

پروتکل ارتباط سریال

آموختیم که در ارتباط سریال، بین فرستنده و گیرنده تنها یک سیم داده (و یک سیم زمین) وجود دارد. فرستنده بیت‌های داده را به صورت پشت‌سرهم (سریال) روی این سیم قرار می‌دهد و در سمت گیرنده نیز بیت‌ها یک‌به‌یک دریافت شده و کنار هم قرار می‌گیرند. با توجه به اینکه برای ارتباط سریال تنها یک جفت سیم ضروری است، این نوع ارتباط برای تجهیزات مخابراتی و ارتباطات راه دور کاربرد وسیعی دارد. مثلاً ارتباطات در شبکه (با کابل CAT-5 در فواصل چند متری) و اینترنت (خط تلفن در فواصل چند کیلومتری) از این نوع ارتباط استفاده می‌کنند (تصور کنید اگر ارتباطات اینترنتی با اتصال موازی انجام می‌شد، تمام معادن مس دنیا هم برای تأمین این همه سیم کافی نبود!).

در پروتکل ارتباط سریال چند موضوع مهم باید مورد توجه قرار بگیرند:

- نحوه هماهنگی سرعت ارسال و دریافت داده‌ها بین فرستنده و گیرنده
- نحوه قاب‌بندی^۱ اطلاعات (نحوه مشخص کردن آغاز و پایان واحد اطلاعاتی، طول واحد اطلاعاتی و ...)
- مکانیسم کشف خطا
- نحوه کنترل جریان داده

در مورد هر کدام از موارد فوق، شرح مختصری مفید است.

هماهنگی سرعت بین فرستنده و گیرنده

ارتباط سریال را می‌توان مانند نوار نقاله‌ای در نظر گرفت که یک کارگر در یک سمت آن، جعبه‌هایی را روی نقاله قرار می‌دهد و کارگری در سمت دیگر باید جعبه‌ها را از روز نقاله بردارد. بدیهی است سرعت کار هر دو کارگر باید یکسان باشد؛ اگر کارگر اول با سرعت بیشتری کار کند، کارگر دوم فرصت برداشتن جعبه‌ها از روی نقاله را پیدا نمی‌کند. اگر سرعت کارگر اول کمتر باشد، کارگر دوم که سریعتر کار می‌کند بارها به نقاله مراجعه می‌کند بدون اینکه جعبه جدیدی رسیده باشد.

در ارتباط سریال، چون فرستنده داده‌ها را بیت‌به‌بیت روی خط می‌گذارد و گیرنده نیز بیت‌ها را یکی‌یکی از روی خط برمی‌دارد، برای یک ارتباط بی‌نقص باید سرعت فرستنده و گیرنده یکسان باشد؛ وگرنه داده‌های ارسالی به درستی در گیرنده دریافت نمی‌شوند. بنابراین یکی از موارد مهم در برقراری یک ارتباط سریال، توافق بر سر سرعت ارتباط بین فرستنده و گیرنده است.

¹ Framing

هنگامی که با مودم Dial-up به اینترنت متصل می‌شوید، سروصدایی در ابتدای ارتباط از مودم به گوش می‌رسد و سپس ارتباط برقرار می‌شود. این سروصدای مذاکره ابتدایی بین مودم کامپیوتر شما و مودم شرکت تأمین‌کننده اینترنت شما (ISP) است که بخشی از آن مربوط به توافق بر سر سرعت ارتباط است.

سرعت ارتباط سریال با واحد بیت بر ثانیه (bps)^۱ سنجیده می‌شود و نرخ بیتی^۲ نامیده می‌شود^۳؛ مثلاً هنگامی که بعد از اتصال به اینترنت، عبارت 49 Kbps روی مانیتور کامپیوتر نشان داده می‌شود، نشان‌دهنده این است که هم‌اکنون بین مودم کامپیوتر شما و مودم ISP یک ارتباط سریال با سرعت ۴۹۰۰۰ بیت در ثانیه برقرار است^۴. حداکثر سرعت مودمهای Dial-up برابر 56 Kbps است که در عمل از 52 Kbps فراتر نمی‌رود. پورت سریال کامپیوتر از سرعت‌های مختلفی مانند ۱۱۰، ۳۰۰، ۲۴۰۰، ۹۶۰۰ و ... تا ۹۲۱۶۰۰ بیت در ثانیه حمایت می‌کند.

بر اساس نحوه هماهنگی بین فرستنده و گیرنده بر سر سرعت انتقال داده‌ها، دو نوع ارتباط سریال همگام^۵ و غیرهمگام^۶ تعریف می‌شود.

ارتباط سریال همگام

مثال قبلی در مورد نوار نقاله و دو کارگر را در نظر بگیرید. دیدیم سرعت کارگر اول در قرار دادن جعبه‌ها باید با سرعت کارگر دوم در برداشتن جعبه‌ها یکسان باشد. یک روش برای همگام کردن دو کارگر این است که کارگر اول چند لحظه پس از قرار دادن جعبه روی نقاله، زنگی را به صدا درآورد تا کارگر دوم را متوجه کند. کارگر دوم نیز با شنیدن صدای زنگ متوجه می‌شود که وقت برداشتن جعبه فرا رسیده است.

در روش ارتباط سریال همگام، فرستنده به همراه داده‌ها، یک سیگنال ساعت (CLK) نیز ارسال می‌کند. در مواقعی که گیرنده باید خط را بخواند، فرستنده یک لبه بالارونده روی سیگنال ساعت ارسال می‌کند. گیرنده به سیگنال ساعت توجه می‌کند و هرگاه متوجه لبه بالارونده‌ای روی آن شد، خط داده‌ها را خوانده و بیتی که روی آن است را ثبت می‌کند. شکل بعد این نوع ارتباط را نشان می‌دهد.

^۱ Bits Per Second

^۲ Bit Rate

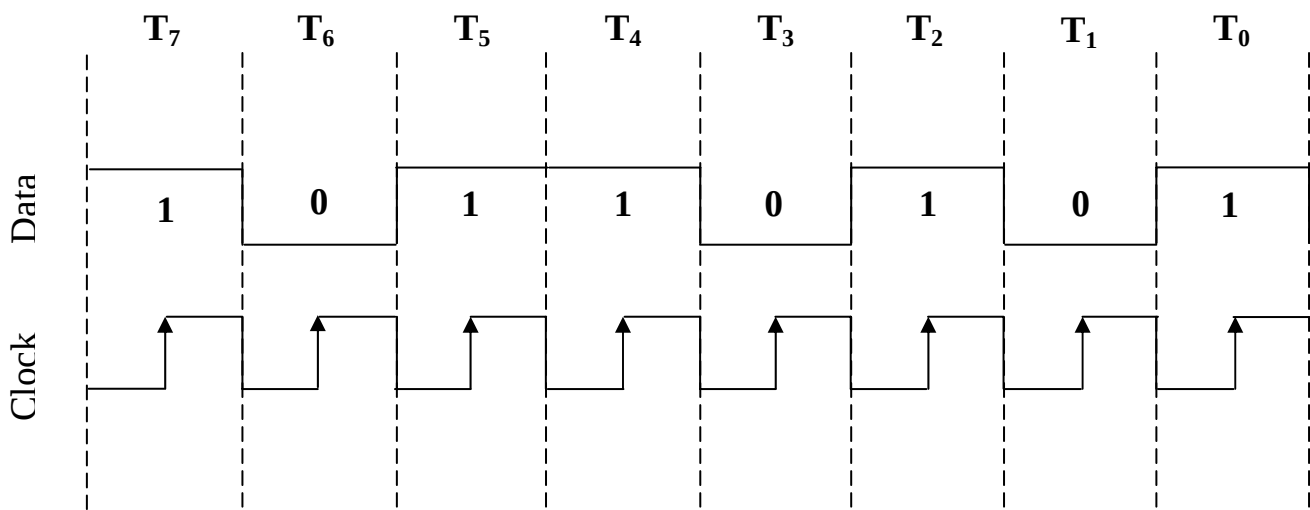
^۳ گاهی در کتابهای مدارهای واسط از این نرخ به عنوان نرخ باود (Baud Rate) نیز یاد می‌شود. نرخ باود در واقع تعداد تغییرات سیگنال داده در ثانیه است که لزوماً در مخابرات با تعداد بیت‌های ارسال یکسان نیست. اما در بحث ما، چون تنها دو ولتاژ وجود دارد که هرکدام نشان‌دهنده یک بیت است، تعداد تغییرات در ثانیه با تعداد بیت در ثانیه یکسان است. بنابراین در این بحث می‌توان نرخ بیتی و نرخ باود را یکسان در نظر گرفت.

^۴ در مخابرات منظور از کیلو، ۱۰۰۰ است. تنها در اندازه‌گیری حافظه کیلو برابر ۱۰۲۴ در نظر گرفته می‌شود.

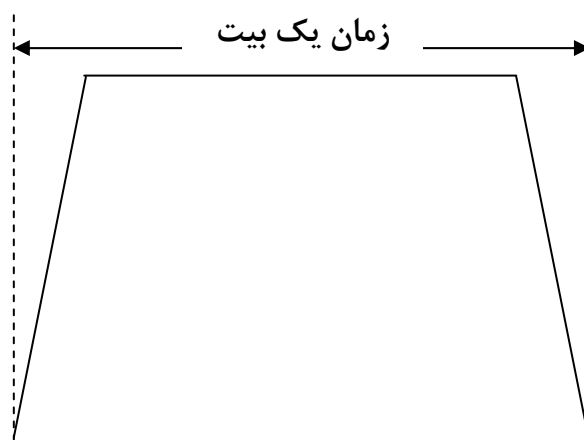
^۵ Synchronous

^۶ Asynchronous

پیشتر در مورد اصلی کامپیوتر یک پورت ۲۵ پین سریال وجود داشت که از این روش برای ارتباط استفاده می‌کرد. ارتباط صفحه کلید با کامپیوتر نیز از این نوع است. در حال حاضر دو استاندارد ارتباط سریال همگام به نامهای BISYNC و SDLC وجود دارد.



همانطور که در شکل بالا دیده می‌شود، فرستنده لبه بالارونده را در «وسط» زمانی که بیت روی خط است روی سیگنال ساعت قرار می‌دهد. شکل زیر دلیل این موضوع را نشان می‌دهد:



همانطور که دیده می‌شود، بیشترین پایداری ولتاژ بیت روی خط داده در «وسط» زمان قرار داشتن بیت روی خط داده است.

ارتباط سریال غیرهمگام

در این نوع ارتباط، دیگر خط CLK بین فرستنده و گیرنده وجود ندارد؛ بنابراین تنها راهی که برای تطابق سرعت بین فرستنده و گیرنده می‌ماند این است که قبل از آغاز ارتباط، نرخ داده بین فرستنده و گیرنده مورد توافق قرار گیرد. این نوع ارتباط سریال کاربردهای فراوانی

دارد. پورت COM کامپیوترهای فعلی که ۹ پین دارد، پورت سریال میکروکنترلرها، ارتباط دستگاه کنترل از راه دور با تلویزیون و ... از جمله این کاربردها هستند.

قاببندی داده‌ها در ارتباط سریال

چون در ارتباط سریال داده‌ها به صورت بیت‌به‌بیت پشت سر هم روی سیم ارتباطی ارسال می‌شوند، باید ابتدا و انتهای یک «مجموعه داده» مشخص باشد. به اصطلاح هر مجموعه داده باید «قاب»^۱ شده و سپس روی خط داده قرار بگیرد.

قاببندی در ارتباط سریال همگام

در ارتباط سریال همگام، چون زمان خواندن هر بیت اطلاعات از روی خط داده دقیقاً مشخص است، فرستنده می‌تواند هر تعداد بیت را پشت‌سرهم ارسال کند (خواهیم دید که در ارتباط غیرهمگام این‌گونه نیست). به بیان دیگر طول بسته اطلاعاتی می‌تواند بزرگ باشد.

ابتدا و انتهای قاب یا بسته اطلاعاتی با الگوهای خاصی که به ترتیب Preamble و Postamble نام دارند مشخص می‌شود. در واقع گیرنده مرتباً خط داده را می‌خواند تا الگوی Preamble را دریافت کند. سپس تا وقتی که الگوی Postamble را دریافت نکرده خط را می‌خواند و اطلاعات روی آن را ثبت می‌کند.

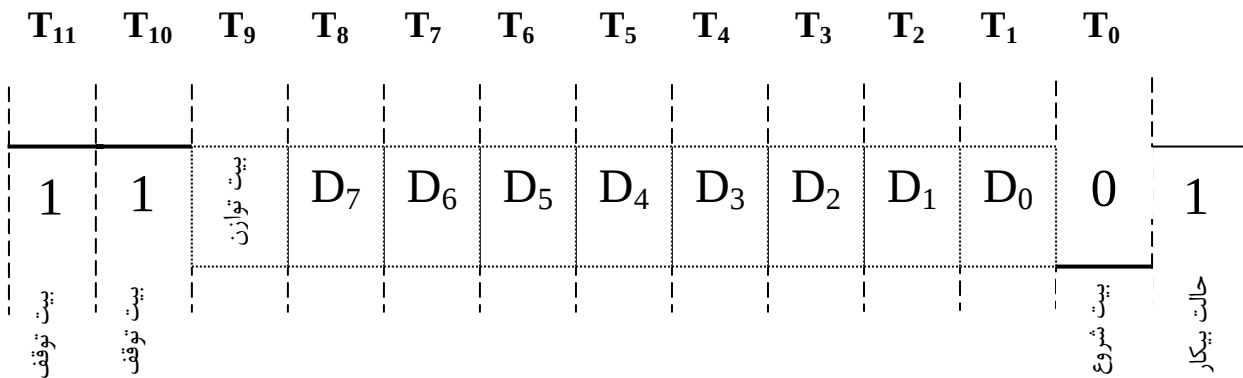
در بعضی طرحهای ارتباط سریال همگام، بعد از الگوی Preamble تعداد بایتهای اطلاعاتی گنجانده شده در بسته توسط فرستنده به گیرنده ابلاغ می‌شود. گیرنده نیز بعد از دریافت Preamble و اطلاعات کنترلی ابتدای بسته، به همان تعدادی که ذکر شده بایتهای اطلاعاتی را از خط می‌خواند.

قاببندی در ارتباط سریال غیرهمگام

همانطور که ذکر شد، در ارتباط غیرهمگام سیگنال ساعت وجود ندارد و نرخ ارسال داده‌ها باید قبل از آغاز ارتباط بین فرستنده و گیرنده مورد توافق قرار گیرد. به همین لحاظ قاب ارتباط سریال غیرهمگام با قاب سریال همگام متفاوت است.

شکل زیر یک قاب داده در ارتباط سریال غیرهمگام را نشان می‌دهد:

¹ Frame



خط ارتباطی سریال غیرهمگام در حالت عادی (بیکار) حاوی «یک» است. هنگامی که فرستنده می‌خواهد ارسال اطلاعات را آغاز کند، باید به نحوی گیرنده را مطلع کند. به همین لحاظ در ابتدای قاب داده، یک بیت «صفر» به نام بیت شروع^۱ ارسال می‌شود تا گیرنده از تغییر «یک» به «صفر» متوجه آغاز ارسال اطلاعات شود.

به دنبال بیت شروع، تعدادی بیت داده با شروع از کم‌ارزشترین بیت (LSB)^۲ ارسال می‌گردد. این تعداد می‌تواند از ۵ بیت (صفحه‌کلیدهای کوچک غیراسکی) تا ۸ بیت (سیستم‌های مهمول انتقال داده) باشد. بیت بعدی، بیت توازن^۳ برای کشف خطا است که می‌تواند از نوع توازن زوج یا فرد باشد. قاب می‌تواند حاوی بیت توازن نباشد.

در انتهای بسته دو بیت «یک» توسط فرستنده ارسال می‌شود که به بیت توقف^۴ معروفند. ارسال این بیت‌های توقف به دو هدف انجام می‌شود: اول اینکه به گیرنده فرصت کافی برای پردازش بسته دریافتی داده داده شود. هدف دیگر این است که خط را به حالت عادی (حالت بیکار که حاوی «یک» است) برگرداند تا آغاز بسته بعدی به کمک بیت شروع که «صفر» است بتواند مشخص شود. تعداد بیت‌های توقف می‌تواند ۱ یا ۱/۵ یا ۲ بیت باشد که معمولاً به تعداد بیت‌های داده وابسته است. در سیستم‌های قدیمی‌تر که کند بودند از دو بیت توقف استفاده می‌شد. در سیستم‌های امروزی استفاده از یک بیت توقف رایج است.

نکته تأمل‌برانگیز در ساختار قاب ارتباط سریال غیرهمگام، تعداد بیت‌های داده است که به حداکثر هشت بیت محدود شده است (البته این تعداد می‌تواند از ۵ تا ۸ بیت متغیر باشد). چون در ارتباط غیرهمگام سیگنال ساعت وجود ندارد، فرستنده و گیرنده باید بر سر یک سرعت به توافق برسند و گیرنده با فرض آن سرعت خط داده را بخواند.

^۱ Start Bit

^۲ Least Significant Bit

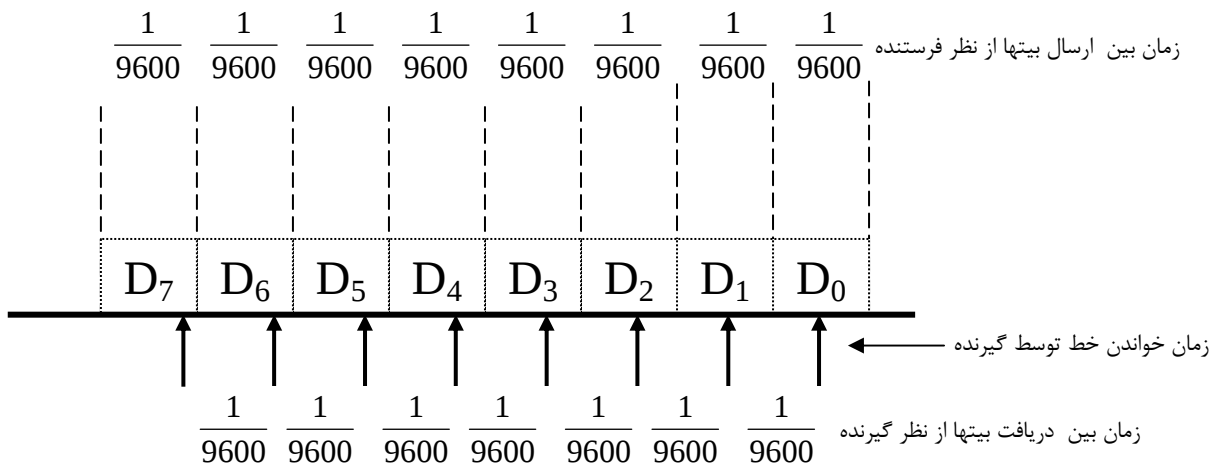
^۳ Parity

^۴ Stop Bit

فرض کنید سرعت توافق شده ۹۶۰۰ بیت در ثانیه است. این بدان معناست که فرستنده در هر ثانیه ۹۶۰۰ بیت ارسال می کند؛ بنابراین زمان ارسال هر بیت $\frac{1}{9600}$ ثانیه است. وقتی گیرنده بیت شروع را دریافت می کند، متوجه آغاز یک بسته اطلاعاتی می شود. پس از سپری شدن زمانی که بیت شروع روی خط قرار دارد ($\frac{1}{9600}$ ثانیه)، گیرنده باید بیت های بعدی را به موقع (یعنی در «وسط» زمانی که بیت روی خط قرار دارد) از روی خط بخواند. بنابراین گیرنده پس از سپری شدن زمانی که بیت شروع روی خط قرار دارد ($\frac{1}{9600}$ ثانیه)، به اندازه نصف زمان یک بیت ($\frac{1}{9600 \times 2}$ ثانیه) صبر می کند و سپس بیت D_0 را می خواند، $\frac{1}{9600}$ ثانیه بعد بیت D_1 را می خواند، $\frac{1}{9600}$ ثانیه بعد بیت D_2 را می خواند و ... تا سرانجام به انتهای بسته برسد. اگر همه چیز طبق برنامه بالا پیش برود، باید فرستنده در هر ثانیه ۹۶۰۰ بیت ارسال و گیرنده این ۹۶۰۰ بیت را به درستی دریافت کند. پس چرا طول قسمت داده قاب اطلاعاتی غیرهمگام محدود به ۸ بیت شده است؟

مشکل اینجا است که همه چیز طبق برنامه جلو نمی رود! فرستنده «با ساعت خودش» یک بیت می فرستد، $\frac{1}{9600}$ ثانیه صبر می کند، بیت بعدی را می فرستد، $\frac{1}{9600}$ ثانیه صبر می کند، بیت بعدی را می فرستد، $\frac{1}{9600}$ ثانیه صبر می کند و ... گیرنده «با ساعت خودش» یک بیت از روی خط می خواند، $\frac{1}{9600}$ ثانیه صبر می کند، بیت بعدی را از روی خط می خواند، $\frac{1}{9600}$ ثانیه صبر می کند، بیت بعدی را از روی خط می خواند، $\frac{1}{9600}$ ثانیه صبر می کند و ...

مسئله همین جاست که ساعت فرستنده با ساعت گیرنده یکسان نیست. در واقع در سیستم های مبتنی بر پردازنده برای تولید پالس ساعت از اجزاء الکترونیکی طبیعی (مانند کریستال) استفاده می شود که پیدا کردن دو جزء کاملاً یکسان در طبیعت تقریباً غیرممکن است. به همین لحاظ زمان $\frac{1}{9600}$ ثانیه در فرستنده با زمان $\frac{1}{9600}$ ثانیه در گیرنده متفاوت است و همین اختلاف زمان باعث ایجاد مشکل می شود. شکل زیر نشان می دهد اگر ساعت گیرنده سریعتر از ساعت فرستنده کار کند چه مشکلی ایجاد می شود:



گیرنده و فرستنده توافق کرده‌اند زمان بین بیتها $\frac{1}{9600}$ ثانیه باشد و هرکدام بر اساس ساعت خود این زمان را اندازه می‌گیرند. همانطور که می‌بینید بعد از خواندن هر بیت، لحظه خواندن خط توسط گیرنده کمی زودتر اتفاق می‌افتد. بعد از دریافت ۸ بیت اطلاعات، لحظه خواندن خط توسط گیرنده از «وسط» زمانی که بیت روی خط دارد فاصله زیادی پیدا کرده و اگر تعداد بیت‌های داده بیشتر شود امکان دارد گیرنده زمانی خط را بخواند که هنوز بیت روی خط نیامده است.

پرسش) با رسم شکل توضیح دهید اگر ساعت گیرنده کندتر از ساعت فرستنده کار کند چه مشکلی ایجاد می‌شود؟

بنابراینچه گفته شد، طول بیت‌های داده بسته اطلاعاتی در ارتباط غیرهمگام محدود به ۸ بیت است و در ادامه ممکن است فرستنده و گیرنده از همگامی خارج شوند.

ممکن این سؤال برای شما پیش بیاید که چرا این مشکل در ارتباط همگام وجود ندارد؟ پاسخ واضح است. در ارتباط همگام، فرستنده به کمک سیگنال ساعت که به همراه داده‌ها ارسال می‌کند به گیرنده اعلام می‌کند که چه زمانی باید خط را بخواند. به همین جهت فرستنده و گیرنده هیچگاه از همگامی خارج نمی‌شوند. اما در ارتباط غیرهمگام چون سرعت قبل از ارتباط مورد توافق قرار می‌گیرد و سپس فرستنده و گیرنده هرکدام طبق ساعت خودشان عمل می‌کنند، به علت عدم تطابق کامل بین ساعت فرستنده و ساعت گیرنده، امکان دارد بعد از ارسال تعدادی بیت فرستنده و گیرنده از همگامی خارج شوند. به همین لحاظ، تعداد بیت‌های داده به حداکثر ۸ بیت محدود شده است. در بسته اطلاعاتی بعدی، به کمک بیت شروع همگامی مجدداً برقرار می‌شود.

مقایسه بین ارتباطات همگام و غیر همگام

هرچند هم‌اکنون بیشتر ارتباطات سریال از نوع غیر همگام هستند، اما مقایسه‌ای بین ارتباطات همگام و غیر همگام مفید است:

- در ارتباط همگام طول بسته اطلاعاتی محدود نیست (مگر اینکه محدودیتهایی از ناحیه لایه‌های بالاتر ارتباطی اعمال شده باشد).
- در ارتباط همگام، فرستنده در حین ارتباط می‌تواند با تغییر سیگنال ساعت، سرعت ارتباط را تغییر دهد (هرچند این کار معمول نیست). اما در ارتباط غیر همگام سرعت باید از ابتدا بین فرستنده و گیرنده مورد توافق قرار بگیرد.
- سربار^۱ اطلاعات در ارتباط غیر همگام بالا است. در این نوع ارتباط برای ارسال ۸ بیت اطلاعات باید یک قاب ۱۲ بیتی ارسال شود که ۴ بیت آن سربار است. بنابراین تنها از $\frac{8}{12} = 66\%$ پهنای باند خط استفاده می‌شود. در ارتباط همگام تعداد بیت‌های داده بیشتر است و به کارایی بیشتری در استفاده از خط ارتباطی می‌انجامد.
- در ارتباط غیر همگام نیازی به سیگنال ساعت (CLK) نیست که این موضوع در کاهش هزینه ارتباط مؤثر است. به علاوه در ارتباطات بیسیم (مانند دستگاه کنترل از راه دور تلویزیون که از سیگنال‌های مادون قرمز استفاده می‌کند یا دزدگیر رادیویی) ارسال دو سیگنال داده و ساعت منطقی به نظر نمی‌رسد.

کنترل جریان داده

پیشتر دیدیم فرستنده باید حجم اطلاعات ارسالی را کنترل کند تا به گیرنده فرصت پردازش اطلاعات بدهد. این موضوع را کنترل جریان داده نامیدیم.

آیا کنترل جریان داده بحثی مجزا از تطابق سرعت بین فرستنده و گیرنده است؟ به بیان دیگر، اگر سرعت فرستنده و گیرنده با هم تطابق داشته باشد، باز هم نگرانی در مورد کنترل جریان داده وجود دارد؟

پاسخ مثبت است. برای مثال فرض کنید شخصی اطلاعاتی برای شما می‌خواند تا یادداشت کنید. ممکن است آن شخص آنقدر تند صحبت کند که شما اصلاً متوجه حرف‌های او نشوید. این مشکل **عدم تطابق سرعت بین فرستنده و گیرنده** است. اما در حالتی دیگر ممکن است آن شخص طوری صحبت کند که شما متوجه حرف‌های او بشوید، اما آنقدر اطلاعات را به سرعت بخواند که شما موفق به یادداشت کردن اطلاعات نشوید. این مشکل **عدم کنترل**

¹ Overhead

صحیح جریان داده است. بنابراین در ارتباط سریال، هم تطابق سرعت بین فرستنده و گیرنده و هم راهکاری برای کنترل جریان داده لازم است.

در ارتباطات سریال غیرهمگام دو نوع کنترل جریان داده وجود دارد. یکی روش سخت‌افزاری است که پس از معرفی درگاه سریال کامپیوتر به آن خواهیم پرداخت. روش دیگر کنترل جریان داده به شیوه نرم‌افزاری است. در این روش، هرگاه بافر گیرنده در شرف پر شدن باشد و نتواند اطلاعات جدید را ثبت کند، یک کاراکتر ویژه به نام Xoff^۱ برای فرستنده ارسال می‌کند. فرستنده با دریافت این کاراکتر، ارسال اطلاعات را متوقف می‌کند و صبر می‌کند تا گیرنده آمادگی خویش برای دریافت اطلاعات جدید را با ارسال کاراکتری ویژه به نام Xon^۲ اعلام کند. این روش کنترل جریان داده به Xon/Xoff موسوم است.

مشخصات ارتباط غیرهمگام

به صورت خلاصه در ارتباط غیرهمگام باید مشخصات زیر بین فرستنده و گیرنده مورد توافق قرار بگیرد:

- نرخ داده (مثلاً ۹۶۰۰ یا ۱۹۲۰۰ بیت در ثانیه)
- تعداد بیت‌های داده (۵ یا ۶ یا ۷ یا ۸ بیت)
- نوع توازن به کار رفته (زوج، فرد، بدون توازن، همیشه «یک»، همیشه «صفر»)
- تعداد بیت‌های توقف (۱ یا ۱/۵ یا ۲ بیت)
- روش کنترل جریان داده (سخت‌افزاری، Xon/Xoff یا بدون کنترل جریان داده)

تراشه‌های ارتباط سریال

به لحاظ پیچیدگی‌هایی که در مدیریت ارتباط سریال وجود دارد، تراشه‌هایی وجود دارند که عمل تبدیل داده‌های موازی به سریال و بالعکس را انجام می‌دهند. این تراشه‌ها قابل برنامه‌ریزی هستند و می‌توانند ارتباطات سریال با مشخصات مختلف را حمایت کنند. یک دسته از این تراشه‌ها UART^۳ نامیده می‌شوند که برای ارتباطات غیرهمگام به کار می‌روند. از جمله این تراشه‌ها می‌توان به ۸۲۵۰ یا ۱۶۴۵۰ اشاره کرد. اکثر میکروکنترلرها یک یا دو کنترل‌کننده UART داخل خود دارند. دسته دیگر از این تراشه‌ها USART^۴ نام دارند و برای مدیریت هر دو نوع ارتباط همگام و غیرهمگام به کار می‌روند که تراشه ۸۲۵۱ از جمله آنهاست.

^۱ کاراکتر اسکی DC3 که با فشردن کلیدهای Ctrl+S تولید می‌شود.

^۲ کاراکتر اسکی DC1 که با فشردن کلیدهای Ctrl+Q تولید می‌شود.

^۳ Universal Asynchronous Reciever/Transmitter

^۴ Universal Synchronous/ Asynchronous Reciever/Transmitter

در گذشته، کامپیوترها از هر دو نوع ارتباط همگام و غیرهمگام حمایت می‌کردند. اما در بوردهای کامپیوتر امروزی، تنها یک نوع پورت سریال یافته می‌شود که ویژه ارتباط غیرهمگام است. برای مدیریت این ارتباط به یک تراشه UART نیاز است. مودمهای خارجی که به پورت سریال متصل می‌شوند این تراشه را دارند؛ اما مودمهای ارزان قیمت فاقد این تراشه هستند و پردازنده کامپیوتر مدیریت ارتباط سریال را انجام می‌دهد که باعث کاهش سرعت ارتباط می‌شود (این مودمها در اصطلاح WinModem یا مودمهای نرم‌افزاری نامیده می‌شوند). در کامپیوترهای امروزی تراشه UART در تراشه SUPER I/O یا South Bridge که تراشه اصلی ارتباطی برد اصلی کامپیوتر است، گنجانده می‌شود. جدول زیر ویژگیهای تراشه‌های UART گوناگونی که در بوردهای کامپیوتر استفاده می‌شوند را نشان می‌دهد.

نوع UART	حداکثر سرعت	طول بافر	نوع سیستم
۸۲۵۰	بالای 9600 bps	-	۸۰۸۸ (XT)
۸۲۵۰A	بالای 9600 bps	-	۸۰۸۸ (XT)
۸۲۵۰B	بالای 9600 bps	-	۸۰۸۸ و ۸۰۲۸۶
۱۶۴۵۰	بالای 19200 bps	۱ بایت	۸۰۳۸۶ و ۸۰۴۸۶
۱۶۵۵۰	بالای 115200 bps	۱۶ بایت	۸۰۳۸۶ و ۸۰۴۸۶ و پنتیوم
۱۶۶۵۰	بالای 230000 bps	۳۲ بایت	کارت‌های خاص I/O و ترمینالهای داخلی ISDN
۱۶۷۵۰	بالای 460000 bps	۶۴ بایت	کارت‌های خاص I/O و ترمینالهای داخلی ISDN
۱۶۹۵۰	بالای 921.6 kbps	۱۲۸ بایت	کارت‌های خاص I/O و ترمینالهای داخلی ISDN

سه تراشه آخر از استاندارد به نام ESP^۱ (پورت سریال با سرعت بالا) پیروی می‌کنند. در ذیل به مشخصات پورت سریال غیرهمگام کامپیوتر و نحوه تبادل اطلاعات با آن می‌پردازیم.

پورت سریال کامپیوتر



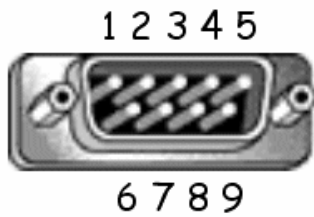
اگر به پشت برد اصلی کامپیوتر خود نگاه کنید، یک پورت با ۹ پایه مانند شکل مقابل می‌بینید که به پورت سریال، پورت COM^۲ یا در اصطلاح فنی الکترونیک به پورت DB9 معروف است. به کمک این پورت، می‌توان یک ارتباط سریال غیرهمگام با مشخصات قابل تغییر داشت^۳.

^۱ Enhanced Serial Port

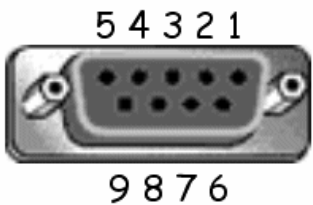
^۲ COMMunication - این نام به دلیل کاربرد این پورت در ارتباطات و مخابرات به آن داده شده است.

^۳ پیشتر در برد اصلی کامپیوتر یک پورت ۲۵ پین سریال وجود داشت که از روش همگام برای ارتباط استفاده می‌کرد. امروزه این نوع پورت را بیشتر در دستگاههای صنعتی می‌توان یافت.

در ادامه به بررسی سیگنالهای پورت سریال کامپیوتر می‌پردازیم. با شناخت این سیگنالها دو مطلب را خواهیم آموخت؛ اول چگونگی عملکرد مودمهای خارجی^۱ در تعامل با کامپیوتر و دوم نحوه کنترل یک دستگاه جانبی به کمک پورت سریال کامپیوتر.



پیش از توضیح عملکرد پایه‌ها، توضیحی راجع به نحوه شماره‌گذاری پایه‌ها لازم است. پورتی که پشت کامپیوتر مشاهده می‌کنید، دارای ۹ میله است که سیگنالها را در اختیار کاربر قرار می‌دهد.^۲ شماره‌گذاری این حفره‌ها به صورت مقابل است:



معمولاً برای استفاده از این پورت، قطعه‌ای به کار می‌رود که به جای میله دارای حفره است و طوری روی پورت سریال کامپیوتر نصب می‌شود که میله‌ها درون حفره‌ها قرار می‌گیرند.^۳ این قطعه نیز به نام DB9 معروف است. شکل مقابل را ببینید. سر کابل مودم خارجی (یا دستگاههای دیگر مانند اسکنر که به پورت سریال متصل می‌شوند) نیز چنین شکلی دارد.



دقت کنید که در پورت اصلی کامپیوتر، پایه بالا سمت چپ و در قطعه دومی که روی پورت نصب می‌شود، پایه بالا سمت راست شماره یک است. چون وقتی قطعه بالا روی پورت سریال نصب می‌شود، سمت چپ آن با سمت راست پورت مقابل هم قرار می‌گیرد. پشت این قطعه در محل

هر میله، جایی برای لحیم کردن سیم تعبیه شده تا به راحتی بتوانید از سیگنالهای پورت سریال استفاده کنید. شکل زیر را ببینید:

استاندارد ارتباط سریال کامپیوتر

در سال ۱۹۶۰ سازمان صنایع الکترونیک (EIA)^۴ برای استاندارد کردن ارتباط سریال بین وسایل مختلف (از جمله کامپیوترها) پروتکلی پیشنهاد کرد که به EIA232 یا RS232 معروف شد.^۵ بخش سخت‌افزاری این پروتکل شامل سیگنالهای داده و کنترلی است که در ادامه با آشنا خواهیم شد.

^۱ External

^۲ این نوع پورت معمولاً به Plug یا Male معروف است.

^۳ این نوع پورت معمولاً به Socket یا Female معروف است.

^۴ Electronics Industries Association

^۵ کامپیوترهای اپل مکینتاش (که دارای پردازنده موتورولا هستند) دارای رابط سریال با استاندارد RS422 هستند.

ولتاژهای منطقی در RS232

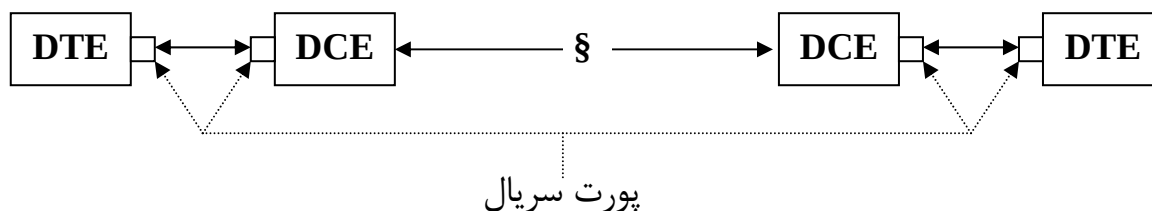
قبلاً آموختیم که اگر ولتاژی بین صفر تا $0/8$ ولت باشد از آن به «صفر منطقی» و اگر بین ۳ تا ۵ ولت باشد از آن به «یک منطقی» تعبیر می‌شود.^۱ چون RS232 قبل از TTL استاندارد شد، سطوح ولتاژ منطقی آن با TTL متفاوت است. در RS232 (که پورت سریال کامپیوتر نمونه‌ای از کاربرد آن است)، ولتاژی بین ۳- ولت تا ۲۵- ولت به عنوان «یک» و ولتاژی بین ۳+ ولت تا ۲۵+ ولت به عنوان «صفر» منطقی در نظر گرفته می‌شود.

توجه کنید چون معمولاً تراشه‌هایی با استاندارد TTL به پورت سریال کامپیوتر متصل می‌شود (مانند تراشه‌های مدیریت ارتباط سریال یا میکروکنترلرها)، به تراشه‌ای نیاز داریم که ولتاژهای RS232 را به TTL و بالعکس تبدیل کند. این تراشه‌ها Line Driver نامیده می‌شوند که MAX232 و MAX233 نمونه‌ای از آن هستند.

یکی از مزایای پورت سریال این است که برای دستگاه‌های کم‌مصرف که به آن متصل شوند، می‌تواند تغذیه مورد نیاز را تأمین کند. جریان خروجی پورت سریال کامپیوتر بین ۱۰ تا ۲۰ میلی آمپر و مقاومت ورودی ۱۰ کیلو اهم است.

سیگنالهای RS232

چون پورت سریال از ابتدا برای اتصال مودم خارجی به کامپیوتر استاندارد شد، سیگنالهای این پورت نیز با نامهایی که در ارتباط کامپیوتر و مودم به کار می‌روند نامگذاری شده‌اند. برای فهم این سیگنالها ابتدا به نحوه برقراری ارتباط بین دو کامپیوتر به کمک مودم خارجی می‌پردازیم. شکل زیر این ارتباط را نشان می‌دهد.



منظور از DTE^۲ هر دستگاهی است که بخواهد اطلاعاتی ارسال یا دریافت کند. کامپیوتر یا میکروکنترلر مثالی از DTE هستند.

DCE^۳ دستگاهی است که وظیفه آن دریافت داده‌ها از DTE و ارسال آن روی خط داده یا دریافت داده‌ها از روی خط انتقال ارائه آن به DTE است. مودم مثال خوبی از DCE است. برای ارسال داده‌ها از DTE اولی به DTE دوم روند زیر دنبال می‌شود:

^۱ این منطق ولتاژ موسوم به TTL است. در منطق‌های ولتاژی دیگر مانند CMOS، ولتاژهای دیگری به «صفر» و «یک» منطقی نسبت داده می‌شوند.

^۲ Data Terminal Equipment

^۳ Data Communication Equipment

- ارسال داده‌ها از DTE اول به DCE اول
- ارسال داده‌ها از DCE اول به DCE دوم
- ارسال داده‌ها از DCE دوم به DTE دوم

اکنون ببینیم پورت سریال چگونه به عنوان رابط بین DTE و DCE عمل می‌کند. جدول زیر سیگنالهای پورت سریال کامپیوتر را نشان می‌دهد:

شماره پایه	عملکرد	جهت سیگنال (از دید کامپیوتر)	نوع سیگنال
۱	Data Carrier Detect (DCD)	ورودی	کنترل
۲	Received Data (RxD)	ورودی	داده
۳	Transmitted Data (TxD)	خروجی	داده
۴	Data Terminal Ready (DTR)	خروجی	کنترل
۵	Ground (Gnd)	-	-
۶	Data Set Ready (DSR)	ورودی	کنترل
۷	Request To Send (RTS)	خروجی	کنترل
۸	Clear To Send (CTS)	ورودی	کنترل
۹	Ring Indicator (RI)	ورودی	کنترل

سیگنالهای فوق را می‌توان به سه دسته کلی تقسیم کرد.

الف) سیگنالهای داده

پایه RxD

کامپیوتر از طریق این پایه، اطلاعات را به صورت سریال از مودم دریافت می‌کند.

پایه TxD

کامپیوتر از طریق این پایه، اطلاعات را به صورت سریال برای مودم ارسال می‌کند.

پایه Gnd

زمین پورت سریال است که باید با زمین DCE مشترک شود.

ب) سیگنالهای اعلام آمادگی ارتباط

پایه DCD

وقتی مودم با مودم طرف مقابل ارتباط برقرار می‌کند و سیگنال حامل^۱ معتبر روی خط کشف می‌کند، از طریق این پایه مراتب را به کامپیوتر سمت خودش اطلاع می‌دهد.

¹ Carrier

پایه DTR

وقتی کامپیوتر آماده ارتباط با مودم است این پایه را فعال می‌کند.

پایه DSR

وقتی مودم آماده ارتباط با کامپیوتر است این پایه را فعال می‌کند.

پایه RI

وقتی مودم سیگنال زنگ (Ring) را روی خط ارتباطی با مودم دیگر دریافت می‌کند، از طریق این پایه به کامپیوتر اطلاع می‌دهد تا به خط جواب بدهد.

ج) سیگنالهای دست‌دهی برای کنترل جریان داده**پایه RTS**

هنگامی که کامپیوتر می‌خواهد برای مودم یک بایت بفرستد، با فعال کردن این پایه از مودم درخواست اجازه می‌کند.

پایه CTS

وقتی مودم سیگنال RTS را از کامپیوتر دریافت کرد، در صورت آمادگی برای دریافت داده، سیگنال CTS را برای کامپیوتر ارسال می‌کند.

پرسش) روش کنترل جریان داده به روش سخت‌افزاری (Hardware Flow Control) به کمک سیگنالهای RTS و CTS را توضیح دهید.

توجه کنید در توضیحات بالا کامپیوتر مثالی از DTE و مودم مثالی از DCE است. اصول ارتباط هر DTE با هر DCE طبق استاندارد RS232 مانند آنچه گفته شد می‌باشد.

به صورت خلاصه می‌توان گفت برای ارتباط بین دو DTE به کمک DCE:

- به کمک سیگنالهای DTR و DSR و DCD و RI ارتباط برقرار می‌شود.
- به کمک سیگنالهای RxD و TxD داده‌ها رد و بدل می‌شوند.
- به کمک سیگنالهای RTS و CTS کنترل جریان داده بین DTE و DCE انجام می‌شود.

مودم پوچ (Null Modem)

در بخش قبل دیدیم چگونه می‌توان دو DTE را به کمک دو DCE به هم متصل کرد. حال فرض کنید می‌خواهیم دو DTE (مثلاً دو کامپیوتر، دو میکروکنترلر یا یک کامپیوتر و یک میکروکنترلر) را به یکدیگر متصل کنیم، اما DCE (مثلاً مودم) در اختیار نداریم. در این

حالت، باید با توجه به عملکرد سیگنالهای RS232، طوری بین پورتهای سریال دو DTE سیمبندی کنیم که تصور کنند از طریق DCE بهم متصل هستند، در حالی که عملاً DCE بین آنها وجود ندارد. به این اتصال، مودم پوچ گفته می‌شود.

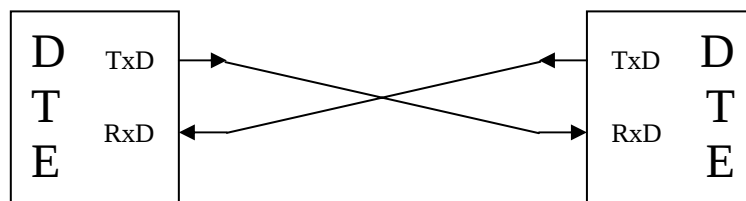
سیمبندی بین پورتهای سریال برای شبیه‌سازی وجود DCE چگونه باید باشد؟

به عنوان مثال، DTE اول می‌خواهد برای DTE دوم اطلاعات ارسال کند. اگر DCE وجود داشته باشد:

- DTE اول از طریق خط TxD خودش اطلاعات را برای DCE خودش ارسال می‌کند
- DCE اطلاعات را برای DCE مقابل ارسال می‌کند.
- DCE مقابل اطلاعات را برای DTE دوم می‌فرستد که از طریق سیگنال RxD پورت سریال DTE دریافت می‌شود.

حال که DCE وجود ندارد، بدیهی است که باید خط RxD هر DTE به خط TxD در DTE مقابل و بالعکس متصل شود.

شکل زیر ساده‌ترین نوع مودم پوچ را نشان می‌دهد:

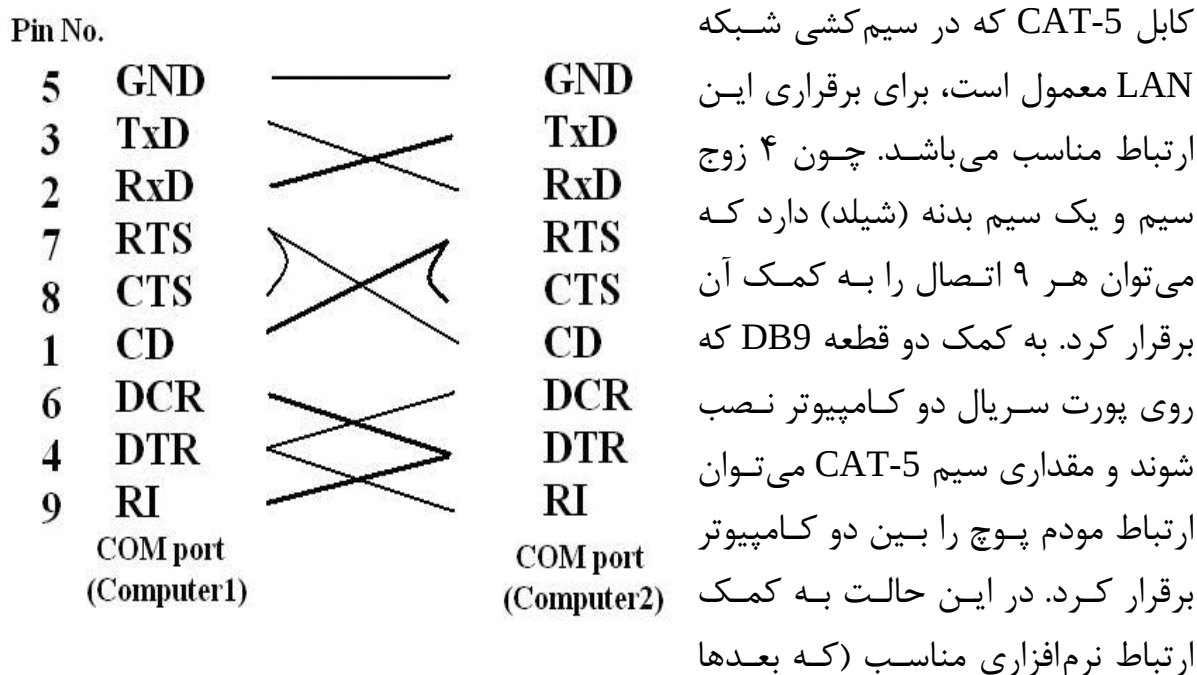


بدیهی است زمین هر دو DTE باید مشترک باشد؛ به همین لحاظ سیگنال Gnd دو پورت سریال به هم متصل می‌شود.

تکلیف سیگنالهای دست‌دهی چیست؟

گفتیم هنگامی که DTE می‌خواهد برای DCE یک بایت بفرستد، با فعال کردن این پایه از وی درخواست اجازه می‌کند. وقتی DCE سیگنال RTS را از DTE دریافت کرد، در صورت آمادگی برای دریافت داده، سیگنال CTS را برای DTE ارسال می‌کند. حال که DCE وجود ندارد که DTE بخواهد از آن اجازه بگیرد، یعنی این اجازه باید همیشه داده شود، به همین لحاظ در ساختار مودم پوچ، RTS هر DTE به CTS آن متصل می‌شود. یعنی هرگاه DTE برای اجازه گرفتن از DCE سیگنال RTS را فعال کند، بلافاصله سیگنال CTS خودش فعال می‌شود و DTE تصور می‌کند که اجازه از سوی DCE صادر شده است.

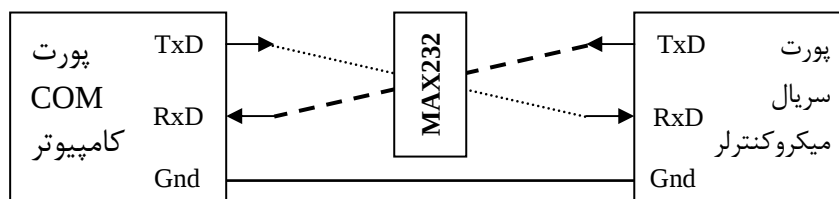
شکل مقابل ساختار کامل مودم پوچ برای ارتباط بین پورت سریال دو کامپیوتر را نشان می‌دهد.



پرسش) با بررسی ساختار کامل مودم پوچ، دلیل سیم‌بندی‌های ذکر شده را توضیح دهید.

اتصال میکروکنترلر به کامپیوتر

اکثر میکروکنترلرها دارای کنترل‌کننده ارتباط سریال غیرهمگام (UART) هستند. برای ارتباط بین میکروکنترلر و کامپیوتر می‌توان از ساختار مودم پوچ به صورت زیر استفاده کرد:



شرح برنامه میکروکنترلر برای ارتباط سریال در حوصله بحث ما نیست. تنها به یاد داشته باشید که مشخصات ارتباط غیرهمگام (نرخ داده، تعداد بیت‌های داده، نوع توازن به کار رفته، تعداد بیت‌های توقف، روش کنترل جریان داده) در کامپیوتر و میکروکنترلر باید یکسان باشد.

نرم‌افزار کامپیوتر برای ارتباط با پورت سریال

اکنون اتصال سخت‌افزاری بین کامپیوتر و DTE دیگری (کامپیوتر دیگر یا میکروکنترلر یا ...) برقرار شده است. می‌خواهیم در کامپیوتر برنامه‌ای بنویسیم که ارتباط سریال را با DTE مقابل برقرار کند؛ یعنی اطلاعات را به صورت سریال ارسال و دریافت کند.

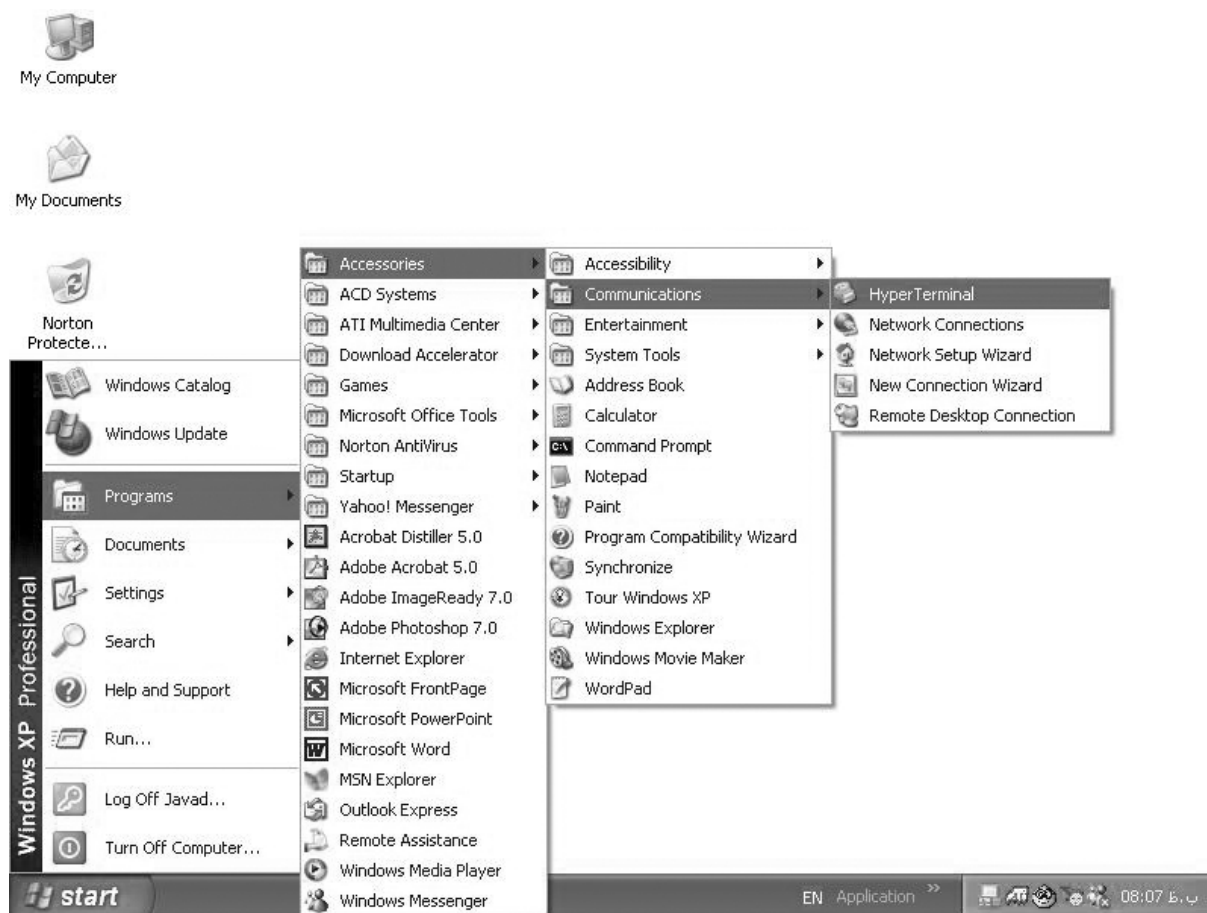
برای برقراری ارتباط نرم‌افزاری در ابتدا باید ارتباط سریال پیکربندی شود؛ یعنی نرخ داده، تعداد بیت‌های داده، نوع توازن به کار رفته، تعداد بیت‌های توقف و روش کنترل جریان داده مشخص شود. این مشخصات عیناً در نرم‌افزار DTE طرف مقابل برای پیکربندی پورت سریال آن نیز باید به کار برود.

نرم‌افزار ارتباط سریال در کامپیوتر می‌تواند به کمک زبانهای سطح بالا مانند C یا Delphi یا Visual C++ یا ... نوشته شود که در ادامه مثالهایی در DOS و ویندوز خواهیم دید. اما راه ساده‌تری نیز وجود دارد؛ ویندوز دارای نرم‌افزاری به نام HyperTerminal است که می‌تواند ارتباطهای مختلف از جمله ارتباط سریال را به راحتی برقرار کند. ابتدا با این نرم‌افزار آشنا می‌شویم.

نرم‌افزار HyperTerminal

این نرم‌افزار در نسخه‌های مختلف ویندوز وجود دارد. برای استفاده از آن (در ویندوز XP) به ترتیب زیر عمل کنید:

- از منوی Start ویندوز به مسیر Programs/Accessories/Communications بروید و گزینه HyperTerminal را انتخاب کنید.



- مرحله بعد شکلی مانند زیر دارد که در آن باید نام ارتباط را انتخاب کنید. نام My Connection در شکل زیر برای ارتباط انتخاب شده است.

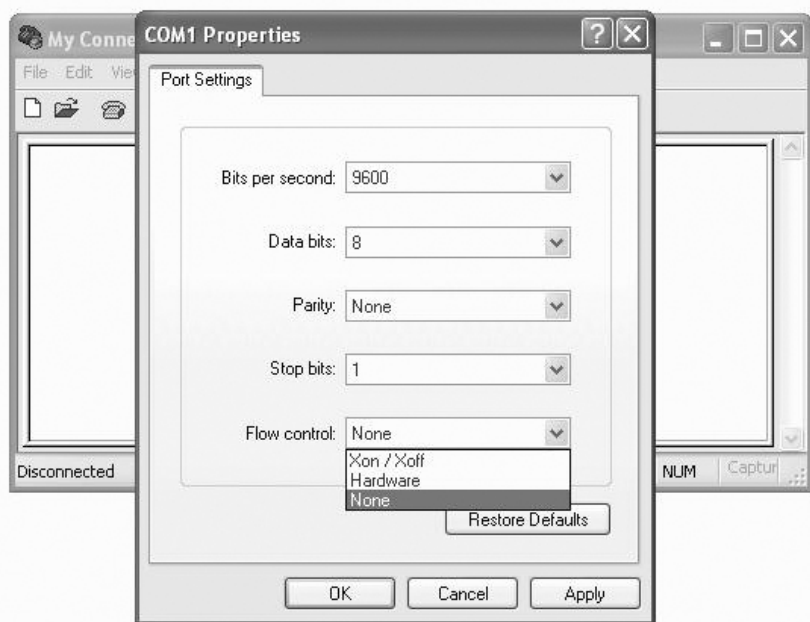


- در مرحله بعد باید وسیله ارتباطی را انتخاب کنید. این وسیله ارتباطی می‌تواند کارت شبکه، کارت مودم یا پورتهای سریال باشد. پورت COM1 را انتخاب کنید.



- توجه کنید که کامپیوتر دارای یک یا دو پورت سریال سخت‌افزاری (COM1 و احتمالاً COM2) است که برای اتصال تجهیزات خارجی به کار می‌روند. به علاوه تعدادی پورت سریال داخلی (COM3، COM4 و ...) نیز وجود دارد که برای اتصال کارت مودمهای داخلی استفاده می‌شوند. در بعضی متون به این پورتهای سریال، پورتهای مجازی^۱ گفته می‌شود.
- در مرحله بعد باید ویژگیهای ارتباط سریال را تنظیم کنید:

¹ Virtual



در ارتباط بالا نرخ داده برابر ۹۶۰۰ بیت در ثانیه، تعداد بیت‌های داده برابر ۸ بیت، بدون بیت توازن، با یک بیت توقف انتخاب شده است. توجه کنید که پورت سریال کامپیوتر از نرخ‌های ارتباط داده محدودی مانند ۱۱۰، ۳۰۰، ۶۰۰، ۱۲۰۰، ۲۴۰۰، ۴۸۰۰، ۹۶۰۰ و ۱۹۲۰۰ بیت در ثانیه حمایت می‌کند و دستگاه متصل به این پورت باید نرخ داده مورد نظرش را این لیست انتخاب کند.^۱

برای کنترل جریان داده می‌توان از روش‌های سخت‌افزاری یا نرم‌افزاری (Xon/Xoff) استفاده کرد. اگر DTE مقابل از یکی از این روش‌ها حمایت کند (مانند میکروکنترلر AWC86 که از روش سخت‌افزاری یا RTS/CTS حمایت می‌کند)، می‌توان روش مناسب را برگزید. اما برای ارتباطات ساده، لازم به استفاده از کنترل جریان داده نیست و باید گزینه None را در قسمت Flow Control برگزید.

پس از این مراحل، وارد محیط HyperTerminal می‌شوید. از این پس هرچه در این محیط تایپ کنید، با مشخصات تنظیم‌شده از پورت سریال کامپیوتر به بیرون فرستاده می‌شود. توجه کنید آنچه از پورت بیرون فرستاده می‌شود، کد اسکی کاراکترهایی است که تایپ می‌کنید. به علاوه آنچه تایپ می‌کنید در محیط HyperTerminal نشان داده نمی‌شود.

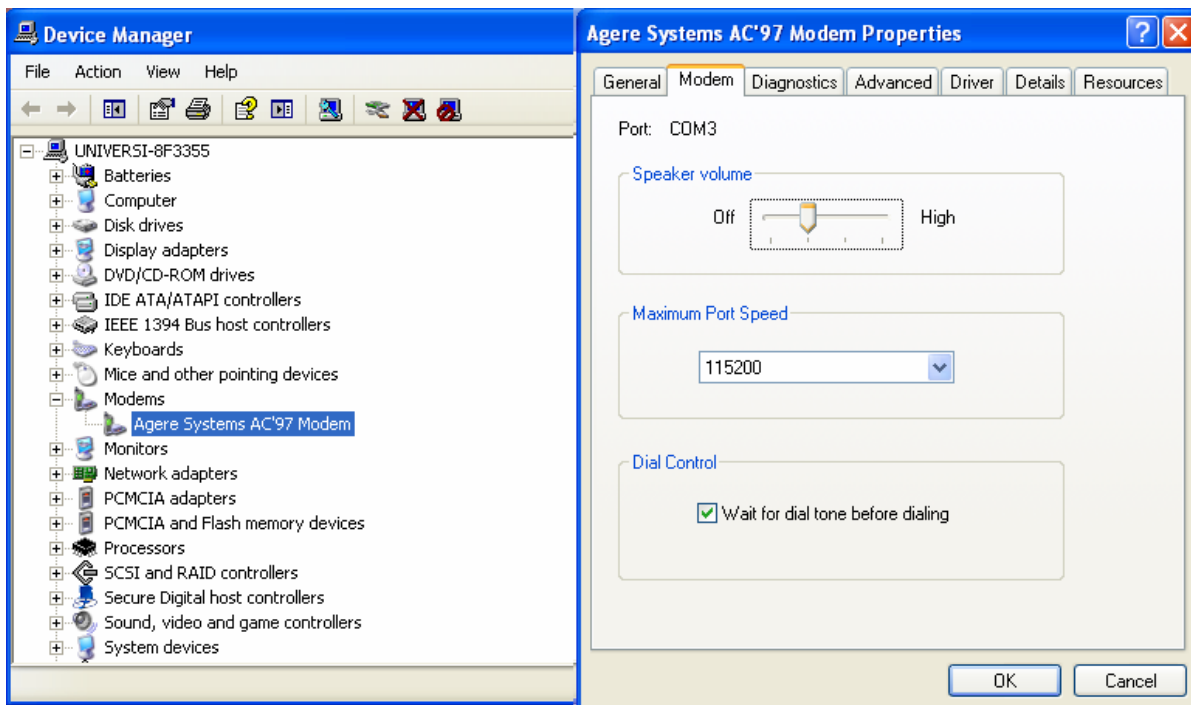
از سوی دیگر، هر اطلاعاتی به صورت سریال با مشخصات تنظیم‌شده به پورت سریال کامپیوتر برسد، در محیط HyperTerminal نمایش داده می‌شود. نحوه نمایش این اطلاعات در HyperTerminal قابل تنظیم است. در حالت پیش فرض، هر عددی دریافت شود، کاراکتری که کد اسکی آن برابر آن عدد است در محیط HyperTerminal نشان داده می‌شود.

^۱ دلیل استفاده از کریستال بسیار دقیق 11.0592MHz برای میکروکنترلر ۸۰۵۱ هنگامی که می‌خواهد با پورت سریال کامپیوتر ارتباط برقرار کند همین موضوع است.

اگر پورت COM دو کامپیوتر را به کمک کابل مودم پوچ (که ساختار آن پیشتر گفته شد) به هم وصل کنید، هرچه در محیط HyperTerminal یک کامپیوتر تایپ کنید در محیط HyperTerminal کامپیوتر دیگر دیده می‌شود. به علاوه به کمک منوی Transfer می‌توانید بین دو کامپیوتر فایل نیز رد و بدل کنید.

اتصال مودم

همانگونه که گفته شد، کامپیوتر دارای تعدادی پورت COM داخلی (COM3 و COM4 و ...) است که برای اتصال مجازی کارتهای مودم داخلی به پورت سریال استفاده می‌شوند. با انتخاب گزینه Modem در Device Manager می‌توانید ببینید مودم کامپیوتر شما به کدام پورت COM مربوط شده است؟



ارتباط دو کامپیوتر به کمک خط تلفن و HyperTerminal

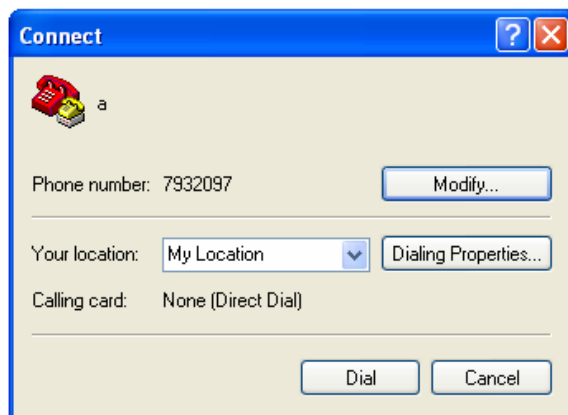
حتماً برای شما پیش آمده که نیاز به فایلی داشته باشید که دوستان باید آن را از روی کامپیوترش برای شما بفرستد.

ساده‌ترین راه این مسأله چیست؟ خواهید گفت اینترنت! حق با شماست. اما اگر شما یا دوستان به اینترنت دسترسی نداشته باشید چه باید کرد؟ نگران نباشید! به کمک HyperTerminal و مودم، می‌توانید با استفاده از خطوط تلفن اطلاعات را با کامپیوتر دوستان مبادله کنید.

برای این کار، یکی از دو کامپیوتر باید به عنوان تماس‌گیرنده و دیگری به عنوان پذیرنده تماس مشخص شود. فرض کنید شما تماس‌گیرنده باشید. برای این کار مانند آنچه قبلاً گفته

شد، وارد محیط HyperTerminal شوید، نامی برای ارتباط انتخاب کنید و به مرحله بعد بروید.

در این مرحله، از ابزارهای مختلف ارتباطی، مودم را انتخاب کنید و در کادرهای بالا، شماره و کد تلفن منزل دوستان را وارد کنید. با انتخاب گزینه OK به مرحله بعد بروید:



اگر اکنون گزینه Dial را انتخاب کنید، کامپیوتر شما به کمک مودم با شماره تلفنی که مشخص کرده‌اید (در مثال بالا ۷۹۳۲۰۹۷) تماس می‌گیرد و می‌خواهد با مودم مقابل تماس ارتباطی برقرار کند.

قبل از این کار، باید دوست شما در کامپیوتر منزل خودش همین مراحل را طی کند و با رسیدن به مرحله بالا، با انتخاب گزینه Cancel وارد محیط HyperTerminal شود. در ادامه، دوست شما باید از منوی Call گزینه Wait for Call را انتخاب کند تا مودم کامپیوتر منتظر تماس تلفنی شما بماند.

اکنون شما باید گزینه Dial را در HyperTerminal کامپیوتر خودتان انتخاب کنید. وقتی مودم کامپیوتر شما با تلفن منزل دوستان تماس می‌گیرد، زنگ تلفن منزل او به صدا درمی‌آید. اگر کسی در منزل او گوشی تلفن را بردارد، جز سر و صدای گوشخراش مودم کامپیوتر شما چیزی نمی‌شنود! اگر کسی به تلفن جواب ندهد، مودم کامپیوتر دوستان که منتظر تماس از سوی مودم کامپیوتر شماست گوشی را برمی‌دارد. در این حالت همان سر و صداهایی که هنگام اتصال به اینترنت از مودم کامپیوترتان می‌شنوید به گوش می‌رسد و سپس ارتباط بین دو کامپیوتر به کمک خط تلفن و مودمها برقرار می‌شود.

از این به بعد هرچه شما در HyperTerminal کامپیوتر خودتان تایپ کنید در محیط HyperTerminal کامپیوتر دوستان دیده می‌شود و عکس این موضوع نیز برقرار است. در ضمن با انتخاب گزینه Transfer می‌توانید برای یکدیگر فایل ارسال کنید.

استفاده از پورت سریال در زبانهای سطح بالا

نرم افزار HeyperTerminal تنها برای برقرار کردن ارتباطهای داده‌ای ساده به کمک ابزارهای ارتباطی کامپیوتر به کار می‌رود. چنانچه بخواهید یک ارتباط هوشمند برقرار کنید (مثلاً نمودار تغییرات دمای اتاق در طول یک شبانه‌روز را روی کامپیوتر مشاهده کنید یا یک روبات را به کمک کامپیوتر کنترل کنید)، دیگر استفاده از HyperTerminal به کار نمی‌آید و باید به کمک برنامه‌ای که در کامپیوتر می‌نویسید این کار را انجام دهید.

اکثر زبانهای سطح بالا از ارتباط با پورت سریال حمایت می‌کنند. برای نوشتن برنامه ارتباط سریال در هر زبانی باید موارد زیر مشخص شود:

- چگونه پورت سریال باید پیکربندی شود؟ به بیان دیگر ویژگیهای ارتباط سریال مانند نرخ داده، تعداد بیت‌های داده، نوع توازن به کار رفته، تعداد بیت‌های توقف و روش کنترل جریان داده باید به چه صورتی مشخص می‌گردد؟
- ارسال داده به پورت سریال به چه صورت انجام می‌شود؟
- چگونه می‌توان از دریافت کاراکتر توسط پورت سریال مطمئن شد؟
- دریافت داده از پورت سریال به چه صورت انجام می‌شود؟

در ادامه به سؤالات بالا در زبان C پاسخ می‌دهیم. توجه داشته باشید که همانطور که قبلاً گفته شد، داستان ارتباط با پورتهای ورودی/خروجی کامپیوتر (از جمله پورت سریال) در DOS و ویندوز متفاوت است. آنچه در ادامه می‌آید، برقراری ارتباط در C تحت DOS را نشان می‌دهد. برای برقراری ارتباط سریال در زبانهای تحت ویندوز باید به شیوه‌ای متفاوت عمل کنید که در بخش برقرار ارتباط با پورها در ویندوز، به نحوه ارسال و دریافت داده‌ها در زبان Visual C++ خواهیم پرداخت.

ثباتهای پورت سریال

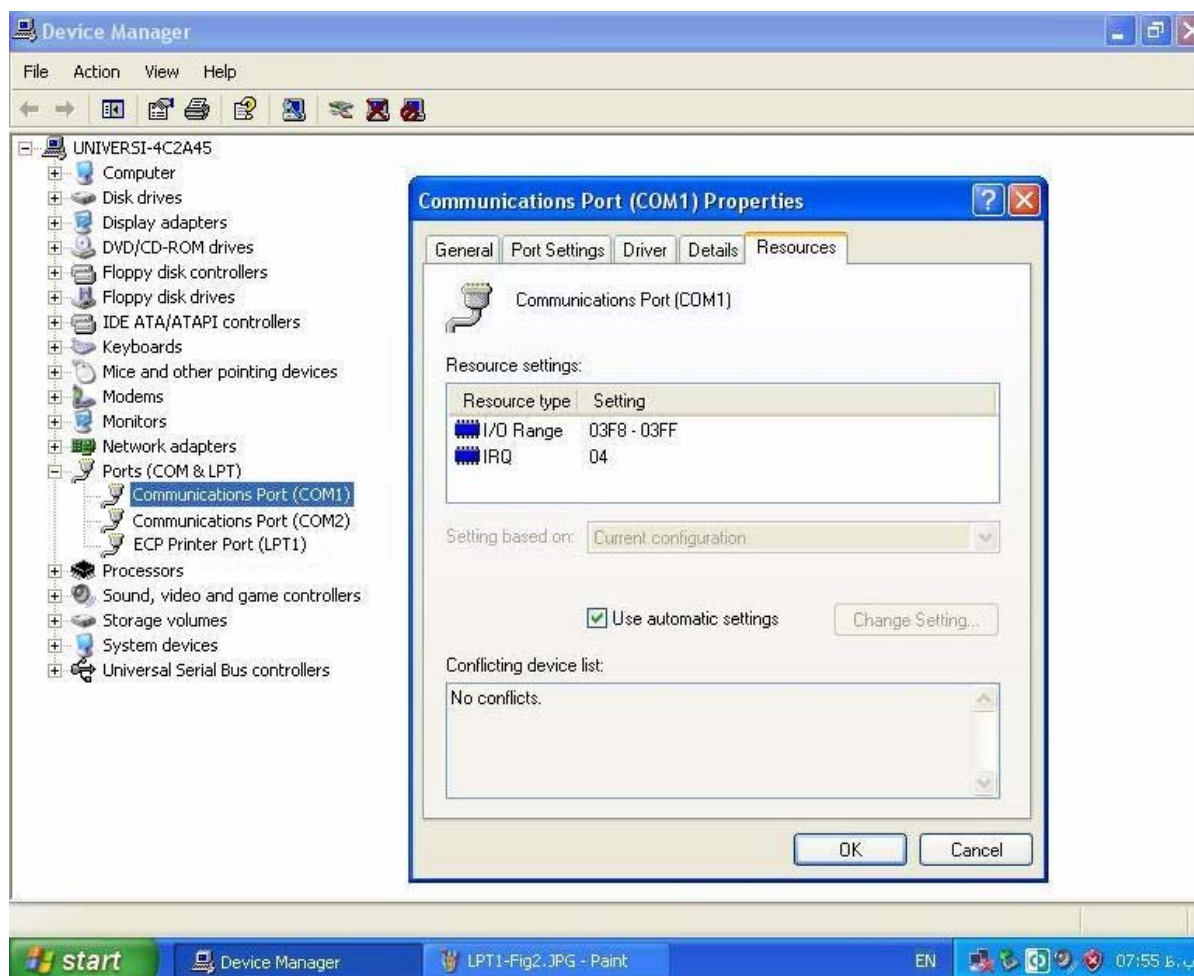
پورت COM	آدرس پورت
COM1	03F8-03FF
COM2	02F8-02FF
COM3	03E8-03EF
COM4	02E8-02EF

مانند پورت موازی که دارای یک مجموعه ثبات است، هر پورت سریال کامپیوتر نیز یک مجموعه‌ای متشکل از هشت ثبات برای مدیریت ارتباط سریال دارد. آدرس پایه این ثباتها برای پورتهای COM مختلف را در جدول مقابل می‌بینید:

در ویندوز می‌توانید پورتهای سریال سخت‌افزاری کامپیوتر خود و نیز آدرس‌های تخصیص داده شده به ثباتهای ویژه پورت را ببینید. برای این کار:

- از Control Panel گزینه System را انتخاب کنید.
- به سربرگ Hardware بروید و گزینه Device Manager را انتخاب کنید.

حال در قسمت Ports می‌توانید مشخصات پورتهای کامپیوتر خود را ببینید. شکل زیر، گزینه Ports مربوط به یک کامپیوتر نوعی را نشان می‌دهد.



همانطور که دیده می‌شود، هشت ثبات با شروع از 03F8 h به پورت COM1 اختصاص داده شده است.

پورت COM	آدرس پایه ورودی/خروجی ثباتهای پورت
COM1	0040:0000 – 0040:0001
COM2	0040:0002 – 0040:0003
COM3	0040:0004 – 0040:0005
COM4	0040:0006 – 0040:0007

به کمک BIOS نیز می‌توان آدرس مبنای ثباتهای ورودی/خروجی پورت سریال را دانست. در واقع این آدرسها از آفست 0000 به بعد سگمنت 0040 حافظه ذخیره می‌شود. جدول مقابل را ببینید.

مثلاً دستور `D 0040:0000 L8` در نرم‌افزار Debug، آدرس پایه پورتهای سریال نصب شده در کامپیوتر را نشان می‌دهد. قبلاً با نرم‌افزار Debug و فرمانهای آن آشنا شدیم.

به کمک برنامه‌نویسی C نیز می‌توان آدرسهای ذخیره شده در این بخش حافظه را مشاهده کرد. برنامه زیر، آدرس پورت COM1 و در صورت عدم نصب پورت مزبور، پیغامی را نمایش می‌دهد.

```
#include <STDIO.H>
#include <DOS.H>

void main(){
    unsigned int far *fptr;
    fptr = (unsigned int far *) 0x00000400;
    if (*fptr > 0)
        printf("COM1 I/O base address = %X",*fptr);
    else
        printf("COM1 not found");
}
```

جدول زیر ثباتهای پورت سریال و آدرس آنها نسبت به آدرس مبنا (Base) را نشان می‌دهد:

آدرس	عملکرد	خواندنی یا نوشتنی	نام	بیت DLAB
Base+0	بافر ارسال اطلاعات	نوشتنی	-	0
	بافر دریافت اطلاعات	خواندنی	-	0
	قسمت پایین نرخ داده	خواندنی/نوشتنی	-	1
Base+1	فعال کننده وقفه	خواندنی/نوشتنی	IER	0
	قسمت پایین نرخ داده	خواندنی/نوشتنی	-	1
Base+2	شناسایی وقفه	خواندنی	IIR	-
	کنترل کننده FIFO	نوشتنی	FCR	-
Base+3	کنترل کننده قالب اطلاعات	خواندنی/نوشتنی	LCR	-
Base+4	کنترل کننده مودم	خواندنی/نوشتنی	MCR	-
Base+5	نمایشگر وضعیت خط	خواندنی	LSR	-
Base+6	نمایشگر وضعیت مودم	خواندنی	MSR	-
Base+7	ثبات حافظه پنهان	خواندنی/نوشتنی	-	-

مثلاً اگر آدرس مبنای پورت سریال COM1 برابر 03F8 h باشد، در برنامه کامپیوتر برای ارسال داده به این پورت و نیز دریافت داده از آن باید از ثبات واقع در آدرس 03F8 h استفاده کنیم (به خط اول جدول توجه کنید). با نوشتن داده ۸ بیتی در این ثبات، تراشه UART آن را به صورت مناسب قالب‌بندی کرده و از پورت سریال به بیرون ارسال می‌کند. وقتی نیز بسته‌ای توسط پورت سریال دریافت می‌شود، تراشه UART داده را از دل آن بسته بیرون

کشیده و در ثبات فوق قرار می‌دهد. توجه داشته باشید که داده‌های ارسالی و دریافتی در بافرهای جداگانه‌ای از نوع FIFO^۱ ذخیره می‌شوند. با سایر ثباتها در ادامه آشنا می‌شویم.

پیکربندی ارتباط سریال

گفتیم پیش از آغاز ارتباط، ویژگیهای ارتباط سریال مانند نرخ داده، تعداد بیت‌های داده، نوع توازن به کار رفته و تعداد بیت‌های توقف باید مشخص شود. برای مشخص کردن نرخ داده باید عدد ویژه‌ای (وابسته به نرخ داده) را در ثباتهای اول و دوم قرار بدهید. در این حالت باید بیت DLAB (بیت شماره ۷ ثبات کنترل کننده قالب اطلاعات) یک باشد. برای تعیین ویژگیهای دیگر ارتباط سریال باید بیت‌های ثبات کنترل کننده قالب اطلاعات مقداردهی شوند:

DLAB	Break	Parity	Parity	Parity	Stop	Data	Data
		000: بدون توازن 001: توازن فرد 011: توازن زوج 101: توازن همیشه «یک» 111: توازن همیشه «صفر»			0: یک بیت توقف 1: دو بیت توقف		00: ۵ بیت داده 01: ۶ بیت داده 10: ۷ بیت داده 11: ۸ بیت داده

با «یک» شدن بیت قطع (Break)، فرستنده ارسال اطلاعات را قطع می‌کند.

پیکربندی پورت سریال در C (اسمبلی)

در سیستم عامل DOS، وقفه 14H برای ارتباط با پورت سریال تعریف شده است که شامل توابعی برای پیکربندی پورت سریال و ارسال و دریافت داده با این پورت است. برای پیکربندی پورت سریال در زبانهای تحت DOS مانند اسمبلی یا C تحت DOS، از تابع شماره ۴ این وقفه استفاده می‌کنیم. با مقداردهی ثباتهای مختلف پردازنده و اجرای این تابع، می‌توان بدون نیاز به درگیر شدن با ثباتهای پورت سریال به پیکربندی دلخواه برای ارتباط سریال دست یافت.

جزئیات این تابع در ذیل آمده است:

AH = 04

AL = 00H (بدون توقف)، 01H (توقف)

DX = (COM1 = 0, COM2 = 1,...) شماره پورت

¹ First In First Out - مانند صف

BH = 00H (بدون توازن)

01H (توازن فرد)

02H (توازن زوج)

03H (توازن فرد پایدار)

04H (توازن زوج پایدار)

BL = 00H (۱ بیت توقف)

01H (۱/۵ بیت توقف برای کلمه ۵ بیتی)

02H (۲ بیت توقف برای کلمه های بزرگتر از ۵ بیت)

CH = 00H (کلمه ۵ بیتی)

01H (کلمه ۶ بیتی)

02H (کلمه ۷ بیتی)

03H (کلمه ۸ بیتی)

CL = 00H (110 bps)

01 (150 bps)

02 (300 bps)

03 (600 bps)

04 (1200 bps)

05 (2400 bps)

06 (4800 bps)

07 (9600 bps)

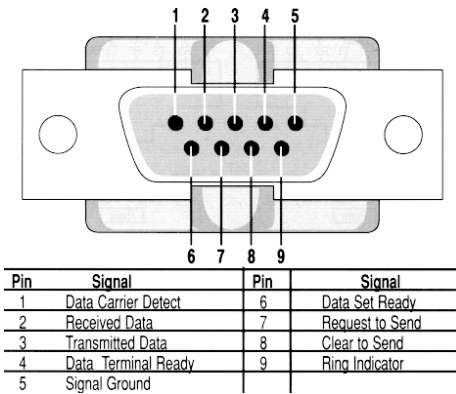
08 (19200 bps)

```
void InitialSerialPort(void) {
    asm {
        mov     ah, 04h
        mov     al, 00h
        mov     bh, 00h    // No Parity
        mov     bl, 00h    // 1 Stop Bit
        mov     ch, 03h    // 8 bit data
        mov     cl, 07h    // 9600 bps
        mov     dx, 00h    // port COM1
        int     14h
    }
}
```

تابع مقابل در زبان C پورت
سریال را با نرخ داده ۹۶۰۰
بیت در ثانیه، کلمه ۸ بیتی،
یک بیت توقف و بدون استفاده
از بیت توازن پیکربندی
می کند.

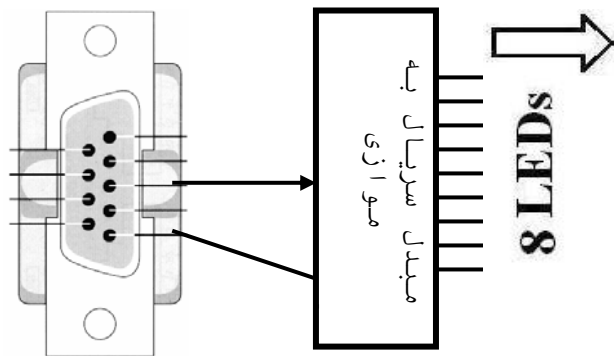
ارسال داده به پورت سریال

برای ارسال داده به پورت سریال، مشابه با آنچه قبلاً دیدیم از تابع `outportb` استفاده می‌شود. البته تابع شماره ۱ وقفه 14h نیز برای این منظور می‌تواند به کار رود. در ادامه یک مثال عملی را بررسی می‌کنیم.



مثال ارسال داده‌ها به کمک پورت سریال

پایه‌های پورت سریال را قبلاً دیدیم. شکل روبرو خلاصه‌ای از آنچه گفتیم و باید در مثالهای طراحی سیستم به کار ببریم را نشان می‌دهد.



```
#include <STDIO.H>
#include <CONIO.H>
#include <STDLIB.H>

void InitialSerialPort(void) {
    asm {
        mov     ah,04h
        mov     al,00h
        mov     bh,00h
        mov     bl,00h
        mov     ch,03h
        mov     cl,07h    // 9600 bps
        mov     dx,00h
        int     14h
    }
}

void main(void) {
    unsigned char data = 0;

    InitialSerialPort();

    while (!kbhit()) {

        outportb(0x3F8,data);
        delay(500);

        if (data == 255)
            data = 0;
        else
            data++;
    }
}
```

فرض کنید می‌خواهیم سیستمی طراحی کنیم که اعداد صفر تا ۲۵۵ را روی هشت عدد LED نمایش دهد. برای این کار، از پورت سریال کامپیوتر به صورت مقابل استفاده می‌کنیم.

همانطور که می‌بینید، پایه `TxD` پورت سریال کامپیوتر به پایه `RxD` مبدل سریال به موازی (مثلاً یک میکروکنترلر) متصل شده است. این مبدل، اطلاعات را به صورت سریال از کامپیوتر دریافت و پس از تبدیل به اطلاعات موازی روی هشت LED نمایش می‌دهد. برنامه را در شکل مقابل می‌بینید.

این برنامه، اعداد صفر (00000000) تا ۲۵۵ (11111111) را با تأخیر ۵۰۰ میلی‌ثانیه به پورت سریال کامپیوتر ارسال می‌کند. پس از رسیدن

متغیر خروجی به ۲۵۵، مجدداً مقدار آن صفر می‌شود. برنامه با فشار یک کلید به پایان می‌رسد.

پرسش) برنامه را به گونه‌ای اصلاح کنید که LEDهای زوج و فرد را به ترتیب و با تأخیر یک ثانیه روشن و خاموش کند و یک رقص نور ایجاد کند. برنامه باید با فشار یک کلید، همه LEDها را خاموش کرده و به پایان برسد.

دریافت داده از پورت سریال

فرض کنید دستگاهی به پورت سریال کامپیوتر متصل است که اطلاعات را به صورت سریال برای کامپیوتر ارسال می‌کند. می‌خواهیم در کامپیوتر برنامه‌ای بنویسیم که این اطلاعات را دریافت کند.

به نظر می‌رسد دستور `inportb` کافی باشد:

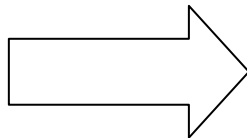
```
data = inportb(0x3F8);
```

این دستور غلط نیست. یعنی اطلاعات را از پورت COM1 (به آدرس 03F8 h) می‌خواند و در متغیر `data` ذخیره می‌کند. اما به کار بردن این دستور به تنهایی صحیح نیست. بلکه: باید ابتدا/از دریافت داده توسط پورت سریال مطمئن شوید و سپس داده فوق را بخوانید.

تابع ۳ وقفه 14 h برای خواندن وضعیت پورت به کار می‌رود:

AH = 03

DX = شماره پورت



بایت وضعیت پورت AH =

بایت وضعیت مودم AL =

بعد از اجرای این تابع، بایت وضعیت پورت سریال در ثبات AH قرار می‌گیرد که بیت‌های آن به شرح زیر هستند:

0 1 2 3 4 5 6 7

شماره بیت

۱

وقت تمام

۱

ثبات جابجایی ارسال

۱

ثبات نگهداری ارسال خالی

۱

توقف شناسایی شد

۱

خطای قاب شناسایی شد

۱

خطای توازن شناسایی شد

۱

خطای عقب ماندن

۱

داده رسیده آماده

به بیت شماره ۷ دقت کنید. وقتی تابع ۳ وقفه 14 h اجرا شود، با بررسی بیت شماره ۷ ثبات

```
int RecvData() {
    unsigned char status;

    asm {
        mov     ah, 03h
        mov     dx, 00h
        int     14h
        mov     status, ah
    }

    status = status & 128;
    if (status == 0)
        return 0;
    else
        return 1;
}
```

AH می‌توان دانست آیا داده‌ای دریافت شده است یا نه؟ اگر پورت سریال داده‌ای دریافت کرده باشد، این بیت «یک» و در غیر این صورت «صفر» خواهد بود.

تابع مقابل بیت دریافت داده را بررسی می‌کند و در صورت آماده بودن داده ارزش منطقی صحیح («یک») و در غیر این صورت ارزش منطقی غلط («صفر») را باز می‌گرداند.

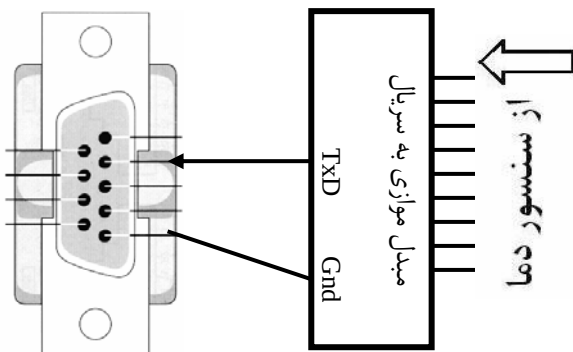
این تابع، بایت وضعیت صفحه کلید که در ثبات AH قرار دارد را در متغیر status کپی می‌کند. سپس این متغیر را با عدد

۱۲۸ (1000000) AND می‌کند. با این کار، چنانچه بیت شماره ۷ متغیر فوق «صفر» باشد، حاصل AND صفر بوده و مقدار «صفر» (ارزش منطقی نادرست) به عنوان خروجی تابع برگردانده می‌شود. در غیر این صورت حاصل AND غیر صفر است و مقدار «یک» (ارزش منطقی درست) به عنوان خروجی به تابع برگردانده می‌شود. بنابراین اگر خروجی تابع RecvData درست باشد یعنی داده‌ای دریافت شده و آماده خوانده شدن است.

اکنون به کمک تابع inportb می‌توانید داده را بخوانید. تابع شماره ۲ وقفه 14h نیز برای این منظور قابل استفاده است.

در ادامه یک مثال عملی را بررسی می‌کنیم.

مثال دریافت داده‌ها به کمک پورت سریال



می‌خواهیم سیستمی طراحی کنیم که دمای اتاق را هر یک ثانیه یک بار خوانده و روی نمایشگر کامپیوتر نمایش دهد. برای این کار، باید از سنسور دما استفاده شود. خروجی سنسور دما که یک کمیت آنالوگ است به کمک تراشه ADC به کمیتی دیجیتال تبدیل

می‌شود. مثلاً با استفاده از یک ADC^۱ هشت بیتی (مانند تراشه ADC804)، خروجی آنالوگ سنسور دما به یک عدد دیجیتالی ۸ بیتی تبدیل می‌شود.^۲ اکنون برای ارسال این عدد هشت بیتی (که در خروجی تراشه ADC به صورت موازی قرار دارد) به پورت سریال کامپیوتر، به یک مبدل سریال به موازی (مثلاً میکروکنترلر) نیاز داریم. همانطور که در شکل بالا می‌بینید، پایه TxD (ارسال) مبدل موازی به سریال به پایه RxD (دریافت) پورت سریال کامپیوتر متصل و زمینهای دو طرف مشترک شده است. اکنون باید در کامپیوتر برنامه‌ای بنویسیم که اطلاعات را از پورت سریال دریافت کرده و نمایش دهد. برنامه زیر را ببینید.

<pre>#include <STDIO.H> #include <CONIO.H> #include <STDLIB.H> void InitialSerialPort(void) { asm { mov ah,04h mov al,00h mov bh,00h mov bl,00h mov ch,03h mov cl,07h // 9600 bps mov dx,00h int 14h } } //////////////////////////////////// int RecvData() { unsigned char status; asm { mov ah,03h mov dx,00h int 14h mov status,ah } status = status & 128; if (status == 0) return 0; else return 1; }</pre>	<pre>void main(void) { unsigned char data; InitialSerialPort(); while (1) { while (RecvData() == 0 && !kbhit()); if (kbhit()) exit(0); data = inportb(0x3F8); printf("%X\n", data); } }</pre>
--	--

به این خط برنامه توجه کنید:

`while (RecvData() == 0 && !kbhit());`
 تا زمانی که کاراکتری دریافت نشده (خروجی تابع RecvData صفر است) و نیز کاربر کلیدی را فشار نداده باشد، برنامه در همین نقطه متوقف می‌ماند. با نقض شدن یکی از این دو شرط،

^۱ Analog to Digital Convertor

^۲ بعضی از میکروکنترلرها خود دارای ADC داخلی هستند.

برنامه از این خط عبور کرده و ادامه می‌دهد. در خط بعدی، بررسی می‌شود که آیا در اثر فشردن کلیدی از خط قبلی عبور کرده‌ایم؟ اگر اینگونه باشد، برنامه به پایان می‌رسد. در غیر این صورت، به کمک تابع `inportb` داده دریافتی خوانده شده و در دستگاه شانزده‌تایی روی نمایشگر کامپیوتر نشان داده می‌شود. به صورت خلاصه، برنامه بالا تا زمانی که کاربر کلیدی را فشار نداده باشد، اطلاعات دریافتی از پورت سریال را خوانده و روی نمایشگر کامپیوتر نشان می‌دهد.

پرسش) برنامه بالا را به گونه‌ای ارسال کنید که قبل از خروج، تعداد بایتهای دریافتی را نمایش دهد.

در زبان C، تابع `bioscom` که در فایل سرآیند `bios.h` قرار دارد، برای پیکربندی ارتباط سریال، ارسال و دریافت داده‌های سریال و نیز بررسی وضعیت ارتباط سریال به کار می‌رود.

در دو مثال طراحی سیستم، سخن از مبدل سریال به موازی و مبدل موازی به سریال گفته شده است. اکثر میکروکنترلرها دارای پورت سریال هستند و می‌توانند نقش این دو مبدل را ایفا کنند.

با مطالبی که در این بخش آموختید، می‌توانید برنامه‌هایی بنویسید که به کمک پورت سریال اطلاعاتی به خارج کامپیوتر ارسال یا از پورت سریال داده‌هایی دریافت کنند. همانگونه که قبلاً گفته شد، ارتباط سریال در زبانهای تحت ویندوز به گونه‌ای متفاوت انجام می‌شود. در بخش ارتباط با پورتهای کامپیوتر تحت ویندوز، برنامه‌هایی به زبان `Visual C++` نوشته شده است که به کمک آنها می‌توانید برنامه‌نویسی پورت سریال در ویندوز را به راحتی انجام دهید.

استفاده از پورت سریال به شیوه وقفه

فرض کنید می‌خواهیم در کامپیوتر برنامه‌ای بنویسیم که داده‌هایی را از پورت سریال کامپیوتر خوانده و نمایش دهد. یک راهکار این است که مرتباً بررسی کنیم آیا داده‌ای در پورت سریال دریافت شده یا خیر و در صورت مثبت بودن پاسخ، آن را بخوانیم و نمایش دهیم (روش سرشماری یا `Polling`). این راه ساده است؛ اما وقت را بیهوده هدر می‌دهد. برنامه در هر ثانیه هزاران بار پورت سریال را کنترل می‌کند که آیا داده‌ای در آن دریافت شده است یا خیر؟ که به خصوص در مورد ارتباطاتی که حجم انتقال داده‌ها در آن زیاد نیست به صرفه نمی‌باشد. حداکثر سرعت ارتباط در این روش بدون از بین رفتن داده‌ها `34.7 Kbps` است.

راه دیگر این است که از پورت سریال در حالت وقفه استفاده کنیم. در این حالت، وقتی داده‌ای توسط پورت سریال دریافت شود وقفه‌ای در کامپیوتر رخ می‌دهد. می‌توانید با بازنویسی زیربرنامه پاسخ به آن وقفه (ISR)، واکنش لازم در قبال دریافت آن داده (مثلاً نمایش آن در کامپیوتر) را نشان دهید. با این کار دیگر نیازی نیست مرتباً پورت سریال را بررسی کنید؛ بلکه هرگاه داده‌ای دریافت شد، وقفه‌ای رخ خواهد داد. با این روش حتی در کامپیوترهای قدیمی سرعت به راحتی تا 115.2 Kbps بالا می‌رود.

تغییر ISR یک وقفه در کامپیوتر چندان پیچیده نیست و می‌توان آن را به کمک برنامه‌های اسمبلی یا سطح بالا (مانند C) انجام داد. در کتابهای زبان اسمبلی در بخش برنامه‌های مقیم در حافظه^۱ می‌توانید شیوه این کار را فرا بگیرید. برای این کار باید بدانید کامپیوتر کدام شماره وقفه را به پورت سریال اختصاص داده است. در شکل زیر که مربوط به یک کامپیوتر نوعی است و قبلاً شیوه مشاهده آن در ویندوز را دیدیم، وقفه IRQ4 (INT 0C h) به پورت سریال اختصاص داده شده است.



وقفه 4 IRQ به پورتهای COM1 و COM3 و وقفه 3 IRQ به پورتهای COM2 و COM4 اختصاص دارد.

اما چگونه باید از سیستم خواست که با دریافت یک کاراکتر اعلام وقفه کند؟ مقداردهی ثبات فعال‌کننده وقفه (از مجموعه ثباتهای پورت سریال) این کار را انجام می‌دهد. بیت‌های این ثبات را ببینید:

0	0	0	0	IN	ER	TBE	RxRDY
---	---	---	---	----	----	-----	-------

اگر بیت RxRDY «یک» باشد، دریافت داده باعث ایجاد وقفه می‌شود.

اگر بیت TBE «یک» باشد، خالی شدن بافر ارسال داده باعث روی دادن وقفه می‌شود.

بیت ER در رابطه با وقفه وضعیت خط و بیت IN در رابطه با وقفه وضعیت مودم عمل می‌کند.

¹ Terminate & Stay Resident - TSR

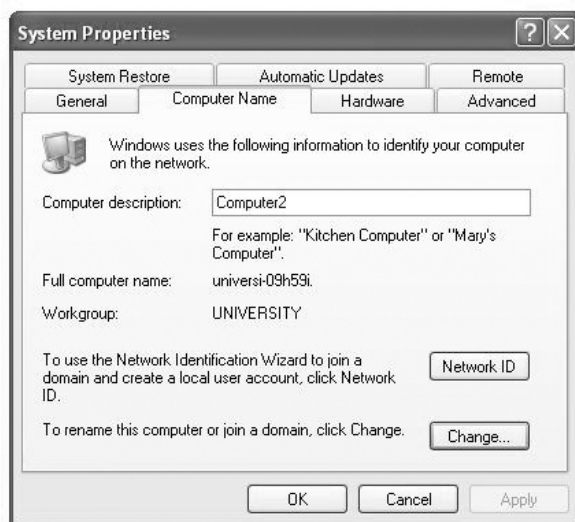
دستور `outportb(0x3F8+1,1)` وقفه دریافت داده را فعال می‌کند. از این پس، با دریافت داده توسط پورت سریال وقفه‌ای رخ می‌دهد.

شبکه مودم پوچ

همانطور که قبلاً دیدیم، با اتصال پورتهای سریال دو کامپیوتر به یکدیگر به کمک یک کابل مودم پوچ (که ساختار کامل آن را دیدید و البته عملاً در بیشتر اوقات، تنها اتصالات Tx و Rx کافی است) و با استفاده از نرم‌افزار HyperTerminal، می‌توانید ارتباط داده‌ای بین دو کامپیوتر برقرار کنید و حتی فایل بین دو کامپیوتر رد و بدل نمایید. در ویندوز ۲۰۰۰ یا XP، قابلیت ایجاد شبکه بین دو کامپیوتر به کمک اتصال پورتهای سریال دو کامپیوتر وجود دارد. در ادامه، با نحوه ایجاد این شبکه در ویندوز آشنا می‌شویم. برای ایجاد این شبکه، یکی از کامپیوترها باید میزبان^۱ (پذیرنده اتصال) و دیگری مهمان^۲ (برقرارکننده اتصال) باشد.

پیکربندی نرم افزاری کامپیوترهای میزبان و مهمان :

پیش از پیکربندی سیستمها مطمئن شوید که هر دو کامپیوتر به یک گروه کاری (مثلاً به نام University) متعلق هستند. با مراجعه به `Control Panel/System/Computer Name` از این موضوع مطمئن شوید و در صورت لزوم نام گروههای کاری دو سیستم را تغییر دهید. شکل زیر گروه کاری کامپیوتر میزبان (Computer2) را نشان می‌دهد.



الف) پیکربندی مشترک در هر دو سیستم

- با مراجعه به `Control Panel/Network Connection/Create a new connection` کار را آغاز کنید.

^۱ Host

^۲ Guest

- نوع شبکه را انتخاب کنید :



- نوع ارتباط را انتخاب کنید :



ب) ادامه پیکربندی نرم افزاری در کامپیوتر میزبان (Computer2) با انتخاب گزینه Host :

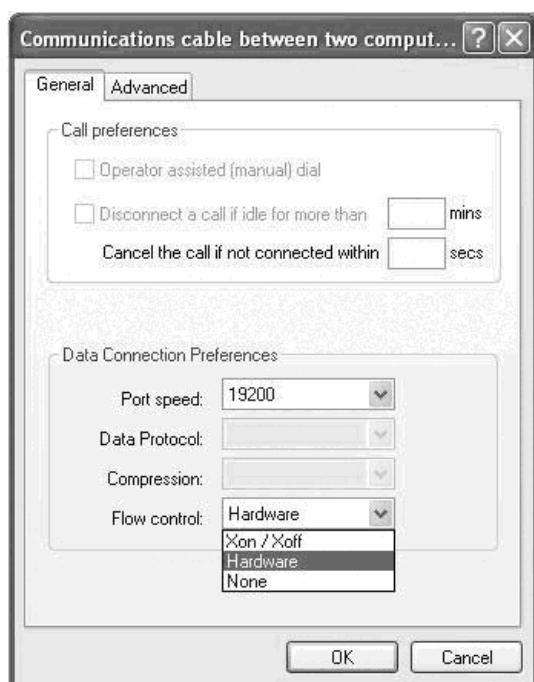
۱- پورت ارتباطی را مشخص کنید :

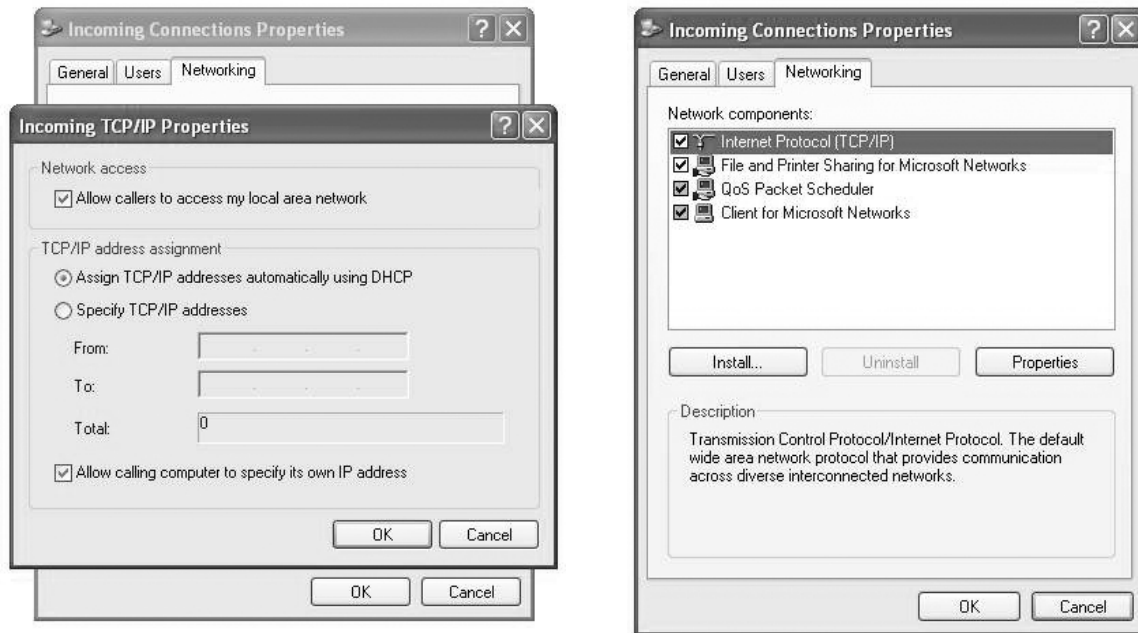


۲ - سطوح دسترسی کاربران را مشخص کنید :



۳- اکنون پیکربندی کامپیوتر میزبان پایان یافته است و در قسمت Network Connections گزینه ای به نام Incoming Connections ساخته شده است. با کلیک راست روی این گزینه و انتخاب Properties یا کلیک دابل روی خود گزینه، می توانید مشخصات ارتباط (از جمله سرعت که باید در دو طرف یکسان باشد) را تغییر دهید:





ب) ادامه پیکربندی نرم افزاری در کامپیوتر میهمان (Computer1) با انتخاب گزینه
: Guest

۱- نام کامپیوتر دیگر را وارد کنید :



۲- نوع ارتباط را مشخص کنید :





اکنون پیکربندی سیستم میهمان نیز پایان یافته است. یک گزینه با نام Computer2 در قسمت Network Connections سیستم میهمان ایجاد شده است. با کلیک روی آن و وارد کردن User Name و Password سیستم میزبان، می توانید به آن متصل شوید و با استفاده از آدرس IP یا نام آن، از تمام منابع به اشتراک گذاشته شده آن استفاده کنید. مشخصات ارتباط (از جمله سرعت که باید در دو طرف یکسان باشد) با کلیک روی Properties قابل تنظیم است.

پروژه‌های کاربردی با پورت سریال کامپیوتر

۱) سیستمی شامل سخت‌افزار و نرم‌افزار طراحی کنید که فایلی را از یک کامپیوتر به کامپیوتر دیگر ارسال کند. برنامه‌نویسی کامپیوترها را خودتان انجام دهید.

۲) سیستمی شامل سخت‌افزار و نرم‌افزار طراحی کنید یک رقص نور روی هشت عدد LED ایجاد کند. برای این کار LEDها باید هر یک ثانیه یک بار به صورت یکی‌درمیان روشن شوند. LEDها به یک میکروکنترلر و میکروکنترلر به پورت سریال کامپیوتر متصل است.

۳) سیستمی شامل سخت‌افزار و نرم‌افزار طراحی کنید که محتویات ۸ عدد کلید دو حالت را بخواند و روی نمایشگر کامپیوتر نشان دهد. کلیدها به یک میکروکنترلر و میکروکنترلر به پورت سریال کامپیوتر متصل است.

۴) سیستمی شامل سخت‌افزار و نرم‌افزار طراحی کنید که جهت و سرعت چرخش یک موتور پله‌ای را به کمک برنامه کامپیوتری کنترل کند. با این سیستم می‌توانید روباتی بسازید که به کمک پورت سریال کامپیوتر کنترل شود.

۵) سیستمی شامل سخت‌افزار و نرم‌افزار طراحی کنید که اعداد صفر تا ۹ را با فواصل زمانی یک ثانیه روی یک نمایشگر هفت‌قسمتی^۱ نمایش دهد. از تراشه مبدل کد BCD به

¹ Seven Segment

کد نمایشگر هفت قسمتی استفاده نکنید. برنامه خود را طوری بنویسید که نوع نمایشگر هفت قسمتی (آند مشترک یا کاتد مشترک) را در ابتدای برنامه از کاربر سؤال کند.

(۶) سیستمی شامل سخت افزار و نرم افزار طراحی کنید که یک ولتاژ آنالوگ بین صفر تا ۵ ولت که به کمک یک پتانسیومتر تولید می شود را از طریق پورت سریال خوانده و مقدار این ولتاژ را با دقت دو رقم اعشار روی نمایشگر کامپیوتر نشان دهد.

(۷) سیستمی شامل سخت افزار و نرم افزار طراحی کنید که یک ولتاژ آنالوگ بین صفر تا ۵ ولت که به کمک یک پتانسیومتر تولید می شود را از طریق پورت سریال خوانده و مقدار این ولتاژ را با دقت دو رقم اعشار روی سه عدد نمایشگر هفت قسمتی که به پورت موازی متصلند، نمایش دهد.

(۸) برنامه ای بنویسید که یک ارتباط همگام با پالس ساعتی که فرکانس آن از طریق برنامه قابل تنظیم باشد را روی دو پایه پورت موازی ایجاد کند.

(۹) سیستمی شامل سخت افزار و نرم افزار طراحی کنید که یک فایل را از طریق امواج مادون قرمز ارسال کند.

منابع

[1] "The 80x86 IBM PC & Compatible Computers", Muhammed Ali Mazidi and et al, Prentice Hall, 2000.

[2] "PC Interfaces under Windows", Burkhard Kaink and Hans-Joachim Berndt, Elektor Electronics Publications, 2002.

(این کتاب با ترجمه کیهان حدادشرق توسط انتشارات نشر علوم به چاپ رسیده است)

[۳] "اصول کامل راه اندازی و کنترل دستگاههای جانبی توسط کامپیوتر"، محسن شکیبافر، انتشارات نص، ۱۳۸۴.

[۴] "گذرگاهها و درگاههای کامپیوترهای شخصی"، عبدالمجید منصوریان فر و اصغر کریمی، انتشارات دانش پژوهان برین - ارکان، ۱۳۸۳

[۵] "دیجیتال پایه و طراحی مدارهای واسط"، شیرزاد شهریاری، انتشارات پرتونگار، ۱۳۸۵.

[۶] "مرجع علمی-کاربردی سخت افزار"، شیرزاد شهریاری، انتشارات جهاد دانشگاهی مشهد، ۱۳۸۴.