاز (الحاسب) كان هو نفسه المرسل والمستقبل فى نفس الوقت . سنرى الآن كيفية الاتصالات بين جهازى حاسب ، أحدهما سيكون المرسل والآخر سيكون المستقبل وسيتم ربط الجهازين كما فى شكل (٩-٧) . هذه التوصيلة تسمى كما ذكرنا توصيلة المودم الافتراضى Null Modem . تعتبر هذه التوصيلة أرخص وأسهل وسيلة لتوصيل جهازى حاسب حيث أنها تستخدم ٣ أسلاك فقط . البرنامج التالى يتم تنفيذه على جهاز المرسل ليقوم بعملية إرسال البيانات :

```
/* this program is at the transmitter */
#define TXDATA   0x3F8H
#define LCR  0x3FB
#define LSR   0x3FD
#include <conio.h>
#include <dos.h>
#include<stdio.h>
/*declaration of function prototypes */
void  setup_serial (void);
void send_character (int);
int main(void)
{ int ch;
  puts("Transmitter program, please enter the text to be send, Ctrl+D to
```

```
  end");
  setup_serial();
  do { ch=getche();
       send_character(ch);
     }while (ch!=04H);
     return(0);
}

void setup_serial(void)
{  outportb(LCR, 0x80H); /* C7=1 to set register address bit */
   outportb(TXDATA, 0x0CH);
   outportb(TXDATA+1, 0x00H);
   outportb(LCR,0x0AH); /* lock at previous program */
}

void  send_character (int ch)
{  char status;
   do { status = inportb(LSR) & 0x04H;
      } while (status!=0x04H);
   outportb(TXDATA, (char) ch);
}
```

وهذا هو البرنامج الذى سيكتب عند المستقبل :

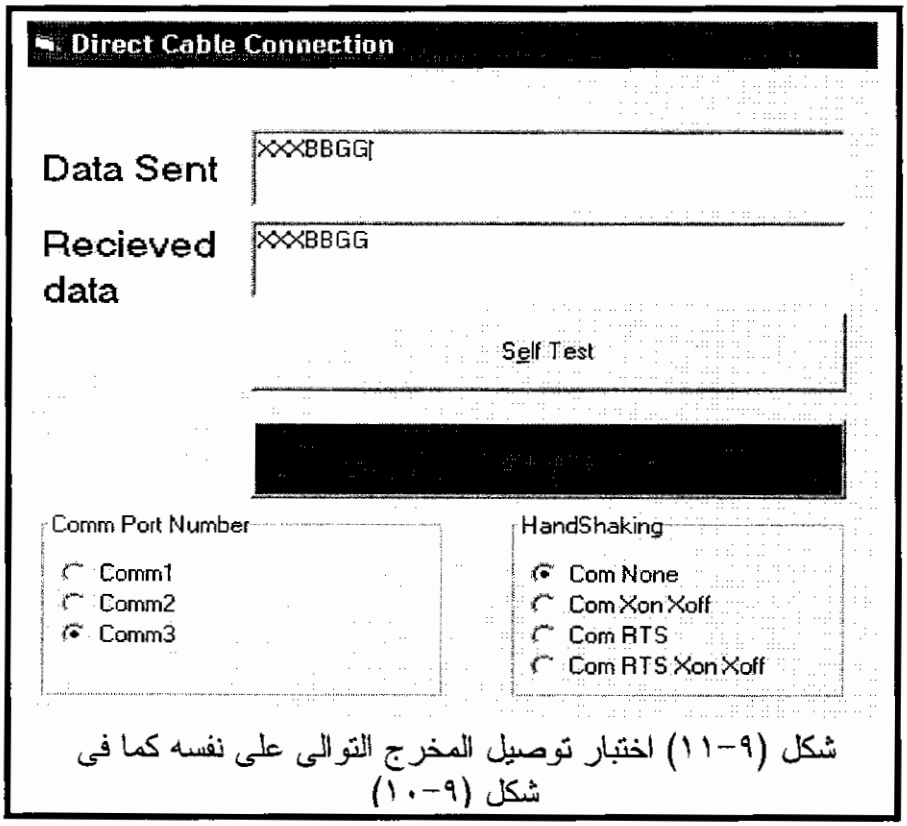
```
/* Receive program */
#define TXDATA   0x3F8H
#define LCR  0x3FB
#define LSR   0x3FD
#include <conio.h>
#include <dos.h>
#include<stdio.h>
/*declaration of function prototypes */
void  setup_serial (void);
int  get_character (void);

int main(void)
{ int inchar;
  setup_serial();
  do {  inchar=get_character();
        putchar(inchar);
     } while (inchar!=0x04H);
   return(0)
}
```

```
void setup_serial(void)
{  outportb(LCR, 0x80H); /* C7=1 to set register address bit */
   outportb(TXDATA, 0x0CH);
   outportb(TXDATA+1, 0x00H);
   outportb(LCR,0x0AH); /* lock at previous program */
}

int get_character (void)
{  int status;
   do { status = inportb(LSR) & 0x01H;
        } while (status!=0x01H);
   return ( (int) inportb(TXDATA));
}
```

إننا هنا جعلنا كل جهاز حاسب يعتقد أنه يتصل مع موديم على الطرف الآخر . لذلك فإن كل طرف إرسال TD تم توصيله بطرف الاستقبال RD فى الناحية الأخرى . ولا ننسى توصيل طرف الأرضى SG بين الجهازين . فى كل جهاز تم توصيل الطرف DTR فى كل جهاز على الأطراف DSR و CD فى نفس الجهاز بحيث عندما ينشط الطرف DTR فإنه ينشط الخط DSR ليفهم الجهاز أن هناك بيانات جاهزة للإرسال وينشط أيضا الطرف CD ليفهم أن هناك موجة حاملة تم الحصول عليها من الجهاز الآخر . بما أن الجهازين يعملان بنفس السرعة فإن الطرف RTS فى كل من الجهازين تم توصيله بالطرف CTS وبذلك يمكن الاستغناء تماما عن عملية التحكم فى تدفق البيانات بين الجهازين . شكل (٩-١١) يبين نفس البرنامج السابق باستخدام لغة الباسيك المرئى visual basic ، والبرنامج المستخدم فى ذلك يمكن كتابته كما يلى :

```
Dim FLAGE As Integer
Private Sub Command1_Click()
If FLAGE = 1 Then
FLAGE = 0
  Command1.BackColor = &H8000000F
  If MSComm1.PortOpen = True Then
     MSComm1.PortOpen = False
  End If
Else
  FLAGE = 1
  Command1.BackColor = &HFF&
  If MSComm1.PortOpen = False Then
     MSComm1.PortOpen = True
  End If
End If
```

Do While FLAGE = 1
Dim Instring As String




شكل (٩-١١) اختبار توصيل المخرج التوالى على نفسه كما فى شكل (٩-١٠)

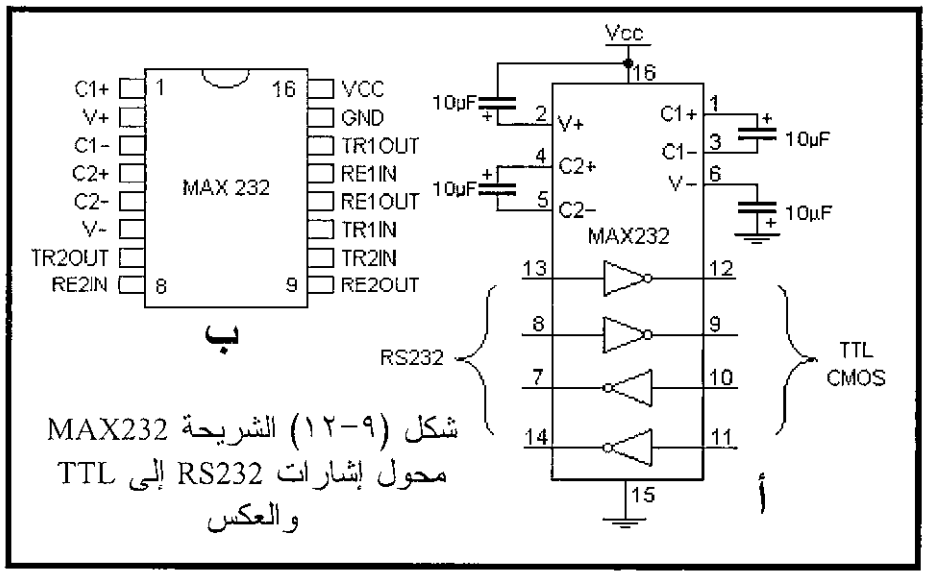
```
'  Retrieve all available data.
   MSComm1.InputLen = 0 ' gives max number of recieved data
'Check for data.
   If MSComm1.InBufferCount Then
'Read data.
      Instring = MSComm1.Input
      Text2.Text = Text2.Text + Instring
   End If
   DoEvents
Loop
End Sub

Private Sub Form_Load()
```

```
FLAGE = 0
End Sub

Private Sub Option1_Click(Index As Integer()
MSComm1.CommPort = Option1(Index).Index + 1
End Sub

Private Sub Option2_Click(Index As Integer()
MSComm1.Handshaking = Option2(Index).Index
End Sub

Private Sub selftest_Click()
Dim buffer As Variant
 '    Set and open port
If MSComm1.PortOpen = False Then
   MSComm1.PortOpen = True
End If
   buffer = Text1.Text
   MSComm1.Output = buffer

Dim Instring As String
 'Retrieve all available data.
MSComm1.InputLen = 0 ' gives max number of recieved data
 'Check for data.
If MSComm1.InBufferCount Then
 '    Read data.
   Instring = MSComm1.Input
End If
Text2.Text = Instring
End Sub
Private Sub Text1_KeyPress(KeyAscii As Integer()
Dim ch As Variant
ch = Chr(KeyAscii()
 '    Set and open port
If MSComm1.PortOpen = False Then
   MSComm1.PortOpen = True
End If
   MSComm1.Output = ch
End Sub
```

عند تنفيذ هذا البرنامج واختيار رقم المخرج المناسب والنقر على زرار Wait Mode ، ثم وضع دليل الكتابة Cursor فى صندوق المرسل فإن أى حرف تتم كتابته فى صندوق كتابة المرسل يظهر فى الحال فى صندوق كتابة المستقبل . يمكن وضع نفس البرنامج فى جهاز آخر كمستقبل وتوصيل الجهازين مع بعضهما كما أوضحنا سابقا وفى هذه الحالة فإن أى كتابة تكتب فى صندوق المرسل فى الجهاز الأول ستظهر فى صندوق المستقبل فى الجهاز الثانى .

## ٩-٥ تحويل الإشارات من نظام RS232 إلى نظام TTL والعكس

معظم الدوائر المنطقية التى نتعامل معها تستخدم النظام المنطقى TTL الشائع الذى يمثل الواحد المنطقى فيه بخمسة فولت والصفر المنطقى بصفر فولت . ولقد رأينا هنا أن النظام RS232 يتعامل مع إشارات يمثل الواحد المنطقى فيها بالقيمة -١٢ فولت ، والصفر المنطقى +١٢ فولت فى أغلب الأحيان . لذلك فإنه فى كثير من التطبيقات التى سنتعامل فيها مع إشارات RS232 سنحتاج إلى تحويلها إلى نظام TTL أو العكس . يوجد فى السوق أكثر من شريحة تقوم بعملية التحويل وسنقدم هنا الشريحة MAX232 التى يمكن استخدامها لهذا الغرض .




شكل (٩-١٢) الشريحة MAX232 محول إشارات RS232 إلى TTL والعكس

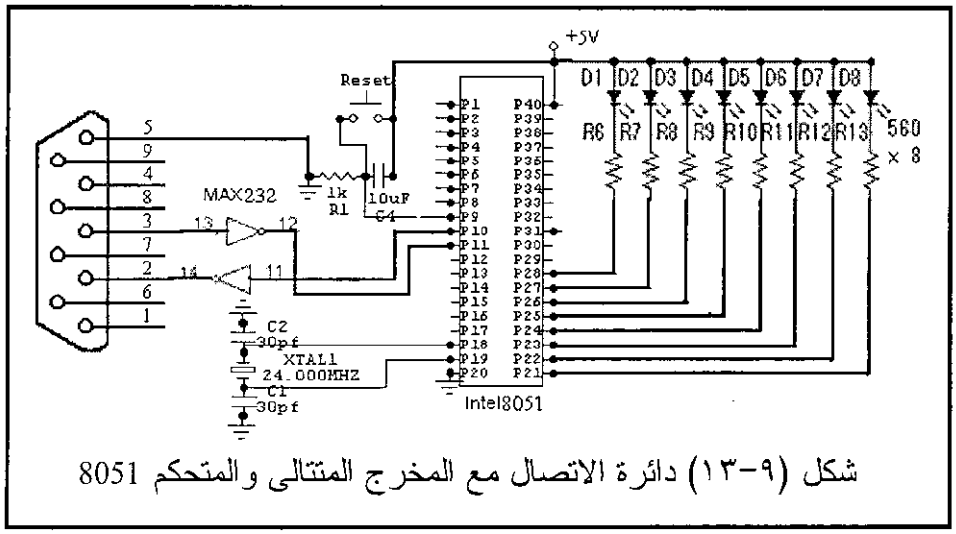
شكل (٩-١٢ب) يبين الرسم الطرفى للشريحة MAX232 حيث نجد أن هذه الشريحة يمكنها أن تستقبل إشارتين TTL على الطرفين ١١ و ١٠ وهما TR1IN و TR2IN وتحولهم إلى RS232 على الطرفين ١٤ و ٧ وهما TR1OUT و TR2OUT . أيضا يمكن لهذه الشريحة أن تستقبل إشارتين RS232 على الطرفين ١٣ و ٨ وهما RE1IN و RE2IN وتحولهم إلى TTL على الطرفين ١٢ و ٩ وهما RE1OUT و

RE2OUT . شكل (٩-١٢أ) يبين كيفية توصيل هذه الشريحة حيث نلاحظ أنها تعمل من مصدر قدرة مقداره ٥ فولت وذلك على العكس من الكثير من الشرائح التى تقوم بهذه المهمة والتى تحتاج لأكثر من مصدر قدرة . سنقدم فيما يلى بعض التطبيقات على التراسل المتتالى بين مخرج البيانات المتتالية فى الحاسب وبعض المتحكمات .

## ٩-٦ التحكم فى أجهزة من خلال مخرج الحاسب المتتالى

**مثال :**

الهدف من هذا المثال هو إرسال واستقبال بيانات متتالية بين الحاسب والمتحكم الدقيق 8051 . سنبرمج الحاسب باستخدام لغة الباسيك المرئى VB بحيث يستطيع الحاسب في البداية أن يتيقن من وجود المتحكم الدقيق علي المسار المتوالي RS232 عن طريق إرسال شفرة معينة للمتحكم ثم يقوم المتحكم بالرد بنفس الشفرة للتأكيد علي اتصاله بالمخرج المتوالي كما أنه يخرج أيضا علي البوابة P2 الرقم 0FH للتأكيد على سلامة الاتصال مع الحاسب . البوابة P2 موصل عليها ثمانية لمبات بيان LEDS يستطيع مستخدم الحاسب أن يجعل أي طرف منها يساوي واحد أو صفر (مضيء أو مطفى) من خلال النقر على زرار فى الباسيك المرئى . شكل (٩-١٣) يبين الدائرة المقترحة لهذا المثال .




شكل (٩-١٣) دائرة الاتصال مع المخرج المتتالى والمتحكم 8051

بعد الانتهاء من بناء الدائرة كما فى شكل (٩-١٣) نبدأ فى كتابة البرنامج الذى سيشغل الدائرة والذى يتكون من جزأين ، جزء سيكتب على الحاسب بلغة الباسيك المرئى والجزء الآخر سيكتب بلغة التجميع الخاصة بالمتحكم وسيتم حرقه على المتحكم نفسه فى ذاكرة البرمجة الخاصة به .

فى برنامج الباسيك المرئى سنتعامل مع المخرج المتتالى من خلال أداة التحكم فـــى الاتصالات Microsoft comm. Port والتى من مميزاتها عند إضافتها علـــى نافـــذة التصميم أنها تمكننا من التعامل مع المخرج المتتالى بسهولة من خلال مجموعة من الدوال التى سنذكر بعضها والتى سنستخدمها فى هذا البرنامج فقط . هذه الأداة لهـــا شكل التليفون كما فى شكل (٩-١٤) . من دوال هذه الأداة ما يلى :

Commport: تحدد رقم المخرج المتوالي الذي سيتم التعامل معه

Portopen: تحدد حالة المخرج المتوالي هل هو مفتوح أم مغلق

Settings:

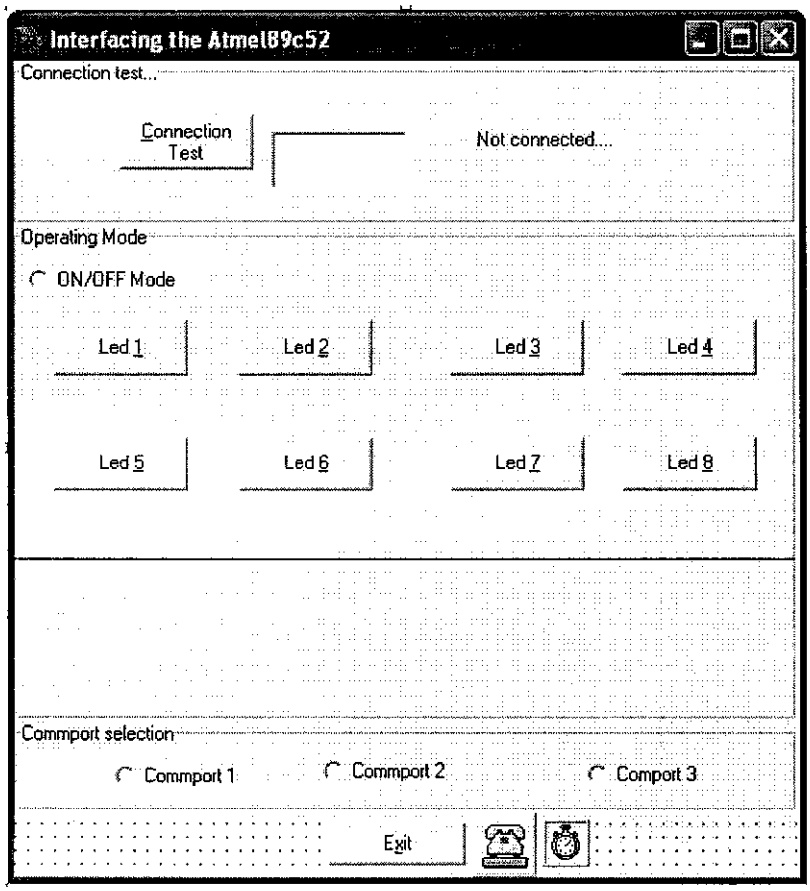
تحدد معاملات المخرج المتوالي مثل معدل إخراج البتات وبت البداية النهاية والباريتى وغير ذلك.

Inputlen: تحدد عدد الحروف التي سيتم قرأتها من مخزن الاستقبال بواسطة خاصية الإدخال

Input: خاصية الإدخال والتي تعود وتمحو سلسلة من الحروف من مخزن الاستقبال

Output: تكتب سلسلة من الحروف في مخزن الإخراج

Commevent: تعود بآخر حدث اتصال أو خطأ




شكل (٩-١٤)

شكل (٩-١٤) يبين محتويات شاشة تصميم برنامج الباسيك المرئى حيث روعى فيها أن المستخدم يقوم أولا باختيار رقم البوابة المتواليـــة Comport1 أو Comport2 أو Comport3 تبعا للمتاح في حاسبه الشخصي . بعـــد ذلـــك يـــتم ضـــغط الـــزرار connection test والذي يتأكد من وجود مسار سليم بين الحاسب والمتحكم الدقيق عن طريق إرسال شفرة معينة (84 مثلا) إلي المتحكم الدقيق . إذا كان المتحكم الـــدقيق متصلا فإنه يرد بنفس الشفرة إلي الحاسب الشخصي ، وفي هذه الحالة فإن الحاسب يعطي رسالة تفيد أن المسار سليم . إذا لم يكن المسار سليما فإن الحاسب بعد فتـــرة انتظار محددة مقدارها ٥ ثوانى مثلا يعطي رسالة تفيد بعدم سلامة الاتصال . الشفرة المطلوبة لذلك هي كالتالى كما فى نافذة البرمجة لهذا الزرار Connection test :

```
Private Sub cmdtest_Click()
 optionport1.Enabled = False
 optionport2.Enabled = False
 optionport3.Enabled = False
 cmdtest.Enabled = False
 MSComm1.commport = commport
 MSComm1.PortOpen = True
 MSComm1.Output = Chr(84)
 Timer1.Enabled = True
End Sub
```

حيث تم إخماد أزرار التعليم الثلاثة فى الثلاثة أوامر الأولى التى بها يختار المستخدم قناة الاتصال التى سيتعامل من خلالها ، كما تم إخماد زرار اختبار الاتصال نفســـه فى الأمر الرابع . الأمر الخامس يضع رقم البوابة المفتوحـــة (١ أو ٢ أو ٣) فـــى المتغير MSComm1.commport ، وبعدها ينشط هذه البوابة ثم يرسل الحرف (84) على البوابة وينشط المؤقت . إذا كان الاتصال سليما بين المؤقت والحاســـب فـــإن المؤقت سيرد بنفس الحرف (84) الذى يتم قراءته بالباسيك المرئى من خـــلال الأداة MSComm كما فى الأوامر التالية فى شاشة برمجة هذه الأداة :

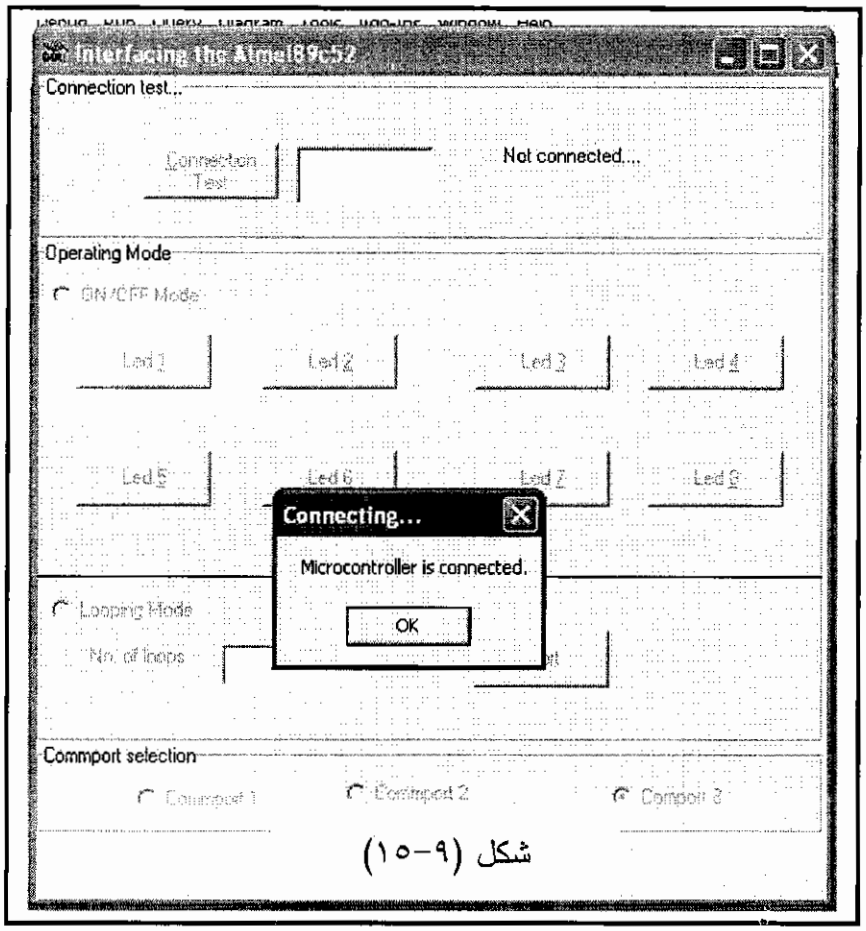
```
Private Sub MSComm1_OnComm()
  MSComm1.InputLen = 1; تحديد عدد الأحرف التى يتم قراءتها من مخزن الإدخال
  Select Case MSComm1.CommEvent
    Case comEvReceive  ;  فى حالة الاستقبال
      commdata = MSComm1.Input  ; وضع الحرف المقروء فى متغير
      If Asc(commdata) = 84 Then ; ٨٤ إذا كان الحرف المقروء هو نفس الحرف
        Timer1.Enabled = False  ; إخماد المؤقت
        Msg = "Microcontroller is connected." ; الرسالة الدالة على سلامة الاتصال
        Style = vbOKOnly  ;  من خواص مربع النص
        Title = "Connecting..."  ; عنوان مربع النص
        Response = MsgBox(Msg, Style, Title)
```

```
        If Response = vbOK Then ; إذا ضغط المستخدم زرار مربع الرسالة
           optionport1.Enabled = False  ; إخماد كل مداخل الاتصال
           optionport2.Enabled = False
           optionport3.Enabled = False
           cmdLed1.Enabled = True  ; تنشيط كل أزرار الإظهار الثمانية
           cmdled2.Enabled = True
           cmdled3.Enabled = True
           cmdled4.Enabled = True
           cmdled5.Enabled = True
           cmdled6.Enabled = True
           cmdled7.Enabled = True
           cmdled8.Enabled = True
           cmdtest.Enabled = False
           Label1.Caption = "connected..."
           Exit Sub
        End If
    Msg = "The Microcontroller is not connected"
    Style = vbOKOnly
    Title = "ERROR"
    Response = MsgBox(Msg, Style, Title)
    End If
  End Select
End Sub
```

شكل (٩-١٥) يبين نتيجة تنفيذ البرنامج وقد ظهرت الرسالة الدالة على سلامة الاتصال وفى نفس الوقت فإن كل الأزرار مخمدة وبمجرد الضغط على زرار OK فى مربع الرسالة فإن أزرار التعامل مع المتحكم الثمانية يتم تنشيطها .
بعد تأكيد الاتصال فإن المستخدم يمكنه الآن أن يقوم بجعل أي طرف من أطراف البوابة P1 واحد أو صفر (يضىء أو يطفىء) من خلال النقر على الزرار المقابل لهذا الطرف من أطراف البوابة الثمانية . الأوامر التالية توضح الأوامر الخاصة بكل زرار والشفرة التى سيتم إرسالها للمتحكم عند النقر على أى واحد من هذه الأزرار .

```
Private Sub cmdLed1_Click()
  MSComm1.Output = Chr$(1)
End Sub
Private Sub cmdled2_Click()
  MSComm1.Output = Chr$(2)
End Sub
  Private Sub cmdled3_Click()
MSComm1.Output = Chr$(4)
```

```
End Sub
Private Sub cmdled4_Click()
  MSComm1.Output = Chr$(8)
End Sub
Private Sub cmdled5_Click()
  MSComm1.Output = Chr$(16)
End Sub
Private Sub cmdled6_Click()
  MSComm1.Output = Chr$(32)
End Sub
Private Sub cmdled7_Click()
  MSComm1.Output = Chr$(64)
End Sub
Private Sub cmdled8_Click()
  MSComm1.Output = Chr$(128)
End Sub
```




شكل (٩-١٥)

بعد أن ينتهى المستخدم من إضاءة أو إطفاء الأزرار التى يريدها يمكنه أن يخرج من البرنامج عن طريق الضغط على الزرار Exit الذى يغلق قناة الاتصال المتتالى وينهى البرنامج .
والآن ننتقل إلى المتحكم لنقدم فيما يلي شفرة لغة التجميع الخاصة به والتى ستتفاعل مع الحاسب من خلال المخرج المتوالى كما ذكرنا .

```
;*******************************************************
;INTERFACING ATMEL89C52 WITH PC
;CRYSTAL 11.05920 MHZ.
;*******************************************************
ORG 0000H
```

الأمر التالي للقفز عبر عناوين خدمة المقاطعة.

```
LJMP MAINLOOP
ORG 0080H
```

الجزء التالي من الشفرة يهيئ البوابة المتوالية ويحدد معدل إخراج البتات ( baud rate) . السطر الأول من الشفرة يحدد التعامل مع المؤقت T1 فقط ، السطر التالي يضع هذا المؤقت في وضع إعادة التحميل الأتوماتيكي . السطر الثالث يحمل المؤقت بالرقم FDH الذي يجعل معدل اخراج البيانات ٩٦٠٠ بت/ ثانية ، ولكن ذلك يحتاج أيضا كما علمنا إلي جعل البت smod بصفر وهو ما تم في السطر الرابع.

```
MAINLOOP:
    ANL tmod, #0fh      ; to alter timer1 configuration only
    orl tmod, #20h      ;timer1,mode2"AUTO RELOAD"
    MOV TH1,#FDH        ;9600 baud
    anl pcon,#7Fh       ;clear smod bit
```

أما الشفرة التالية فتحدد شكل الإرسال المتوالي وهو بت للبداية ثم ٨ بت بيانات ثم بت واحدة للنهاية كما تجعل المتحكم الدقيق جاهزا لاستقبال البيانات عن طريق تفعيل البت (REN = Receive Enable) . السطر الذي يليه يبدأ تشغيل المؤقت T1 لإخراج التردد المطلوب .

```
    MOV SCON,#50H
    SETB TR1
```

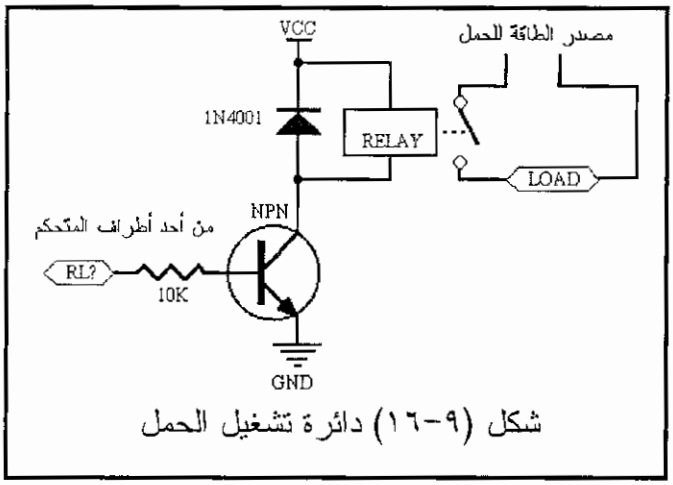
الجزء التالي من الشفرة يجعل المتحكم الدقيق في انتظار لاستقبال أى بيانات من الحاسب . عند وجود بيانات في المسجل SBUF فإنه يتم نقلها إلي المركم ، ومن هنا يبدأ البرنامج فى التعرف على هذه البيانات .

```
here:
    JNB RI,here
```

```
      MOV A,SBUF
      CLR RI
```

في السطر التالي من الشفرة يتم مقارنة المركم بالرقم 84 فإذا لم يساويه فذلك يعني أن الحاسب يريد إضاءة أو إطفاء أحد اللمبات وعندها سيقفز المتحكم إلى حيث ينفذ ذلك ، أما إذا كانت البيانات تساوى 84 فإن ذلك يعنى أن الحاسب يختبر التوصيل بينه وبين المتحكم وسيقوم المتحكم بالرد بنفس البيانات كما يلى :

```
      CJNE A,#84,ONOFF
      MOV A,#84
      ACALL send
      MOV A,#0FH
      MOV P2,A
      LJMP MAINLOOP
ONOFF:
      JNB RI,ONOFF
      MOV A,SBUF
      CLR RI
      MOV P1,A
      lJMP ONOFF
 send:
      MOV SBUF,A
  h:
      JNB TI,h          ; انتظر لإتمام إرسال حرف
      CLR TI            ; بعد الانتهاء من إرسال حرف استعد لإرسال التالى
  RET
END
```



شكل (٩-١٦) دائرة تشغيل الحمل

فى النهاية فإنه بدلا من إنارة لمبة أو إطفاؤها يمكن تشغيل جهاز أو إيقافه من خلال لاقط relay كما فى شكل (٩-١٦) ، بل يمكن تطوير البرنامج بحيث يمكن الاتصال بالحاسب من بعد من خلال خط تليفون أو الشبكة الدولية بحيث يمكن التعامل مع كل هذه الأجهزة من بعد .