

دانشگاه صنعتی امیرکبیر

آزمایشگاه مکترونیک

آزمایش شماره ۶

دانشکده مهندسی مکانیک

2(serial port)

PC INTERFACING

## ۱-مقدمه

از آنجائیکه کامپیوتر دستگاهی منطقی است، ما به ندرت با واقعیت‌های فیزیکی که در پشت این منطق نهفته است، سروکار داریم. در حقیقت، در برخی موارد با در نظر گرفتن بخشهای پیچیده و بسیار زیاد کامپیوتر دقت در اینکه چه عناصری در داخل آن قرار دارند کاملاً سخت و پیچیده است. برای مثال همه می‌دانند در کامپیوترها، داده‌های منطقی به صورت بیت (bit) بکار رفته‌اند.

## ۲-حالت‌های انتقال داده‌ها

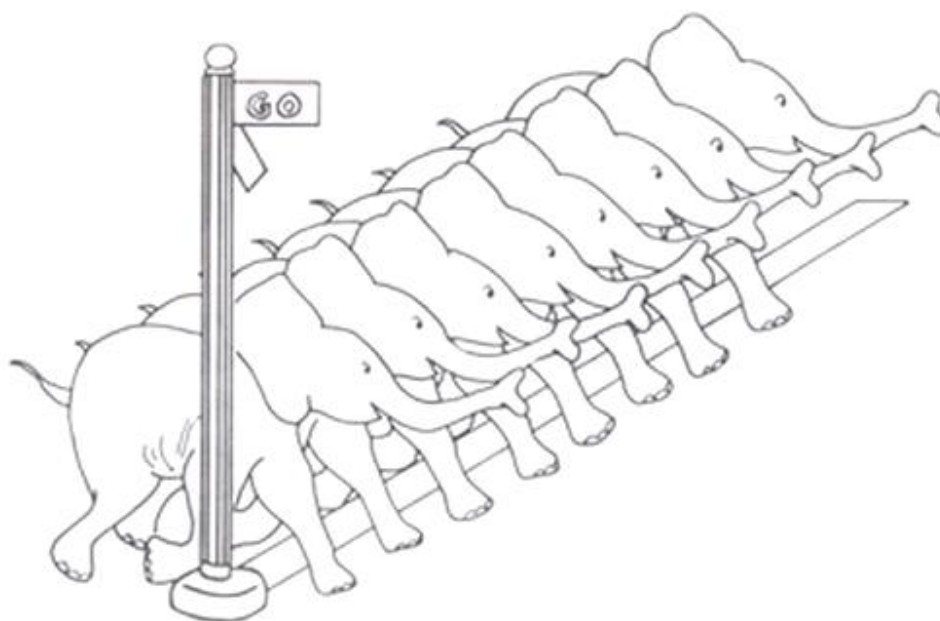
اگر چه بیت یک مفهوم ذهنی است، با این وجود از لحاظ فیزیکی ولتاژی است که دامنه آن محتوای بیت را نشان می‌دهد. جریان الکتریکی که داده‌های کامپیوتر توسط آن مشخص می‌شوند، با جریانی که در یک استریو از نقطه‌ای به نقطه‌ای دیگر توسط سیم‌ها انتقال پیدا می‌کند تفاوتی نمی‌کند. هنگامی که لازم است بیت‌ها درون خود کامپیوتر جابجا شوند، دقیقاً مانند هر ولتاژ دیگری اینکار توسط چند سیم انجام می‌گیرد. در صورتیکه داده‌هایی که انتقال می‌یابد بصورت مجموعه ۸ بیتی موسوم به بایت (byte) که در اغلب کامپیوترها اینگونه است، هستند و به هشت سیم مجزا که همزمان هشت جریان الکتریکی موجود را بین دو نقطه انتقال دهد، نیاز است. با نگاه‌گذاری به درون کامپیوتر، می‌توان این مجموعه سیستم‌ها را مشاهده کرد. این سیستم‌ها به صورت مجموعه‌ای از نوارهای مسی که در دو طرف مدار چاپی قرار دارند از طرفی به طرف دیگر و از بین مدارات مجتمع و سایر قطعات گذشته‌اند.

انتقال همزمان این ولتاژهای هشت بیتی که یک بایت را تشکیل می‌دهند، به انتقال موازی (Parallel) موسوم است. بنابراین

انتقال موازی، بایت به بایت صورت می‌گیرد. چون هر هشت بیت، همزمان به مقصد خود می‌رسند، لذا می‌توان انتقال موازی



داده‌ها را با سرعت بسیار زیادی انجام داد. سرعت زیاد این روش، استفاده از آن را در موارد ارجح می‌سازد. شکل ۱ مفهوم ساده انتقال موازی را نشان می‌دهد.



شکل ۱) مفهوم ساده انتقال موازی [۱۵]

انتقال داده‌ها، مخصوصاً انتقال سریع آنها، نیاز به محیط شدیداً کنترل شده‌ای دارد. باید دمای داخلی کامپیوتر ثابت بوده و مشخصات الکتریکی مقاومتها، خازنها و سلفها بدقت محاسبه شده باشد. در مدت زمانی که داده‌ها در حال جابجایی در کامپیوتر هستند، شرایط این محیط پایدار و از قبل مشخص شده است. اما بخش عمده‌ای از داده‌های کامپیوتر باید به دنیای خارجی متفاوت با آن انتقال یابد. در حقیقت، موفقیت تجاری کامپیوتر مستقیماً بستگی به ویژگی‌های برتر آن در جابجایی داده‌ها از کامپیوتر به دستگاههای جانبی از قبیل چاپگرها، ترمینالها، مودم‌ها و بافرهای چاپ دارد. انتقال داده‌ها از کامپیوتر، به یکی از دستگاههای خارجی را خروجی (output) و برعکس، انتقال داده‌ها از یکی از دستگاههای خارجی به درون کامپیوتر را ورودی (input) گویند. این دو عمل را در مجموع ورودی / خروجی (input/ output) یا به طور مختصر I/O می‌گویند.



- اگر محیط داخلی کامپیوتر شما به طور کامل با شرایط صحیح تطبیق داده شده باشد، باز هم در خارج کامپیوتر، بی‌نظمی و عدم تطبیق وجود دارد. از نظر طراحی، مطلقاً هیچ نظری در مورد وضعیت محیطی داده‌ها در خارج کامپیوتر، با آن مواجه خواهند شد، نمی‌توان داد. مشکلی که با توجه به مطالب فوق بوجود می‌آید، به صورت پرسشهای زیر، قابل طرح است:
  - چگونه می‌توان داده‌ها را از مدار بسیار حساسی که با کمی طویل کردن سیمها و یا حتی تغییر موقعیت آنها، کار سیستم را مختل می‌کند، به بیرون کشید؟
  - چگونه می‌توان کار طبیعی کامپیوتر را در جایی که بعضی‌ها می‌خواهند از آن در سیستم کامپیوتری نان برشته کنی یا زنگ درب خانه خود استفاده کنند، تضمین کرد؟
- به عبارت دیگر، باید به نحوی از وقوع حوادث زیان آور که در خارج کامپیوتر، رخ می‌دهد جلوگیری کرده و از صدمه دیدن مدارات داخلی کامپیوتر ممانعت کرد. پس در اینجا به یک میانجی یا اینترفیس (Interface) نیاز است که دو محیط خارج و داخل کامپیوتر را به یکدیگر مرتبط سازد. از آنجائیکه میانجی را می‌توان نوعی درب ورودی-خروجی برای کامپیوتر دانست، لذا بعضی اوقات به آن پورت I/O یا فقط پورت (یعنی درب) نیز گفته می‌شود.
- یک مثال آشنا برای میانجی، چیزی است که بین یک منزل و شرکت برق منطقه‌ای وجود دارد. شرکت برق انرژی الکتریکی را به صورت ولتاژ بسیار زیاد انتقال می‌دهد. آوردن برق با ولتاژ ۱۰۰۰۰۰ ولت به درب خانه‌ها، کار عاقلانه‌ای نیست و بی‌شک منجر به مرگ صاحبخانه خواهد شد. بنابراین بعنوان اولین مرحله برقراری ارتباط با دستگاه‌های برقی داخل خانه، باید این ولتاژ زیاد، تا مقادیر کم و بی‌خطر کاهش یابد. بعلاوه شرکت برق شرایط متقاضیان استفاده از شبکه را تعیین می‌کند. بعنوان مثال قبل از اینکه انشعاب برق مصرف کننده برقرار شود، شرکت برق محل ساختمان را برای اطمینان از رعایت شرایط تعیین شده بازرسی می‌کند. در این مثال، میانجی سه هدف عمده را برآورده می‌کند. اولاً این عمل از خسارت دیدن سیستم توزیع شرکت برق جلوگیری می‌کند. دوماً از بروز آتش سوزی جلوگیری می‌کند و سوماً امنیت و قابلیت استفاده از برق را بیشتر می‌کند.



- محافظت از سیستم، یکی از هدفهای جانبی میانجی است و هدف عمده آن فراهم آوردن امکان انتقال داده‌ها است.

نکته آخر، قابلیت استفاده بیشتر نیز شایان اهمیت است. اگر محافظت از سیستم به قیمت از دست دادن قابلیت

استفاده بیشتر بدست آید، دیگر این میانجی نه جالب خواهد بود و نه مفید. بطور نسبی طراحی مداری که در مقابل

صدمات خارجی وارده، مقاوم باشد ساده است اما ساخت وسیله‌ای که کار با آن حتی برای مبتدیان نیز راحت باشد،

به مهارت بیشتری نیاز دارد.

از زمانی که اینچنین میانجی‌هایی بوجود آمدند، انتقال داده‌ها به دنیای خارج میسر شد. اما متأسفانه سریعاً مشکلات

جدیدی به وجود آمد. اولین مشکل بسیار واضح بود: قیمت سیستم‌ها در مسیرهای طولانی مبلغ زیادی می‌شد. در انتقال

موازی حداقل نه سیم، هشت سیم برای انتقال بیت‌های داده و یکی برای سیم مشترک، نیاز است. علاوه بر این به طور معمول

برای کنترل انتقال داده‌هایی که به میانجی می‌روند سیم‌های بیشتری نیاز است. کسی که تا حال یک کابل میانجی برای

برای موازی کاملی را ساخته باشد، تصدیق خواهد کرد که سرو کار داشتن با ۳۰ سیم. یا بیشتر گران، پرهزینه و

خسته کننده است. مشکل دیگری که با انتقال موازی داده‌ها ارتباط دارد، ماهیت ولتاژ بیت‌ها است. هنگامی که ولتاژ یک

بیت، حالت خود را از یک به صفر یا برعکس تغییر می‌دهد، این عمل بسیار سریع، در حدود چند نانوثانیه صورت

می‌گیرد. سرعت زیاد انتقال از حالت یک به صفر، به خودی خود یک قسمت ضروری از عمل پردازش در انتقال داده‌ها

است تغییرات آهسته بین حالات صفر و یک بعنوان داده قابل تشخیص نیست. متأسفانه هر چه کابل بلندتر شود،

مشخصات الکتریکی آن (ظرفیت خازنی و سلفی) سرعت تغییرات را محدود کرده و احتمال از دست رفتن داده‌ها بیشتر

می‌شود. بدلیل مشکل فوق، سرعت ذاتی انتقال موازی داده‌ها در کابل‌های بلند مسئله ساز است.

این دو مانع در انتقال موازی داده‌ها- گرانی و ایجاد اختلال در داده‌ها- استفاده از این روش را به بعضی از دستگاه‌های

جانبی (مثل چاپگرها) که اغلب در نزدیکی کامپیوتر قرار داشته و باید با سرعت زیادی کار کنند، محدود می‌کند. بدلیل

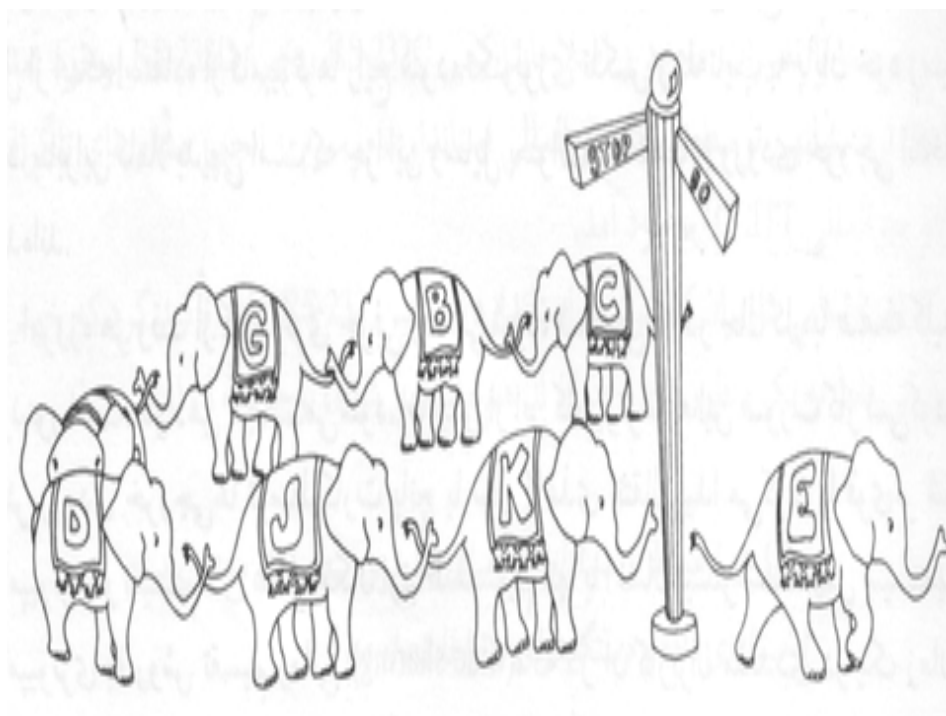
عدم نیاز به سیم‌های طولانی در داخل کامپیوتر، برای انتقال اطلاعات در سیستم داخل آن از روش انتقال موازی داده‌ها

استفاده می‌کنند. برای انتقال داده‌ها نیاز به روشی با هزینه کمتر و در عین حال پایدارتر داریم. از طرف دیگر، برای قابل



رقابت بودن این روش با دیگر روشها باید علاوه بر انتقال بدون اشکال داده‌ها کاهش عمده‌ای از لحاظ قیمت و حجم نیز در آن به عمل آید.

به جای ارسال همزمان همه بیت‌ها بر روی چند سیم، روش به سیمی دیگر، فرستادن تک تک آنها بعد از دیگری است. در طرف دیگر، پردازش به صورت برعکس انجام می‌گیرد و بیت‌های مجزا دوباره به صورت بایت اصلی جمع می‌شوند. در این روش از مدار الکتریکی ساده‌ای که تنها از دو سیم تشکیل شده استفاده می‌شود و در هر لحظه فقط یک بیت از داده‌ها، انتقال می‌یابد. روش فوق موسوم به انتقال سری یا Serial شکل ۲ بوده و موجب کاهش حجم و قیمت، نسبت به تکنیک انتقال موازی می‌شود.



شکل ۲) مفهوم ساده انتقال سریال [۱۵]

متأسفانه امتیاز کاهش حجم و قیمت، با کاهش بازده این روش از دست می‌رود. در این روش زمان لازم برای انتقال هشت بیت داده به طور جداگانه و پشت سرهم، حداقل هشت برابر زمان انتقال همزمان آنها است. خوشبختانه در اکثر کاربردها کم بودن سرعت این روش محدودیت زیادی را بوجود نمی‌آورد. اگر به چند نمونه از دستگاه‌های جانبی توجه کنیم، خصوصیت جالب و مشترکی را در آنها مشاهده می‌کنیم. این دستگاه‌ها حداقل در مقایسه با سرعت داخلی ریز پردازنده



بسیار کند هستند. تمام این دستگاه‌ها مقداری وقت تلف می‌کنند. مخصوصاً قسمت‌های مکانیکی، محدودیت سرعت بیشتری بوجود می‌آورند؛ مثلاً محدودیت سرعت چاپگرها بخاطر کند بودن سرعت حرکت هد چاپ آنها است. در مودمها محدودیت فرکانسی خطوط تلفن ایجاد محدودیت می‌کند، و کند بودن دیسک درایوها بخاطر کند بودن سرعت چرخش دیسک است.

و بنابراین داشتن سرعت، در روش انتقال موازی داده‌ها، عملاً بخاطر کند بودن دستگاهها محدود می‌شود. در روش انتقال سری هنگام کار با دستگاه‌های جانبی، سرعت کاهش می‌یابد. در این حالت، کاهش سرعت در مقایسه با انتقال درست و کامل داده‌ها اهمیت کمتری دارد.

### ۳- میانجی‌های استاندارد

تا به حال در مورد قرار دادن میانجی بین کامپیوتر و دنیای خارج از آن که قابلیت انتقال داده‌ها را به صورت سری داشته باشد، صحبت کردیم. اما چون روشهای زیادی برای طراحی صحیح مدارات میانجی وجود دارد، لذا امکان طراحی میانجی‌های متعددی که کار خاصی را انجام دهند، وجود دارد. در طراحی مدارات میانجی فوق، مسائل و مشکلات گوناگون و اساسی وجود دارد؛ از جمله سازگار بودن با میانجی‌های دیگر.

به مرور زمان، میانجی‌های سری انواع مختلفی پیدا کرده‌اند. از سال ۱۹۸۶ به بعد امکان انتقال بین قاره‌ای داده‌ها میسر شده است. در حقیقت با این کار علائم نقطه و خط در موریس، امروزه به علائم صفر و یک در مخابره داده‌ها، در کامپیوتر تبدیل شده است.

رشد و توسعه تکنولوژی انتقال سری و افزایش انواع مختلف دستگاه‌های تله تایپ و تلکس، قطعاً موجب ایجاد اشکالاتی در

ارتباط بین هر دستگاه و میانجی مربوط به آن می‌شود. در علم الکترونیک، از خطوط تلگراف به دلیل عدم توجه دقیق به

تغییرات زیاد مشخصه‌های الکتریکی وسایلی که به آنها متصل هستند، نمی‌توان در همه موارد به عنوان یک میانجی مطلوب

استفاده کرد. چون این خطوط در موارد مختلف مورد استفاده قرار می‌گیرند، طراحان دستگاه‌های سری با اطمینان کامل، نکاتی

که مربوط به بهبود مشخصه‌های الکتریکی میانجی می‌شود را در نظر نمی‌گیرند. بنابراین از این جهت انگیزه واقعی برای تکمیل

و توسعه میانجی‌های استاندارد وجود ندارد.



ورود تدریجی کامپیوترها به شاخه ارتباطات انگیزه اصلی طرح مسائل فوق است. قبل از اینکه استفاده از کامپیوترها رایج شود، تکنولوژی تلکس و تله تایپ به پایان خود رسیده بود. بنابراین کاملاً بدیهی است که چرا این وسایل به عنوان دستگاههای ورودی خروجی انتخاب شده‌اند.

#### ۴- پورت چاپگر (به آزمایش 1 PC INTERFACING مراجعه شود)

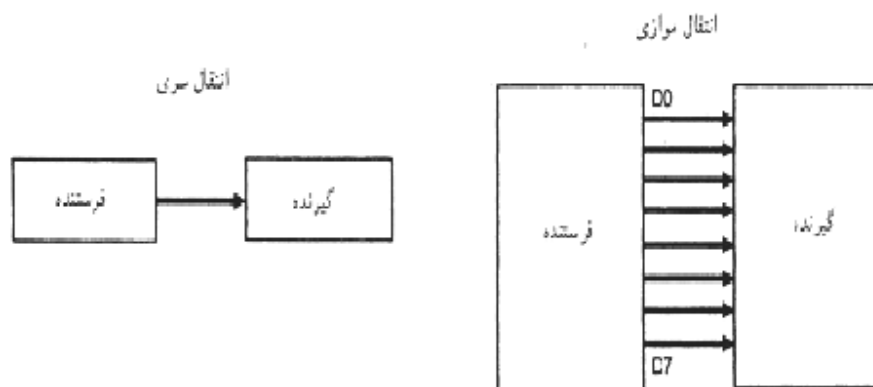
#### ۵- انتقال سریال

کامپیوترها داده را به دو طریق موازی و سریال انتقال می‌دهند. در انتقالهای موازی داده، اغلب از ۸ خط داده و یا بیشتر استفاده می‌شود. در این حالت داده می‌تواند بوسیله‌ای که فقط چند قدم دورتر است انتقال یابد (حداکثر ۱/۸m) مثالهایی از انتقال موازی عبارتند از چاپگرها و دیسک‌های سخت که از کابلهای نواری استفاده می‌کنند. گرچه در این موارد حجم بسیاری از داده در زمان کوتاهی جابجا می‌شود، ولی فاصله نمی‌تواند چندان طولانی باشد، در انتقال به وسیله‌ای که چندین متر دورتر واقع شده است، روش سریال بکار برده می‌شود. در تبادل سریال برعکس تبادل موازی، هر بار یک بیت ارسال می‌گردد.

#### ۶- اصول تبادل سریال

وقتی که در ریز پردازنده‌ای با دنیای خارج تبادل اطلاعات می‌کند بسته‌های داده را در ابعاد بایت فراهم می‌نماید. در بعضی موارد همچون چاپگرها، اطلاعات به راحتی از گذرگاه داده ۶ بیت اخذ شده و به گذرگاه داده ۸ بیتی چاپگر تحویل داده می‌شود. در این روش اگر کابل زیاد طولانی نباشد عملی است. زیرا کابلهای طولانی سیگنالها را تضعیف و حتی تغییر شکل می‌دهند. بعلاوه مسیر ۸ بیت گران هم هست. بنابراین در تبادل داده بین دو سیستم با فاصله‌ای تا صدها فوت یا میلیونها کیلومتر از انتقال سریال استفاده می‌شود. شکل ۳ نمودار انتقال داده سریال را در مقایسه با موازی نشان می‌دهد.





شکل ۳) انتقال سریال در مقایسه با سریال موازی [۱۵]

- واقعیت استفاده از یک خط داده در تبادل سریال به جای خط داده ۸ بیت در تبادل موازی، نه تنها موجب ارزانی

فرایند می شود بلکه موجب تعادل اطلاعات دو کامپوتر واقع در دو شهر از طریق خط تلفن می گردد. برای به

راه اندازی تبادل سریال داده، بایت داده باید از گذرگاه ۸ بیتی ریز پردازنده گرفته شده و با استفاده از شیفت رجیستر

ورودی- موازی / خروجی- سریال به بیت های سریال تبدیل گردد.

که آنگاه قابل ارسال به یک خط داده خواهد بود. واضح است که در سمت گیرنده باید یک شیفت رجیستر ورودی - سریال /

خروجی موازی برای دریافت داده ارسالی وجود داشته و پس از بسته بندی کردن به صورت بایت، آنها را به گیرنده تحویل دهد.

البته اگر قرار باشد از طریق خط تلفن انتقال یابد، باید آن را از صفر و یک به سیگنال های صوتی که سینوسی باشند تبدیل کرد.

این تبدیل توسط دستگاهی بنام مودم که خلاصه مودولاتور / دمودولاتور است، انجام می شود.

وقتی که فاصله کوتاه است، سیگنال دیجیتال به همان شکل طبیعی اش روی یک سیم معمولی ارسال می گردد. و نیازی به

مدولاسیون ندارد. این طریقی است که در PC داده بین صفحه کلید و بورد اصلی انتقال می یابد. با این وجود برای راه دور، انتقال

داده بوسیله تلفن نیاز به مودم دارد تا صفرها و یک ها را به سیگنال های صوتی برعکس تبدیل نماید.

تبادل داده دو روش را بکار می برد: یکی روش همزمان و دیگری روش غیرهمزمان. روش همزمان هر بار یک بلوک از داده ها

را انتقال می دهد، درحالی که روش غیرهمزمان هر بار یک بایت را منتقل می سازد. می توان برای پیاده کردن هر یک از دو روش

فوق، نرم افزاری تهیه کرد، ولی برنامه ها می توانند طولانی و خسته کننده تر باشند. به این دلیل، تراشه های IC خاصی بوسیله



سازندگان متعددی برای تبادل سریال داده ساخته شده است. این تراشه‌ها عموماً UART (فرستنده-گیرنده غیرهمزمان یونیورسال) و USART (فرستنده-گیرنده همزمان-غیرهمزمان یونیورسال) خوانده می‌شود.

## ۷-ارسال نیمه و تمام دوطرفه

در ارسال داده اگر بتوان داده را ارسال و یا دریافت کرد گوییم انتقال دوطرفه است. این برخلاف انتقالهای ساده همچون چاپگرهاست که در آن، کامپیوتر فقط داده را ارسال می‌نماید. ارسال می‌تواند نیمه و یا تمام دوطرفه باشد. این بستگی به انتقال همزمان داده در دو جهت دارد. اگر داده در هر زمان فقط در یک جهت ارسال شود به آن نیمه دوطرفه گویند. اگر امکان ارسال دو جهت همزمان داده وجود داشته باشد آن را تمام دوطرفه می‌نامند. البته تمام دوطرفه علاوه بر خط زمین نیاز به دو خط داده دارد، که یکی برای ارسال و دیگری برای دریافت می‌باشد. بدین طریق ارسال و دریافت همزمان صورت می‌گیرد.

## ۸-تبادل سریال غیر هماهنگ و بسته‌بندی داده

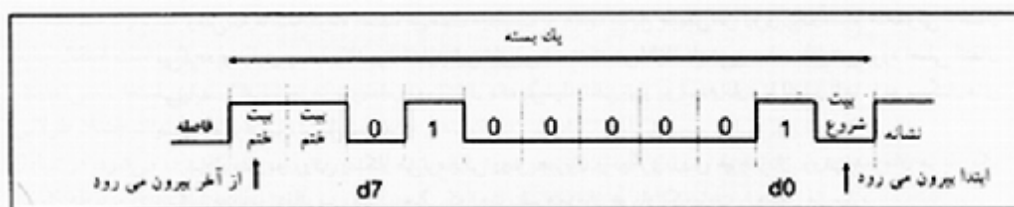
به هنگام انتقال سریال، داده رسیده به مقصد از صفرها و یک‌ها تشکیل شده و بنابراین تشخیص آن مشکل است مگر اینکه فرستنده و گیرنده براساس مجموعه قوانینی با یکدیگر توافق داشته باشند. این قوانین که به آن پروتکل می‌گویند، چگونگی بسته بندی داده، تعداد بیت‌های تشکیل دهنده کاراکتر و زمان ابتدا و انتهای داده را معین می‌سازند.

## ۹-بیت‌های شروع و ختم

تبادل سریال غیرهمزمان داده به طور گسترده‌ای برای ارسال‌های مبتنی بر کاراکتر مورد استفاده قرار می‌گیرند، ولی انتقال داده مبتنی بر بلوک، از روش همزمان استفاده می‌نماید. در روش غیرهمزمان، هر کاراکتر بین بیت‌های شروع و ختم قرار می‌گیرد. به این عمل، بسته‌بندی می‌گویند. بیت شروع همیشه یک بیت است. ولی ختم می‌تواند یک یا چند بیت باشد. در روش بسته‌بندی داده برای تبادل غیرهمزمان، داده‌ها مانند کاراکترهای اسکی، بین بیت‌های شروع و ختم بسته‌بندی می‌شوند. بیت شروع همیشه صفر و بیت‌های ختم همواره ۱ است. به عنوان مثال در شکل ۴ کاراکتر اسکی A با باینری ۰۱۰۰۰۰۱ بین بیت شروع و دو بیت ختم قرار گرفته است.



در شکل ۴ هنگامی که انتقالی صورت نمی‌گیرد، سیگنال ۱ است که به آن نشانه می‌گویند. به ۰.۰ فاصله گویند. باید دقت کرد که ارسال، با بیت شروع، آغاز شده و به دنبال آن D0 یعنی کم ارزش‌ترین بیت و پس از آن بقیه بیتها تا MSB (D7) و نهایتاً دو بیت ختم به معنای پایان کاراکتر A ارسال می‌شوند.



شکل ۴) انتقال سریال کاراکتر اسکی A با باینری ۰۱۰۰۰۰۰۱ [۱۵]

در تبادل غیر همزمان، تراشه‌های جانبی و مودم‌ها می‌توانند برای داده‌های ۵، ۶، ۷ یا ۸ بیت عرض برنامه‌ریزی شوند. این تعداد علاوه بر ۱ یا ۲ بیت ختم می‌باشند. در سیستم‌های قدیمی‌تر، کاراکترهای اسکی، ۷ بیت بودند، ولی در کاراکترهای اسکی توسعه یافته، ۸ بیت برای هر کاراکتر اختصاص یافته است. صفحه کلیدهای کوچک غیراسکی از کاراکترهای ۵ و ۶ بیتی استفاده می‌کنند. در سیستم‌های کمی قدیمی‌تر به دلیل کندی وسایل مکانیکی گیرنده، ۲ بیت ختم بکار برده می‌شد تا به دستگاه، قبل از بایت بعدی فرصت کافی برای سازماندهی داده شود. با این وجود در PC های مدرن، استفاده از یک بیت ختم رایج است. فرض کنید که می‌خواهیم یک فایل متن از کاراکترهای اسکی را با ۲ بیت ختم ارسال کنیم. در این صورت برای هر کاراکتر ۱۱ بیت خواهیم داشت زیرا ۸ بیت متعلق به کد اسکی است. بنابراین برای هر کاراکتر ۸ بیتی، ۳ بیت اضافه وجود دارد یا به نوعی دیگر ۳۰٪ سر بار موجود است.

## ۱۰- استاندارد RS232

### ۱۰-۱- تاریخچه:

برای ایجاد سازگاری میان تجهیزات تبادل ساخت سازندگان مختلف، یک استاندارد واسطی بنام RS232C بوسیله سازمان صنایع الکترونیک (EIA) در ۱۹۶۰ برپا شد. در سال ۱۹۶۳ استاندارد فوق اصلاح و RS232A نامیده شد. RS232C, RS232B به



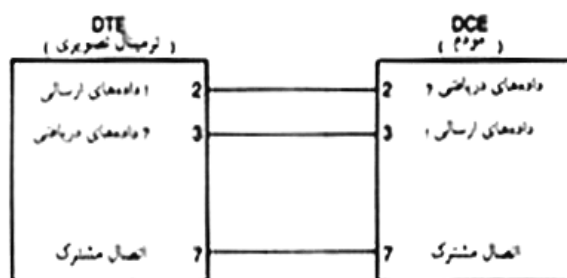
ترتیب در سالهای ۱۹۶۵ و ۱۹۶۹ معرفی گردیدند. با این وجود، چون استاندارد مذکور قبل از خانواده منطقی TTL بنا نهاده شد، سطوح ولتاژ ورودی و خروجی، سازگار با TTL نیستند. در RS232 منطق ۱ با ۲۵- ولت تعریف می شود ضمن اینکه ۳+ تا ۲۵+ هم بیت ۰ است. فاصله تا ۳+ تعریف نشده است. به این دلیل برای اتصال هر RS232 به یک سیستم مبتنی بر ریز پردازنده، باید از مبدل‌های ولتاژی همچون MC1488, MC1489 برای تبدیل از سطح TTL به سطح RS232 و بر عکس استفاده کنیم.

MC1488 برای تبدیل سطح ولتاژ TTL به RS232 و MC1489 برای تبدیل از سطح ولتاژ RS232 به TTL استفاده می شود.

#### ۱۰-۲- اصول اتصال میانجی:

ساده ترین شکل RS232 فقط شامل دو سیم است که یکی برای انتقال داده ها و دیگری اتصال مشترک است. اتصال مشترک را اغلب به اشتباه زمین مدار می نامند. البته این سیم مطلقاً کاری با زمین واقعی ندارد، بلکه تنها بعنوان یک مبنای ولتاژ برای مدار میانجی بکار می رود. یعنی نقطه‌های از مدار که تمام ولتاژها نسبت به آن نقطه سنجیده می شوند. مفهوم اتصال مشترک، حتی برای افرادی که در تجزیه و تحلیل مدارات الکتریکی تبحر پیدا کرده اند، کاملاً گیج کننده است.

ما برای ساخت یک میانجی از دو وسیله استفاده می کنیم. ارسال داده ها بطور ضمنی، لزوم وجود دستگاهی، برای دریافت آنها را بیان می کند. در شکل ۵ ما یک دستگاه DCE که شباهت زیادی به مودم دارد را در نظر می گیریم. داده های ارسال شده، از پایه ۲ دستگاه DTE خارج شده و به پایه ۲ دستگاه DCE وارد می شود. اگر داده های ارسال شده از طریق یک پایه در حال ورود به DCE باشد منطقی است که تصور کنیم که در آن لحظه DCE باید در حال دریافت همان داده باشد.

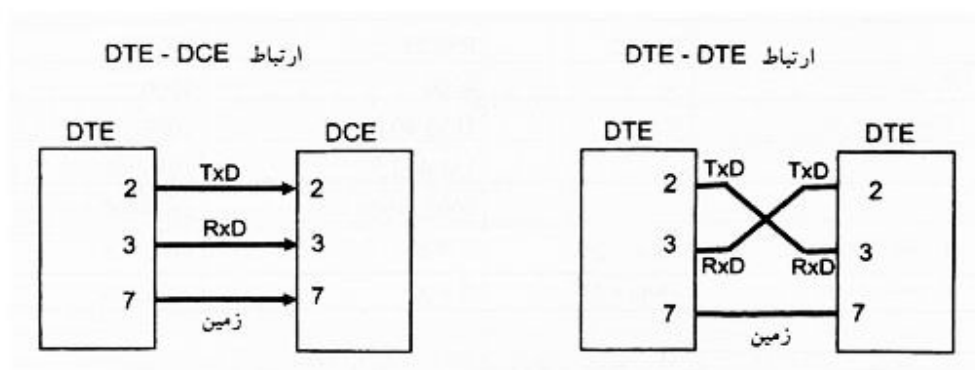


شکل ۵) ارتباط سریال مودم [۱۵]



## ۱۰-۳- داده های دوطرفه

ترمینال فرضی ما کاراکترهایی که روی صفحه کلید آن تایپ می شود را ارسال می دارد. مودم این کارکترها را دریافت کرده و آنها را روی خط تلفن می فرستد. ترمینالها و مودمها معمولاً یکطرفه نیستند و هر کدام مکن است در جهت معکوس نیز کار کنند. بعنوان مثال مودمها معمولاً کاراکترها را از خط تلفن گرفته و آنها را به ترمینال می فرستند. بطور مشابه، ترمینال نیز کارکترها را از خروجی مودم دریافت کرده و آنها را روی صفحه تصویر نشان می دهد. بنا براین هر دو دستگاه ساده ما باید توانایی ارسال و دریافت داده ها را داشته باشند. در شکل ۶ جریان داده ها بصورت مبادله دو طرفه بین دو دستگاه برقرار شده است. این عمل دقیقاً مانند اتصال دو تلفن به یکدیگر مانند شکل ۷ می باشد.

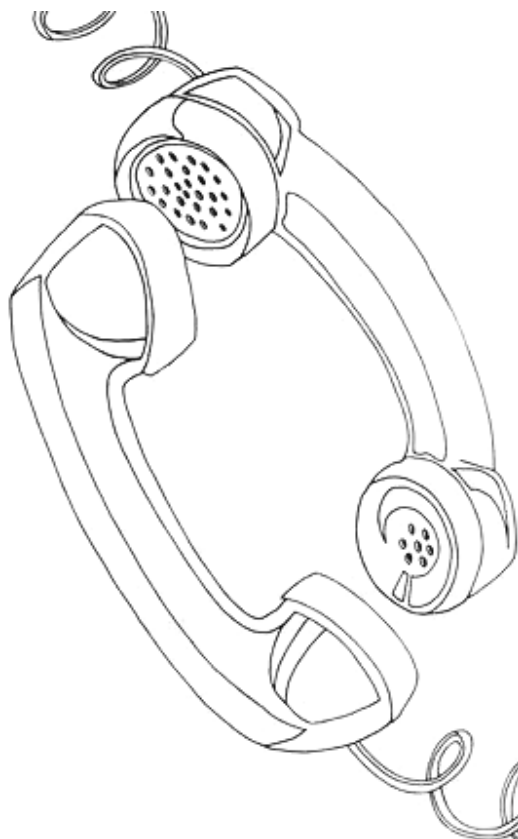


شکل ۶) ارتباط DTE-DTE و DTE-DCE [۱۵]

دستگاههای DTE داده ها را از پایه ۲ ارسال داشته و از پایه ۳ دریافت می کنند.

دستگاههای DCE داده ها را از پایه ۳ ارسال داشته و از پایه ۲ دریافت می کنند.






شکل ۷) مفهوم ساده انتقال دو طرفه بین دو دستگاه [۱۶]

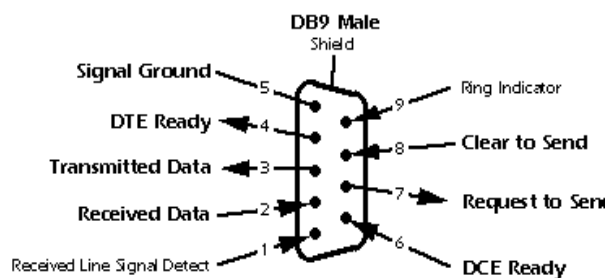
#### ۱۰-۴- پایه های RS232

شکل ۸ پایه ها و اسامی آنها را برای کابل **RS232** نشان می دهد که اغلب آنها را رابط DB\_25 می خوانند. به دلیل اینکه همه پایه ها در کامپیوترهای جدید بکار نرفته اند، IBM نوع DB\_9 را که فقط ۹ پایه داشت طبق شکل ۸ معرفی کرد. پورت ۹ پایه سریال در شکل ۹ نشان داده شده است.



	9 Pin Connector on a DTE device (PC connection)
Male RS232 DB9	
Pin Number	Direction of signal:
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) Outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal
5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

شکل ۸) شماره پینهای پورت سریال [۱۷]



شکل ۹) پورت ۹ پایه سریال [۱۷]

#### ۱۱-۵- عمل دستداد (Handshaking)

نکات زیادی در مورد استفاده از میانجی کامپیوتر وجود دارد که باید آنها را فهمید. تنها موضوع اصلی، کنترل تاثیر متقابل دستگاهها بر روی یکدیگر است. این مفهوم به ظاهر نا مانوس را با یک لغت آشنا دستداد (Hand Shaking) یعنی دست به دست، به هم دادن بیان می کنند. دستداد روشی است که به کمک آن، جهت انتقال داده ها در میانجی، کنترل و تثبیت می شود. دستداد نرم افزاری هنگامی صورت می گیرد که یک دستگاه از طریق محتوای داده ها دستگاه دیگر را کنترل کند.



بعنوان مثال یک روش کنترل چاپگر این است که کامپیوتر کاراکترهای خودش را به صورت خط به خط بفرستد. پس از هر خط کامپیوتر بطور خودکار، کارکتری به مفهوم زیر به چاپگر ارسال می کند: \*سطر تمام شد، برای ارسال سطر بعدی منتظر نظر شما هستم \* چاپگر خط را دریافت کرده و آنرا چاپ می کند و پس از آن کاراکتری به کامپیوتر ارسال می کند که به این معنی است: \*آماده دریافت خط بعدی هستم \* این کار روش سادهای برای کنترل چاپگر است.

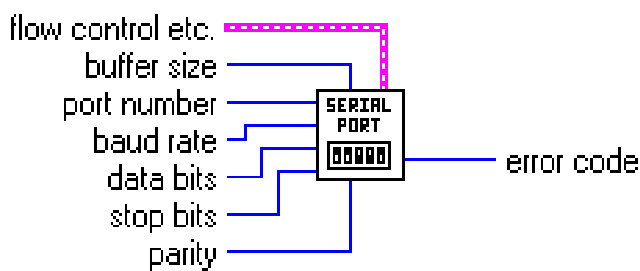
در دستداد سخت افزاری روش اصولیتری بکار می رود که در آن چاپگر با عوض کردن سطح ولتاژ یک سیم کامپیوتر آنرا مجبور به مکث در فرستادن کاراکترها می کند. عیب این نوع دستداد، این است که تنها درجایی از این روش می توان استفاده کرد که دستگاهها را بتوان از طریق کابل به کامپیوتر وصل کرد.

## ۱۲-آزمایش شماره ۱ **Loopback Test for Serial Port**

در این قسمت با یک آزمایش ساده و اتصال پین ۳ و ۲ پورت سریال به یکدیگر، عمل نوشتن روی پورت و خواندن اطلاعات از پورت را در نرم افزار labview مورد بررسی قرار می دهیم.

### ۱۲-۱-توضیح دستورات استفاده شده:

اولین قدم در برنامه نویسی برای پورت سریال تنظیم پارامترهای مربوط به انتقال سریال در دستور Serial Port Init (شکل ۱۰) است.



شکل ۱۰- دستور Serial Port Init

✓ flow control etc: کلاستری است برای تنظیمات مربوط به handshaking که ما در این آزمایشگاه از

مقادیر پیش فرض آن استفاده می کنیم.

✓ buffer size: حجم بافر ورودی و خروجی را بر حسب بایت تعیین می کند .



✓ **port number** : شماره پورت مورد استفاده را تعیین می کند: COM1: 0 و COM2: 1

✓ **baud rate**: سرعت انتقال اطلاعات است (مقدار پیش فرض ۹۶۰۰)

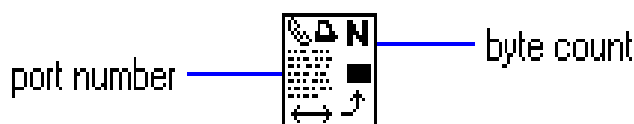
✓ **Data bits**: تعداد بیت‌های مربوط به اطلاعات انتقال داده شده (مقدار پیش فرض ۸)

✓ **Stop bits**: تعداد بیت‌های نشاندهنده پایان انتقال (مقدار پیش فرض 0 که نشانگر یک بیت پایان است)

✓ **Parity**: نوع parity را مشخص می کند 0 نشانگر این است که بیت parity نداریم. 1 برای parity فرد و

2 برای parity زوج

-دستور بعدی که در این آزمایش استفاده می شود Bytes At Serial Port است که تعداد بایت اطلاعات موجود در بافر را نشان می دهد.

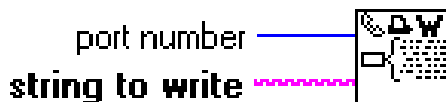


شکل ۱۱- دستور Bytes at Serial Port

✓ **Port number**: شماره پورت، که در دستور قبلی توضیح داده شد.

✓ **byte count**: عددی است که نشانگر تعداد بایت اطلاعات موجود در بافر است.

-دستور بعدی که در این آزمایش استفاده می شود Serial Port Write است که برای ارسال اطلاعات یا اصطلاحاً نوشتن بر روی پورت از آن استفاده می شود.



شکل ۱۱- دستور Serial Port Write

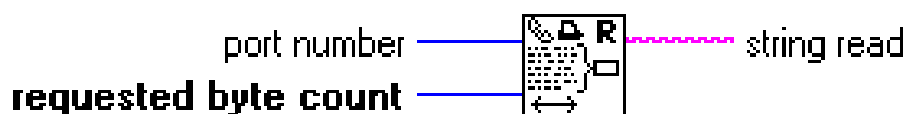
✓ **Port number**: شماره پورت، که در دستور قبلی توضیح داده شد.



✓ **string to write** : اطلاعاتی است که باید بر روی پورت ارسال شود. با ایجاد یک کنترل برای این ورودی از

دستور، یک **string control** در **front panel** ایجاد می شود.

-دستور بعدی **Serial Port Read** که برای دریافت اطلاعات یا اصطلاحاً خواندن اطلاعات بکار می رود .



شکل ۱۲- دستور **Serial Port Read**

✓ **Port number** : شماره پورت ، که در دستور قبلی توضیح داده شد .

✓ **requested byte count** : تعداد بایت اطلاعاتی که باید از بافر خوانده شود.

✓ **string read** : خروجی دستور است و با ایجاد یک **indicator** برای آن تمامی بایت اطلاعات خوانده شده قابل

مشاهده خواهد بود.

۱۲-۲- توضیحات مربوط به برنامه نوشته شده:

برنامه شامل ۴ سکانس است.

سکانس اول (شکل ۱۳) برای تنظیمات اولیه بکار رفته که در تمامی مقادیر به جز شماره پورت ، مقادیر پیش فرض

پذیرفته شده است. و برای شماره پورت یک کنترل ایجاد شده است.

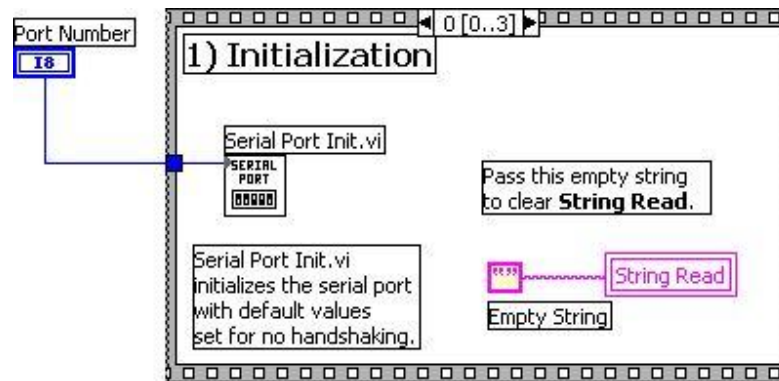
سکانس دوم (شکل ۱۴) به منظور نوشتن بر روی پورت استفاده شده است.

سکانس سوم (شکل ۱۵) از یک توقف یک ثانیه ای استفاده شده تا برنامه فرصت لازم را برای فرستادن اطلاعات بر روی

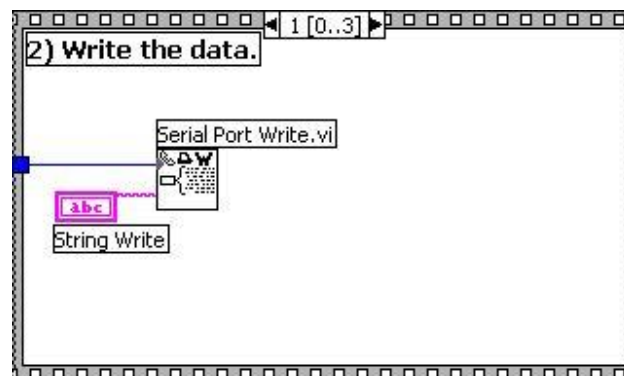
پورت را داشته باشد .



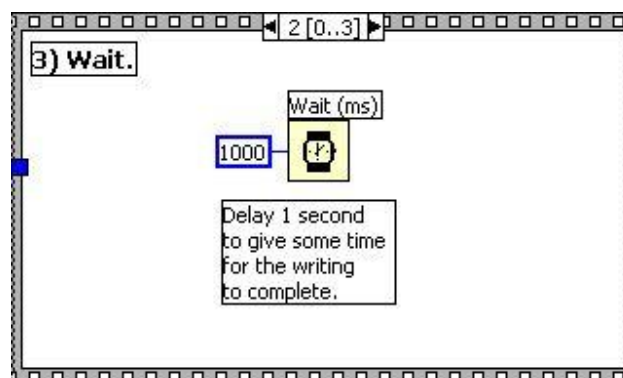
سکانس چهارم (شکل ۱۶) در این قسمت عمل خواندن از پورت سریال صورت می گیرد و تعداد بایت اطلاعاتی که باید خوانده شود توسط دستور Bytes At Serial Port تعیین می شود یعنی تمامی اطلاعات موجود در بافر پورت خوانده خواهد شد.



شکل ۱۳-سکانس اول

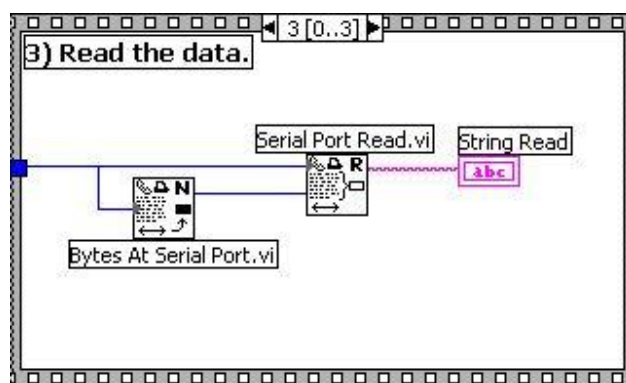


شکل ۱۴-سکانس دوم



شکل ۱۵-سکانس سوم





شکل ۱۶-سکانس چهارم