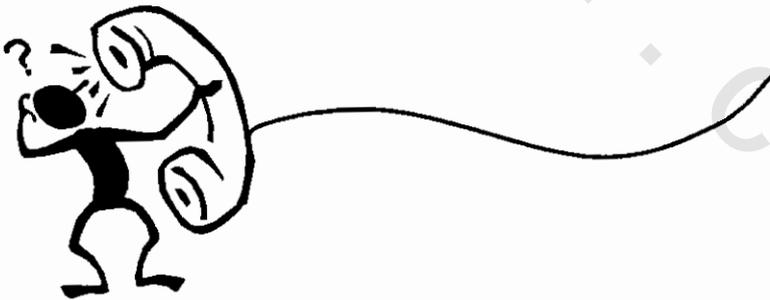


13 الفصل الثالث عشر

المواجهة على التوالي Serial Interfacing



1-13 مقدمة

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

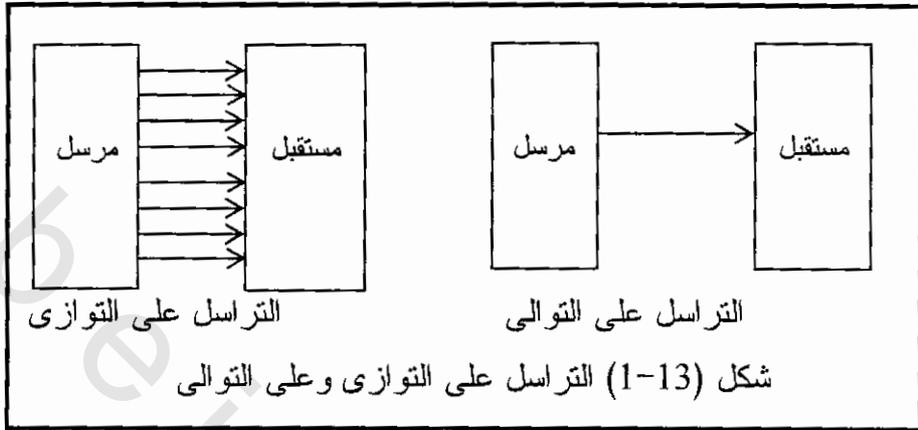
- أن الإشارات الرقمية في أثناء إرسالها على شبكة الاتصالات تكون أقل عرضة للضوضاء المضافة من وسط الانتقال أو من أجهزة التراسل نفسها .
- يمكن إضافة بعض البتات إلى البيانات الرقمية المرسلة لاستخدامها في تصحيح الأخطاء التي قد تطرأ على المعلومة ، هذه البتات تسمى بتات الباريتي parity bits .
- بهذا النظام يمكن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

لقد انتشرت الاتصالات الرقمية الآن انتشارا واسعا وكادت تلغي وجود الاتصالات الانسيابية ، فهناك الآن السنترالات الرقمية ، وشبكات الحاسب ، والفاكس ، والبريد الإلكتروني ، والشبكة الدولية internet والكثير من التطبيقات التي لا يمكن حصرها هنا .

- إرسال البيانات الرقمية بين نقطتين يمكن أن يتم على صورتين مختلفتين :
- 1- الإرسال على التوازي حيث يمكن إرسال بايت كاملة (أو 2 بايت أو 4 أو 8 بايت على حسب عرض مسار البيانات في الحاسب المستخدم) بين المرسل والمستقبل مرة واحدة على 8 (أو 16 أو 32 أو 64) خطوط اتصال تربط بينهما .
 - 2- النظام الثاني هو إرسال البيانات على التوالي حيث يتم إرسال المعلومة بت بعد بت على خط اتصال واحد يربط المرسل والمستقبل .

شكل (1-13) يبين رسما تخطيطيا للطريقتين حيث يمكن أن نستنتج من ذلك أن الإرسال على التوازي يكون أكثر تكلفة نتيجة توصيل 8 خطوط بين المرسل والمستقبل ، ولذلك فإن هذا النوع من التراسل يستخدم في نظم الاتصالات التي لا تمتد لمسافات طويلة . لذلك فإن هذا النظام يستخدم في ربط الأجهزة المحيطة بالحاسب مثل الطابعات ، كما يستخدم في 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 في هذه الحالة

يتم استخدام نظام التراسل على التوالى لتقليل التكلفة والمميزات الأخرى الموجودة فى هذا النظام والتي ستوضح فيما بعد .



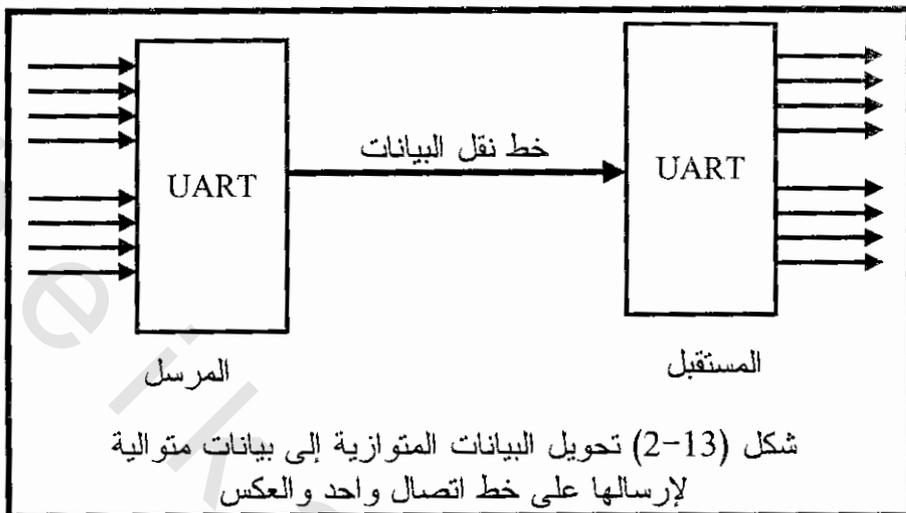
إن الاتصالات على التوالى تكون فى العادة أصعب من الاتصالات المتوازية لأن البيانات تخرج فى العادة من أى معالج فى صورة متوازية ، لذلك لابد من تحويلها إلى الصورة المتوالية باستخدام شريحة تسمى شريحة الإرسال والاستقبال الغير متوافق ، Universal Asynchronous Receiver Transmitter, UART كما فى شكل (13-2) . ولكن بالرغم من ذلك فإن الاتصالات المتوالية لها مميزات يمكن حصرها فيما يلى :

- 1- 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 -25 فولت بينما الصفر المنطقى فيمثله +25 فولت . أى أن كل منهما يختلف عن الآخر بحوالى 50 فولت مما يجعله أقل تأثرا بالضوضاء المضافة .
- 2- عدم الحاجة لأسلاك كثيرة بين المرسل والمستقبل كما رأينا .
- 3- 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 :

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

2- أجهزة الاتصالات Data Communication Equipment, DCE وهذه يمثلها الموديم مثلا أو الراسم plotter أو الطابعة أو أى جهاز يرسل ويستقبل على التوالي .



من أول المسارات القياسية التي استخدمت في الاتصالات على التوالي المسار RS232 الذي تم استخدامه في عام 1962 كأول مسار قياسي يلتزم به كل المشتغلين في هذا المجال . بعد ذلك ظهرت مسارات قياسية أخرى مثل المسار RS422 والمسار RS449 وكل منها يتميز عن سابقه بمعدل البيانات التي يتم نقلها في الثانية ، وطول المسافة التي يتم التراسل عليها . فمثلا المسار RS232 يمكنه أن ينقل البيانات بمعدل 19600 بت في الثانية على مسافات أقل من 20 متر . بينما المسار RS422 فيمكن نقل بيانات عليه بمعدل نقل يصل إلى 10 ميجابايت في الثانية لمسافة أقل من ميل باستخدام كابلات محورية .

من الشفرات التي تستخدم عالميا في نظم التراسل على الكابلات RS232 نظام التشفير ASCII (American Standard Code for Information Interchange) الذي يشفر كل الحروف المطبوعة وغير المطبوعة في 7 بتات ويستخدم البت الثامنة للباريتي . البت الثامنة تكون واحد أو صفر على حسب نوع الباريتي المستخدمة . فإذا كان نظام الباريتي هو النظام الفردي فإن هذه البت تكون واحد أو صفر بحيث تجعل عدد الواحيد في البايت كلها يكون فردي . أما إذا كان نظام الباريتي هو النظام الزوجي فإن هذه البت تكون واحد أو صفر بحيث تجعل عدد الواحيد في البايت كلها عدد زوجي . البرنامج التالي يطبع كل الأحرف

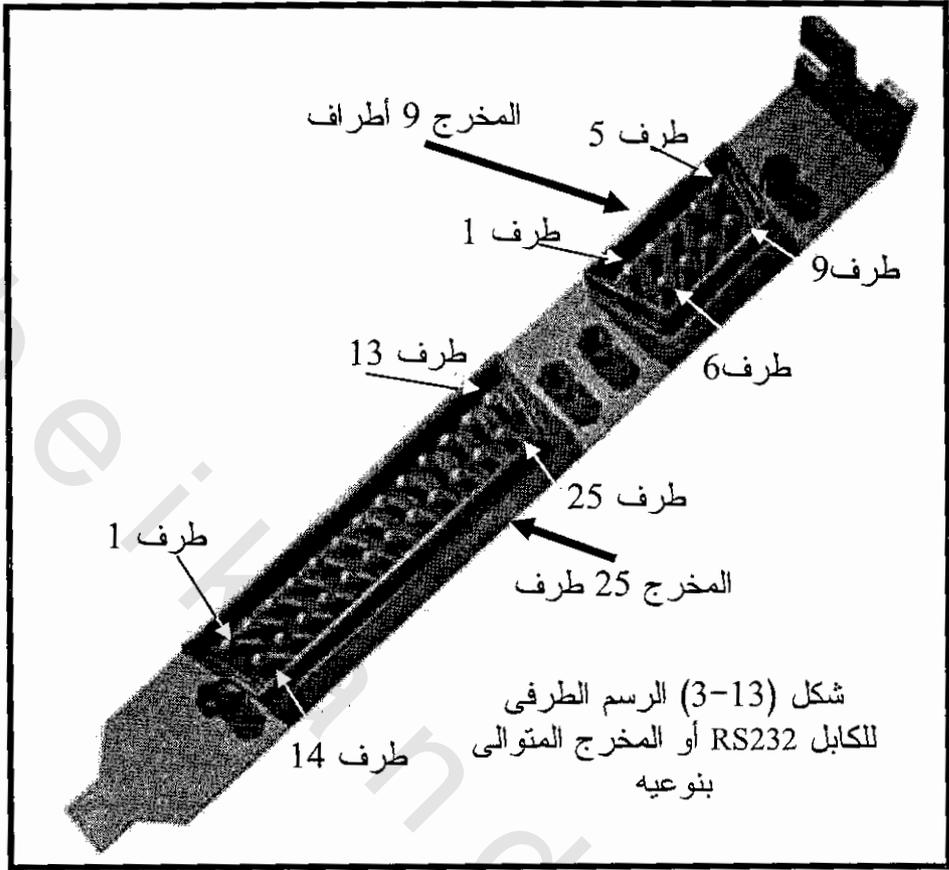
المطبوعة وغير المطبوعة وبجانب كل حرف شفرته ASCII مكتوبة بالنظام العشري :

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

13-2 مواصفات المسار RS232

في المواصفات القياسية للمسار RS232 يمثل الواحد المنطقي بأى جهد مقداره بين 3- فولت و -25 فولت ، كما يمثل الصفر المنطقي بأى جهد مقداره بين 3+ فولت و 25+ فولت ، وأى جهد بين 3- فولت و 3+ فولت يعتبر حالة غير محددة . في حالة عدم إرسال بيانات يكون المسار في الحالة المنطقية واحد أى -25 فولت . ولكن الكثير من التطبيقات الواقعية تستخدم 12+ فولت و -12 فولت بدلا من 25+ و -25 فولت . من ضمن المواصفات القياسية أيضا ألا يتعدى الجهد على أى طرف مفتوح القيمة 25 فولت ولا يتعدى التيار فى أى طرف مقفول 500 ميلي 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 إحداها تسمى COM1 والأخرى تسمى COM2 ، وفى الغالب توصل الفأرة على واحدة منهما ويكون هذا المخرج فى الغالب من النوع ذو 9 أطراف . توجد المخارج التتابعية على واحدة من صورتين ، الأولى هى المخرج ذو 25 طرفا ، والأخرى هى المخرج ذو 9 أطراف . شكل (13-3) 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 الأرضي ، وهو الطرف رقم 1 ، ويتصل به الغلاف الخارجي (الأرضي) للكابل .
- Transmit Data, TD إرسال البيانات ، حيث تخرج البيانات المرسل من الحاسب للموديم على هذا الطرف .
- Receive Data, RD استقبال البيانات ، حيث يتم استقبال البيانات المرسل 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 ، أي 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 جاهزاً لاستقبال البيانات يرد على المرسل (الحاسب) بتنشيط هذا الخط ، ومعناه

- أنه قام بتجهيز المسجلات (تصغيرها) لاستقبال البيانات الجديدة ، أى أن هذه إشارة تحكم داخلة للنهائية الطرفية (الحاسب) .
- Data Send(Set) Ready, DSR ينشط بواسطة الموديم DCE حينما يكون جاهز لإرسال بيانات إلى الحاسب .
 - Signal Ground, SG الأرضى الذى تتسبب إليه إشارة جميع الأطراف .
 - Data Terminal Ready, DTR ينشط بواسطة الحاسب DTE حينما يكون جاهزا لاستقبال بيانات من الموديم . 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 .
 - Carrier Detect, CD ينشط عند استقبال موجة حاملة من الجانب الآخر ، وتعنى أن الموديم موصل الآن على تليفون يعمل .
 - Ring Indicator, RI ينشط لبيان وجود جرس تليفون من الطرف الآخر .
- كل هذه الإشارات تكون فعالة عندما تكون صفر (12 فولت) ، وتكون غير فعالة عندما تكون واحد (-12 فولت) . جدول 1-13 يبين كل طرف من هذه الأطراف على إصدارى المخرج RS232 . لاحظ أن كل الإشارات السابقة كانت تعتبر دائما أن التعامل هو بين حاسب وموديم وهذا هو الأساس لأن الاتصالات على التوالي أول ما استخدمت كانت للاتصال بين الحاسب وموديم .

المخرج 25 طرف	المخرج 9 أطراف	الاسم المختصر	الاسم بالكامل
2	3	TD	إرسال البيانات, Transmit Data
3	2	RD	استقبال البيانات, Receive Data
4	7	RTS	طلب إرسال, Request To Send
5	8	CTS	جاهز فى انتظار الإرسال, Clear To Send
6	6	DSR	بيانات جاهزة عند الموديم, Data Set Ready
7	5	SG	أرضى الإشارة, Signal Ground
8	1	CD	اكتشاف موجة حاملة, Carrier Detect
20	4	DTR	الجهاز الطرفى جاهز, Data Terminal Ready
22	9	RI	وجود جرس تليفون, Ring Indicator

جدول 1-13 بيان بأطراف وأسماء ووظيفة المخارج المتوازية

المسار RS232 يستخدم طريقة التراسل غير التوافقي 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 في شكل (4-13) . نلاحظ في هذا الشكل أن الإرسال يبدأ بنبضة بداية Start bit وهي في الغالب صفر منطقي (0) وبعد ذلك يبدأ إرسال الحرف المراد إرساله في 7 بتات ، ثم توضع البت (أو البتات) الخاصة بالباريتي وذلك أيضا على حسب نوع الباريتي الذي يتم اختياره ، ثم في النهاية يتم وضع 2 بت للدلالة على نهاية الإرسال وتكون في الغالب واحد (1) . شكل (4-13) يبين تفاصيل إرسال الحرف A=1000001 حيث نلاحظ وجود صفر في البداية للدلالة على بداية الحرف ، ثم أضيف واحد في نهاية الحرف ليجعل عدد الواحد فردي حسب نظام الباريتي الفردية ، ثم وضع 11 للدلالة على نهاية الحرف .

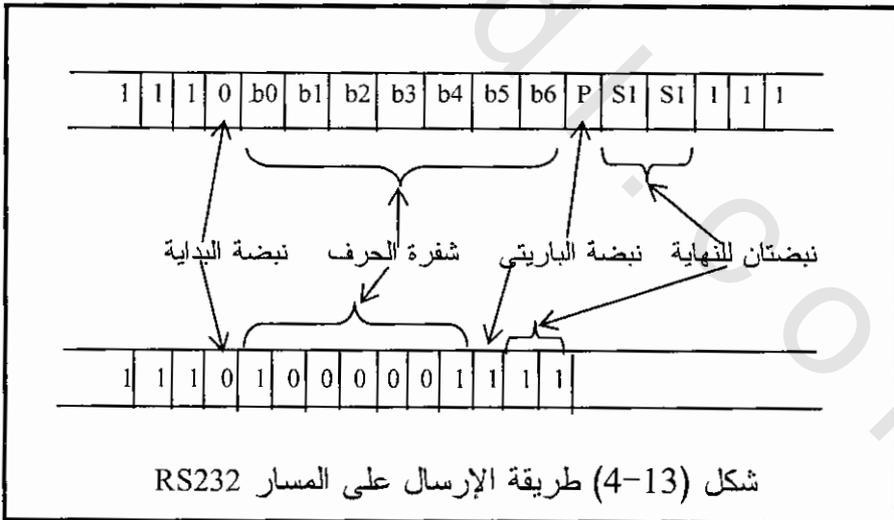
كمثال على الإرسال على المسار RS232 تتبع الرسالة التالية التي تمثل كلمة Apple في نظام الباريتي الزوجية :

$$111111010000010110000011111110000011111110$$

A
p
p

$$0011001111101010011011111$$

l
e



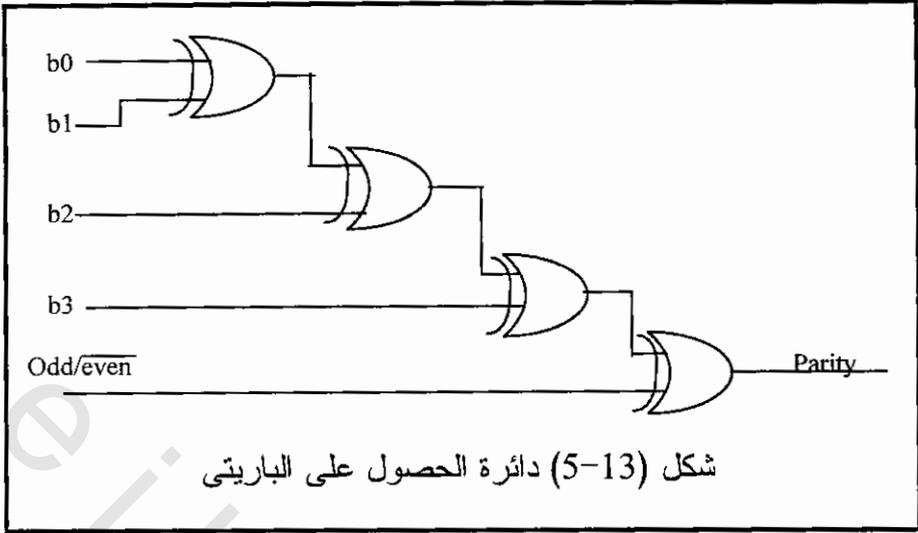
شكل (13-5) يبين استخدام البوابة XOR لبناء دائرة تقوم بتوليد نبضة الباريتي لأربع بتات فقط كما في الشكل . في هذا الشكل يتم وضع البتات التي تمثل الرسالة على الأطراف b_0, b_1, b_2, b_3 ثم يتم اختيار نوع الباريتي (زوجية أم فردية) بوضع صفر أو واحد على خط اختيار الباريتي .

كما رأينا فإن البيانات على المسار RS232 تبدأ عند الإرسال بنبضة البداية Start bit وتكون صفر منطقي ، بعدها تأتي بتات الحرف المراد إرساله بداية بالبت ذات القيمة الصغرى b_0 ، وبعدها b_1 وهكذا حتى آخر بت في الحرف ، ثم يأتي بعد ذلك البت الممثلة للباريتي ، وبعدها البتات التي تحدد نهاية الحرف . أي أن العدد الممثل لأي حرف سيكون 10 بت $(1+1+7+1)$ في حالة استخدام بت واحدة بدلا من 2 للدلالة على نهاية الحرف . فإذا كان معدل إرسال الأحرف هو 960 حرفا في الثانية ، فإن معدل تدفق البتات على المسار سيكون 9600 بت في الثانية ، وهذا ما يطلق عليه معدل بود أو Baud Rate . أي أن معدل بود هو معدل تدفق البتات الممثلة للأحرف المرسل على المسار المتتابع في الثانية . يمكن تحديد زمن البت الواحدة بحساب معدل تدفق البتات . فإذا كان معدل التدفق هو 9600 بت في الثانية مثلا ، فإن زمن البت الواحدة سيكون $(1/9600=104)$ ميكروثانية .

3-13 المصافحة Handshaking

عندما يقوم المرسل بإرسال حرفا على المسار RS232 ، فإن المستقبل عليه أن يقوم باستقبال هذا الحرف وتخزينه في مخزن مؤقت buffer قبل إن يقوم المرسل بإرسال حرفا آخر . إذا لم يحدث ذلك وتأخر المستقبل في عملية التخزين فإن المرسل يمكن أن يرسل حرفا آخر فيكتب فوق حرف سابق عند المستقبل وبالتالي 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 في هذا الجزء . خطوط المصافحة على المسار RS232 كما رأيناها مسبقا هي :

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



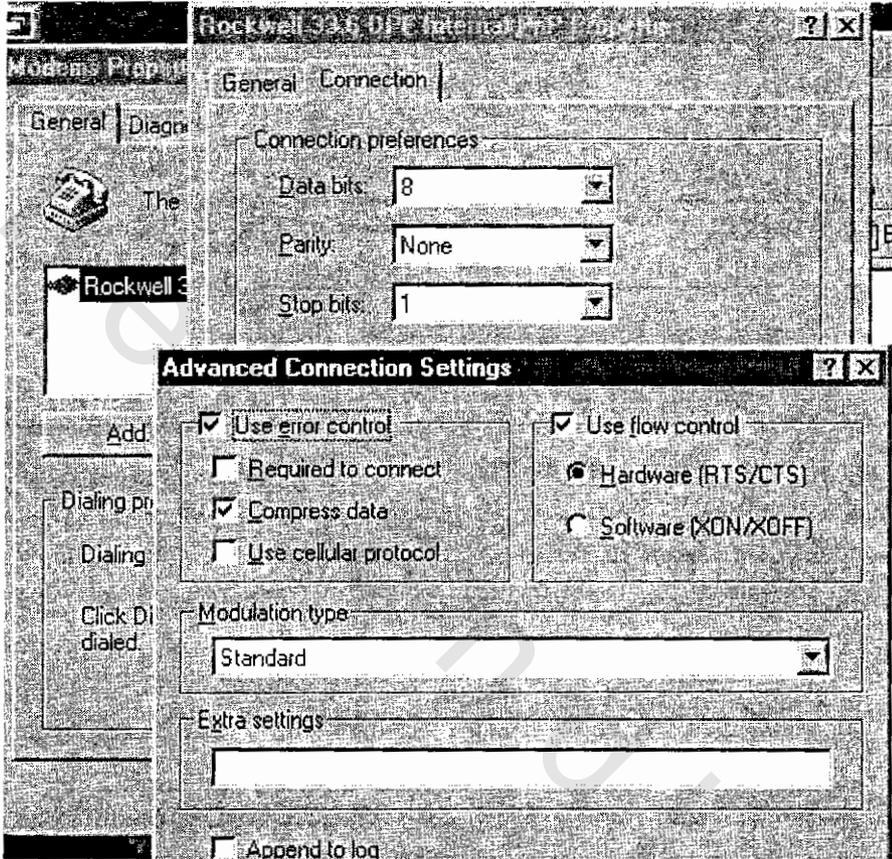
يوجد في السوق ما يسمى بمختبر المسار RS232 وهو عبارة عن دائرة بسيطة تتركب في خرج المرسل ويمكن توصيل كابل المستقبل عليها ، وهذه الدائرة تحتوى لمبات بيان توضح حالة جميع خطوط المصافحة السابقة وخطوط إرسال واستقبال البيانات TD, RD ، وهل هي فعالة أم لا . هذه الدائرة تكون مفيدة بالذات في المراحل الأولى من تصميم دوائر المواجهة أو الاتصالات على التوالي .

أى جهاز حاسب يحتوى على أكثر من مخرج للاتصالات على التوالي وفي العادة يطلق على هذه المخارج COM1 و COM2 و COM اختصار لكلمة communication . فى العادة يتم توصيل الفارة على أحد هذين المخرجين والمخرج الآخر يتاح للاستخدامات الحرة بالمستخدم أو بأحد كروت الاتصالات مثل كارت الفاكس أو كارت الشبكة . يمكن ضبط خواص هذه المخارج من حيث نوع الباريتي ومعدل بود وعدد بتات نهاية الحرف وغيرها من الخواص بالدخول فى برنامج النوافذ بالتتابعات التالية وكما فى شكل (13-6) :

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

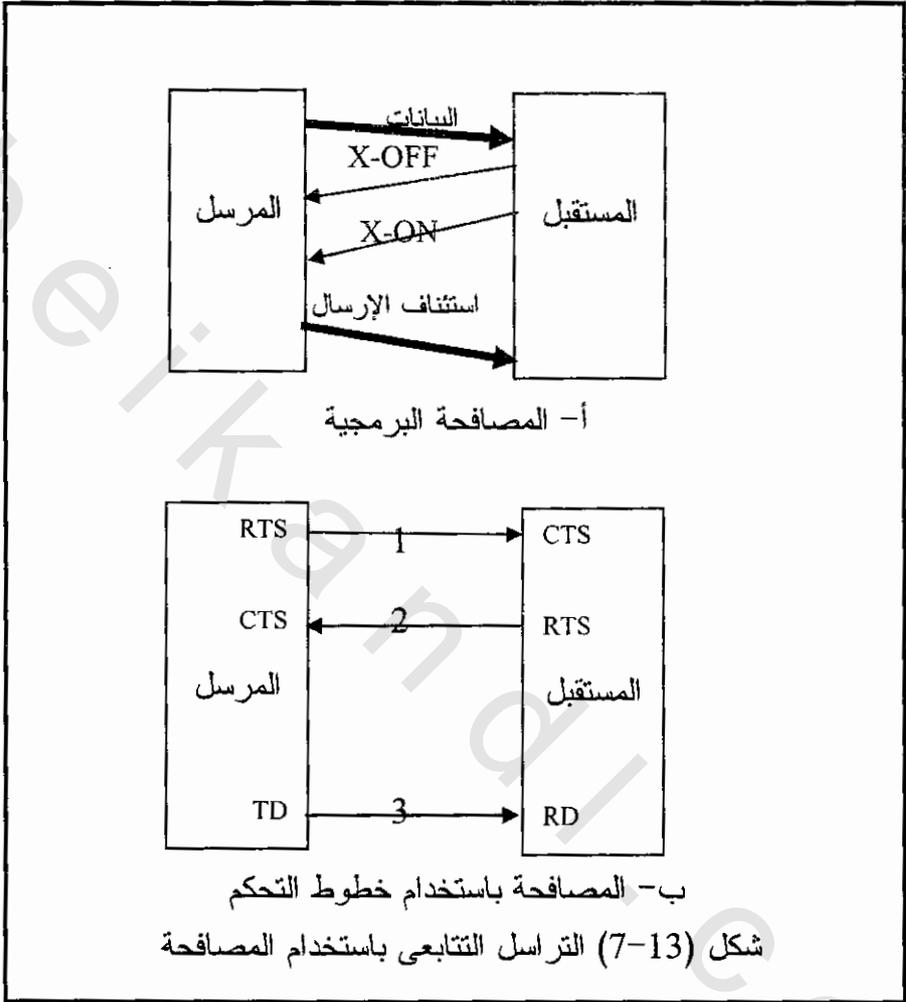
عملية الإرسال . لاحظ أن الحرفان X-ON و X-OFF يستخدمهما المستقبل للسماح للمرسل بعملية الإرسال أو تعطيلها . شكل (13-17) يبين كيفية استخدام هذين الحرفين من قبل المستقبل للتحكم في عملية الإرسال .



شكل (13-6) ضبط خواص المسار RS232 من خلال برنامج النوافذ

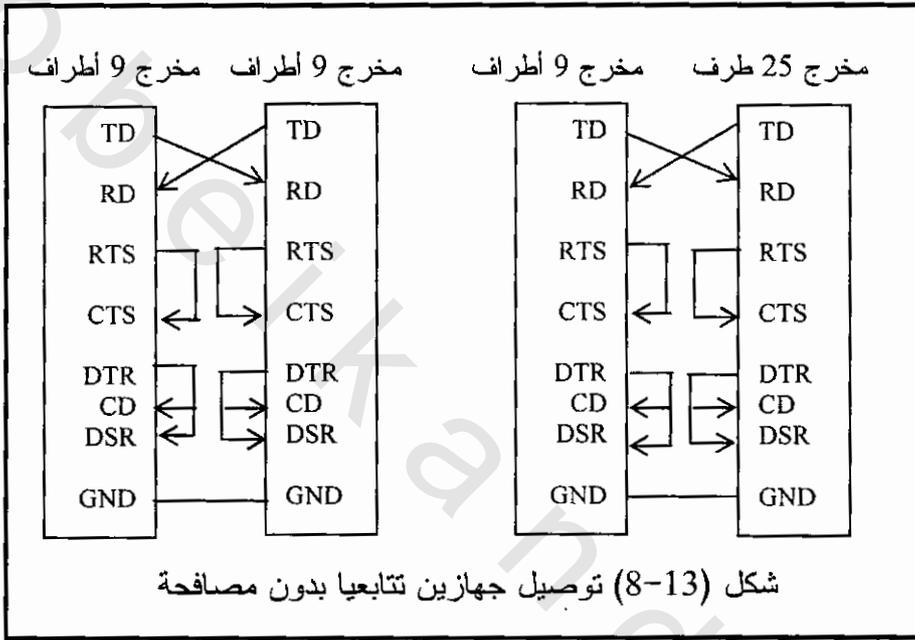
النوع الآخر من المصافحة هو المصافحة باستخدام أطراف التحكم على المسار RS232 . كل خطوط التحكم التي رأيناها على هذا المسار تكون فعالة عندما تكون في حالة الواحد المنطقي high . شكل (13-7ب) يوضح ذلك تماما . عندما يريد أحد الأطراف الاتصال بالطرف الآخر فإنه يقوم بتنشيط الطرف RTS أو جاهز للإرسال الخارج منه . 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 .



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 (8-13) يبين طريقة التوصيل بين جهازين (مرسل

ومستقبل) بدون استخدام المصافحة . لاحظ توصيل الطرف TD ، إرسال البيانات ، في كل من الجهازين على الطرف RD قراءة البيانات ، في الجهاز الآخر . لاحظ أيضا توصيل الطرف RTS الاستعداد للإرسال على الطرف CTS جاهز (clear) للإرسال في كل جهاز وليس مع الجهاز الآخر ، كذلك تم توصيل الطرف DTR مع الطرف DSR في نفس الجهاز والطرف CD .



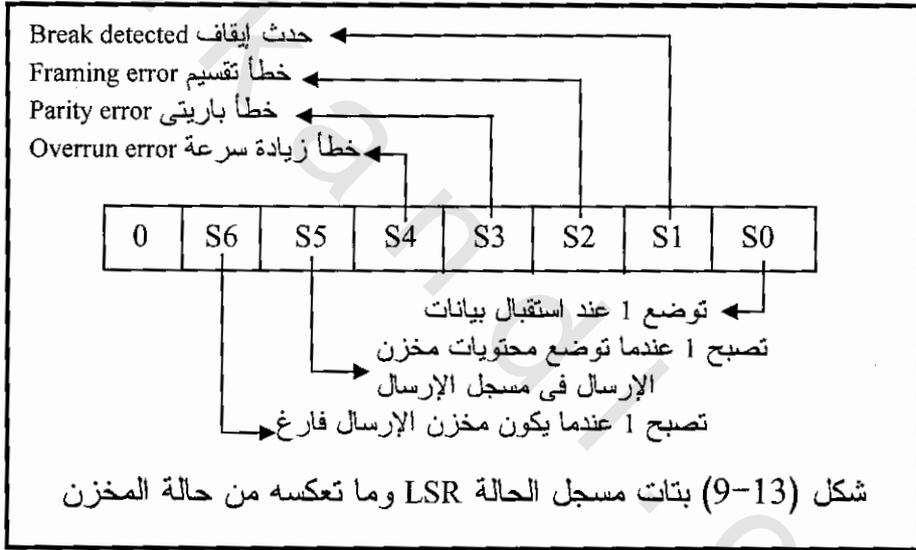
4-13 برمجة مخرج التوالى

- أى مخرج توالى يكون له فى العادة 3 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 2-13 .
 - مسجل التحكم Line Control Register, LCR ويكون عنوانها هو العنوان الأساسى +3 .
 - مسجل الحالة Line Status Register, LSR ويكون عنوانه هو العنوان الأساسى +5 .

مسجل الحالة LSR عبارة عن مسجل يمكن قراءته فقط ولا يمكن الكتابة فيه .
بتات هذا المسجل تعكس حالة المخزن المؤقت عند المرسل والمستقبل . شكل
(9-13) يبين الوظيفة التي تعكسها كل بت من بتات هذا المسجل .

رقم المقاطعة IRQ	عنوان البداية	اسم المخرج
4	3F8	COM1
3	2F8	COM2
4	3E8	COM3
3	2E8	COM4

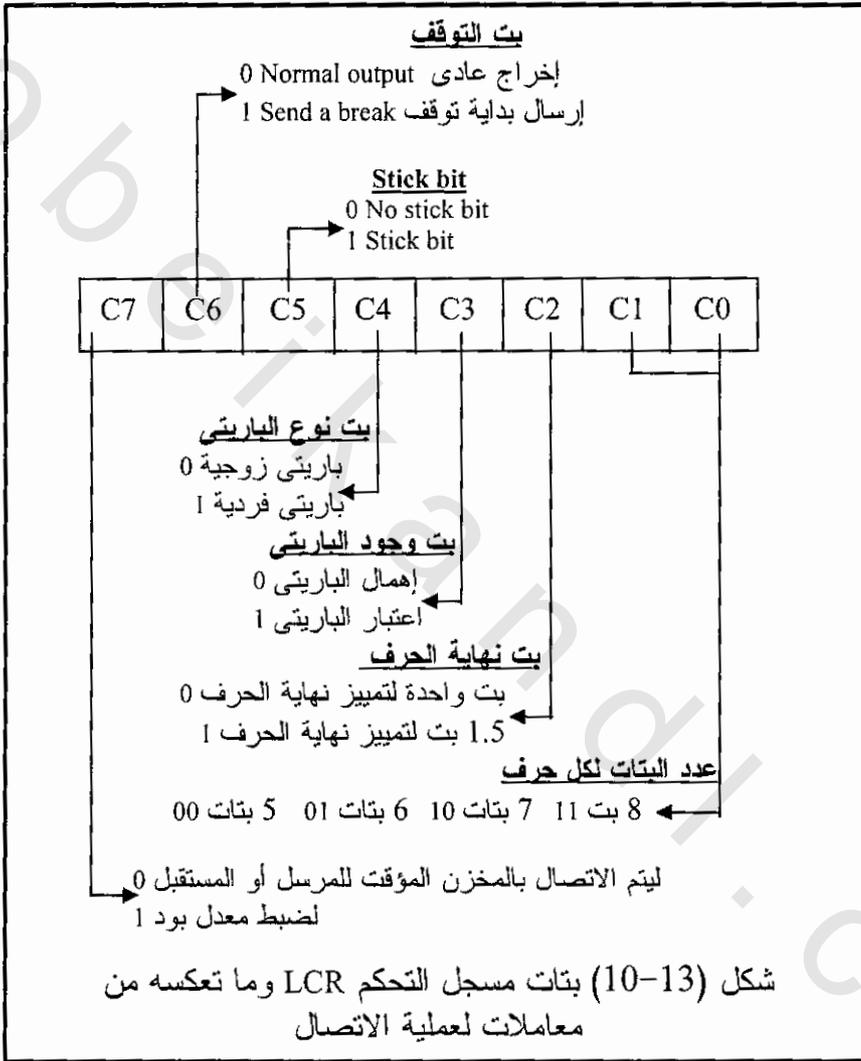
جدول 2-13 مخرج التوالى فى الحاسب و عنوان كل منها



مسجل التحكم LCR يمكن عن طريقه تحديد معاملات عملية الاتصالات مثل عدد البتات التي ستمثل كل حرف ، والباريتى ، وعدد بتات النهاية . هذا المسجل يمكن الكتابة فيه والقراءة منه . شكل (10-13) يبين بتات مسجل التحكم ووظيفة كل بت فى تحديد معاملات الاتصالات .

من الأمور المهمة التي لا بد من الحذر منها هي 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 :



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

بنفس الطريقة عند المستقبل فإن البت $S0=1$ عندما يكون هناك حرف في 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
```

لكي تتمكن من التعامل مع مخزن المرسل أو المستقبل لابد أن تكون البت $C7=0$ في مسجل التحكم . عندما تكون $C7=1$ فإنه في هذه الحالة يمكن ضبط معدل بود عن طريق تحميل رقم من 16 بت في عنوان مخزن الإرسال أو الاستقبال المؤقت والتالي له . هذا الرقم يعتمد على قيمة تردد شريحة الإرسال والاستقبال التتابعي 8250 . هذا الرقم N يمكن حسابه من المعادلة التالية :

$$\text{Baud rate} = \text{Clock frequency} / 16 * N$$

جدول 3-13 يبين قيمة N المقابلة لبعض قيم معدل بود الشهيرة عندما كانت قيمة تردد الشريحة هي 1.8432 ميغاهرتز . تبعا لهذا الجدول فإنه لضبط معدل بود 9600 فإننا يجب أن نحمل العنوان 03F8H بالقيمة 0CH (البايت ذات القيمة الصغرى من القيمة 000CH كما في 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 . في أثناء عملية ضبط معدل بود لابد أن تكون البت C7 تساوى واحد .

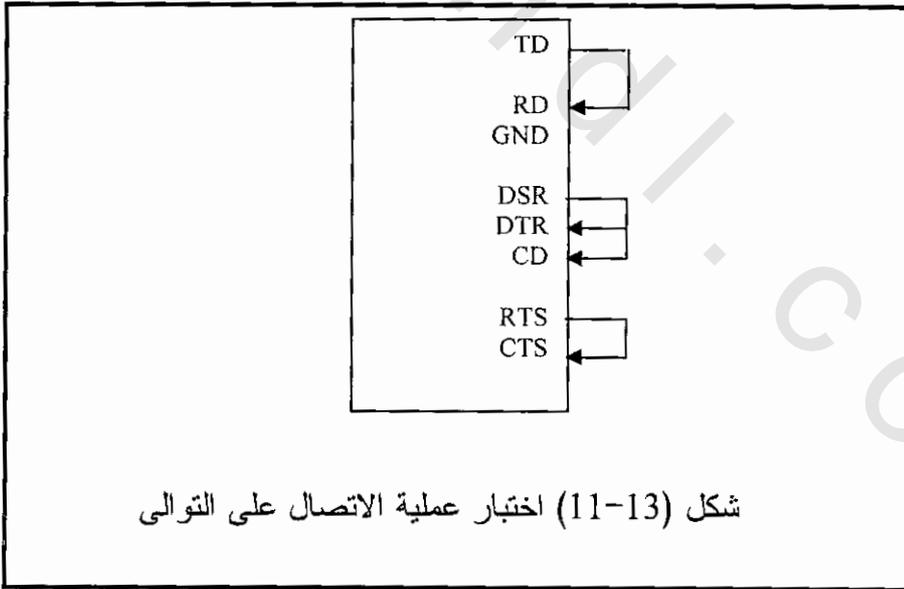
Baud rate	N
110	0417H
300	0180H
600	00C0H
1200	0060H
1800	0040H
2400	0030H
4800	0018H
9600	000CH
19200	0006H

جدول 3-13 ضبط معدل بود

شكل (11-13) يبين كيفية توصيل المخرج التوالى 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 تضرب الحرفين Ctrl+D والتي تمثل نهاية الإرسال EOT حيث يتم الخروج من البرنامج . شفرة الأسكى للحرف Ctrl+D هي 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 ch);
int get_character(void);
```



```
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)
{ outputb(LCR, 0x80H);
  /* this will set bit 7 to 1 to to set baud rate */
  outputb(TXDATA, 0xCH);
  outputb(TXDATA+1, 0x00H);
  outputb(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);
  outputb(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 (13-8) . هذه التوصيلة تسمى كما ذكرنا توصيلة الموديم الافتراضى Null Modem . تعتبر هذه التوصيلة أرخص

وأسهل وسيلة لتوصيل جهازى حاسب حيث أنها تستخدم 3 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 ch);
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)
}

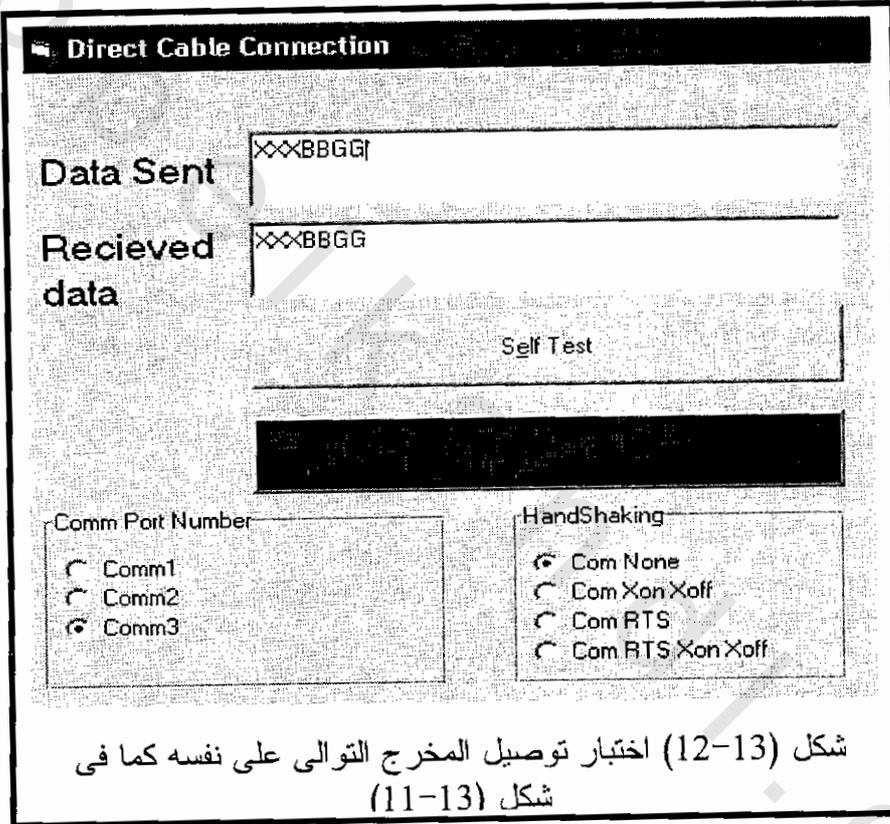
void setup_serial(void)
{ outputb(LCR, 0x80H); /* C7=1 to set register address bit */
  outputb(TXDATA, 0x0CH);
  outputb(TXDATA+1, 0x00H);
  outputb(LCR,0x0AH); /* lock at previous program */
}

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 في الناحية الأخرى . ولا ننسى توصيل طرف الأرضى SG بين الجهازين . في كل جهاز تم توصيل الطرف DTR في كل جهاز على الأطراف DSR و CD في 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 في كل من الجهازين تم توصيله بالطرف CTS وبذلك يمكن الاستغناء تماما عن عملية التحكم في تدفق البيانات بين الجهازين .
 شكل (12-13) يبين نفس البرنامج السابق باستخدام لغة الباسيك المرئى visual basic ، والبرنامج المستخدم في ذلك يمكن كتابته كما يلي :



شكل (12-13) اختبار توصيل المخرج التوالى على نفسه كما في شكل (11-13)

```
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 MSCComm1.PortOpen = False Then
    MSCComm1.PortOpen = True
End If
End If
Do While FLAGE = 1
Dim Instring As String

'Retrieve all available data.
    MSCComm1.InputLen = 0 ' gives max number of recieved data
'Check for data.
    If MSCComm1.InBufferCount Then
'Read data.
        Instring = MSCComm1.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())
MSCComm1.CommPort = Option1(Index).Index + 1
End Sub

Private Sub Option2_Click(Index As Integer())
MSCComm1.Handshaking = Option2(Index).Index
End Sub

Private Sub selftest_Click()
Dim buffer As Variant
'    Set and open port
If MSCComm1.PortOpen = False Then
    MSCComm1.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
```

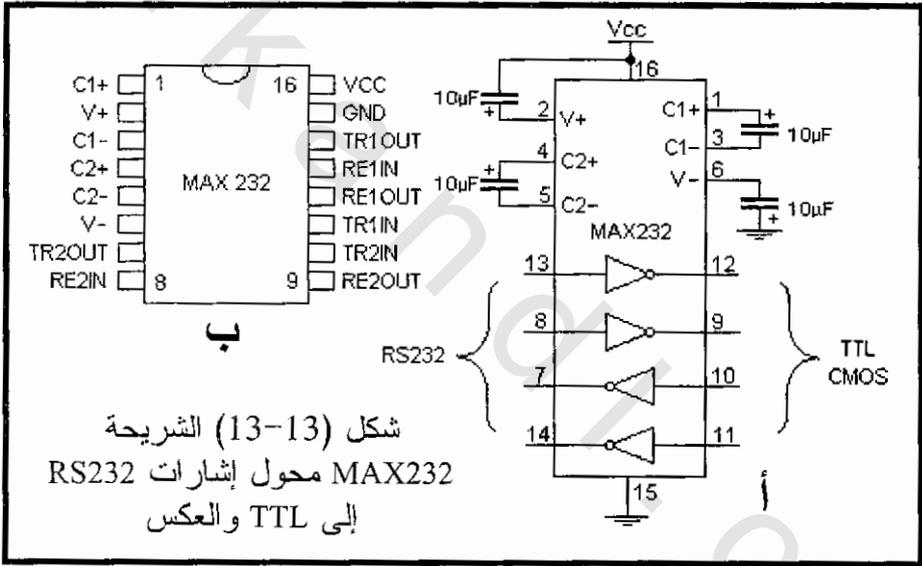
عند تنفيذ هذا البرنامج واختيار رقم المخرج المناسب والنقر على زرار Wait Mode ، ثم وضع دليل الكتابة Cursor في صندوق المرسل فإن أى حرف تتم كتابته في صندوق كتابة المرسل يظهر في الحال في صندوق كتابة المستقبل . يمكن وضع نفس البرنامج في جهاز آخر كمستقبل وتوصيل الجهازين مع بعضهما كما أوضحنا سابقا وفي هذه الحالة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

13-5 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 والصفى المنطقى بصفر فولت . ولقد رأينا هنا أن النظام RS232 يتعامل مع إشارات يمثل الواحد المنطقى فيها بالقيمة 12- فولت ، والصفى المنطقى +12 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 في السوق أكثر من شريحة تقوم بعملية التحويل وسنقدم هنا الشريحة MAX232 التي يمكن استخدامها لهذا الغرض . شكل (13-13ب) يبين الرسم الطرفي للشريحة MAX232 حيث نجد أن هذه الشريحة يمكنها أن ترسل إشارتين TTL على الطرفين 11 و 10 وهما TR1IN و TR2IN وتحولهم إلى RS232 على الطرفين 14 و 7 وهما TR1OUT و TR2OUT . أيضا يمكن لهذه الشريحة أن تستقبل إشارتين RS232 على الطرفين 9 و 8 وهما RE1IN و RE2IN وتحولهم إلى TTL على الطرفين 12 و 13 وهما RE1OUT و RE2OUT . شكل (13-13أ) يبين كيفية توصيل هذه الشريحة حيث نلاحظ أنها تعمل من مصدر قدرة مقداره 5 فولت وذلك على العكس من الكثير من الشرائح التي تقوم بهذه المهمة والتي تحتاج لأكثر من مصدر قدرة .



تمارين

- 1- ما هو أكبر طول للوصلة القياسية RS232 ؟
- 2- إذكر بعض الوصلات التي استخدمت لتحسين هذه المسافة ؟
- 3- كم يبلغ مدى الجهد بين الواحد المنطقي والصفير المنطقي في الوصلة RS232 ؟ وما فائدة هذا المدى الكبير ؟
- 4- كم يبلغ الجهد على الوصلة RS232 الغير فعالة ؟
- 5- ما هو الموديم الافتراضى ؟ وفيما يستخدم ؟
- 6- ارسم وصلة الموديم الافتراضى وشرحها ؟
- 7- كم عدد الأحرف المرسله فى الدقيقة على وصلة تعمل بمعدل إرسال 9600 بت فى الثانية ؟ افترض باريتى بت واحدة و 2 بت لنهاية الحرف وبت لبدايته ؟
- 8- اشرح وظيفة خطوط التحكم التالية : RTS, CTS, DTR, DSR ؟
- 9- اشرح طريقة المصافحة البرمجية soft ware handshaking ؟
- 10- اشرح طريقة المصافحة باستخدام خطوط التحكم المختلفة ؟
- 11- كيف نعرف أن مسجل الإرسال فارغ ويمكن الإرسال ؟ أكتب الكود الذى يقوم بذلك باستخدام لغة C ؟
- 12- كيف نعرف أن مسجل الإستقبال به حرف ويمكن قراءته ؟ أكتب الكود الذى يقوم بذلك باستخدام لغة C ؟
- 13- ارسم بتات مسجل الحالة للوصلة RS232 وشرح وظيفة كل بت ؟
- 14- ارسم بتات مسجل التحكم للوصلة RS232 وشرح وظيفة كل بت ؟
- 15- أكتب برنامج يرسل الحرف A باستمرار على المخرج RS232 واطهر إشارة الخرج على الخط TD على شاشة الأسولوسكوب .
- 16- أعد البرنامج السابق لكل الأحرف الهجائية .