

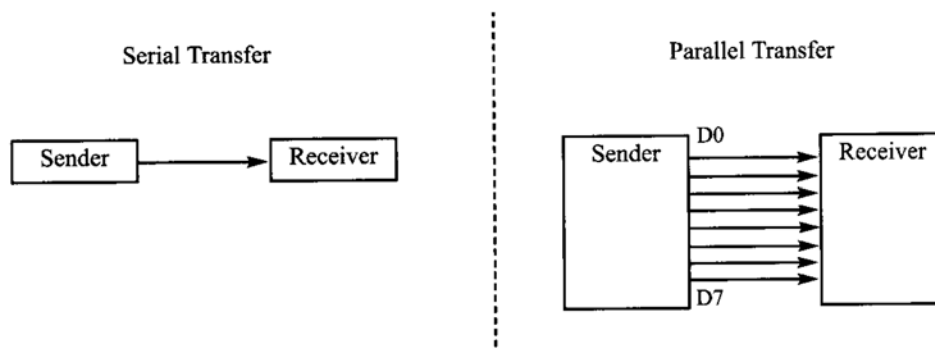
به نام خدا

آزمایشگاه ریزپردازنده دانشکده مهندسی کامپیوتر

## آزمایش شماره نه

### آشنایی با پورت سریال و نحوه استفاده از آن

موضوع این آزمایش ارتباط سریال است، بنابراین لازم است تا تعریفی از این نوع ارتباط داشته باشیم. عموماً ارتباطات به معنای مبادله اطلاعات بر مبنای یک پروتکل (یا قرارداد) است و به دو شکل موازی و سریال امکان آن وجود دارد. در نوع موازی که تا کنون از آن استفاده کرده‌ایم، تعداد بیت (معمولاً ۴، ۸، ۱۶، ...) به طور همزمان منتقل می‌شوند. در ارتباط سریال در هر لحظه تنها یک بیت منتقل می‌شود.



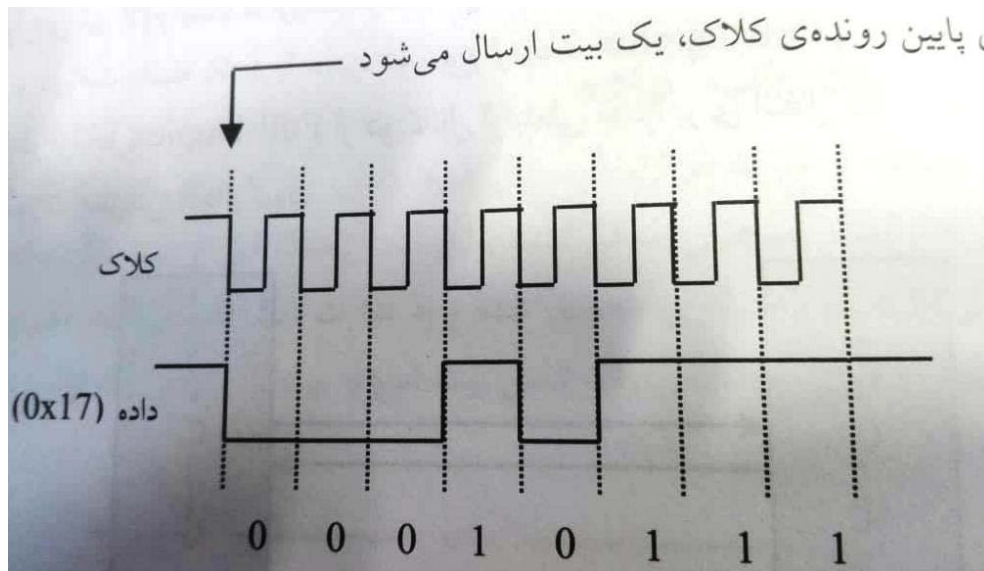
اولین چیزی که در مقایسه دو روش ارتباطی موازی و سریال به ذهن خطور می‌کند این است که در شرایط مساوی ارتباط سریال از نوع موازی کندتر بوده و هزینه‌ی آن نیز پایین‌تر است. اما باید توجه داشت که استفاده از ارتباط سریال محدودیتی ایجاد نمی‌کند چرا که کارایی یک مسیر ارتباطی تا حدود زیادی به سخت‌افزار آن وابسته است.

### آشنایی با مفهوم ارتباط سریال همزمان و غیرهمزمان

در یک ارتباط سریال، فرستنده بیت‌ها را به طور متوالی ارسال کرده و گیرنده نیز آنها را به همین ترتیب دریافت می‌کند. بنابراین برای کنترل جریان بیت‌ها نیاز به یک مرجع زمانی است که این مرجع، یک پالس کلاک است. کلاک تعیین می‌کند که فرستنده در چه زمانی یک بیت را ارسال کند و گیرنده در چه زمانی یک بیت را بخواند. دو نوع ارتباط سریال به نام‌های همزمان و غیرهمزمان وجود دارد که هر یک به شکل متفاوت از کلاک استفاده می‌کنند.

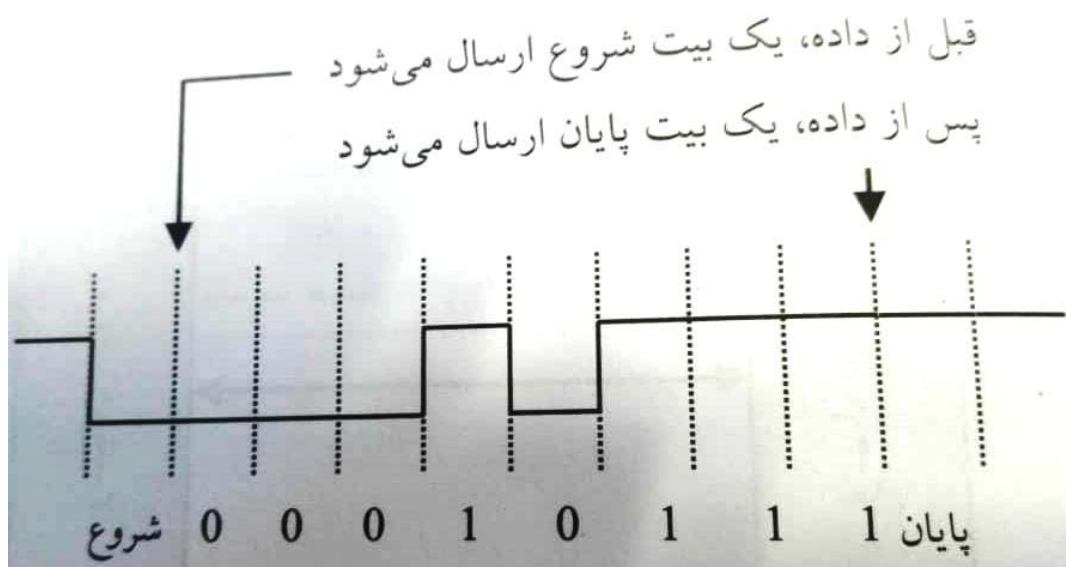
### - ارتباط سریال همزمان

در این روش هر دو وسیله دارای یک کلاک مشترک می‌باشند که این سیگنال کلاک بوسیله‌ی یکی از آنها و یا یک مولد خارجی تولید شده است. تمام بیت‌های ارسالی با کلاک همزمان شده و به عبارت دیگر در لبه‌ی بالا رونده یا پایین رونده‌ی کلاک، ارسال می‌شوند. گیرنده با استفاده از سیگنال کلاک تشخیص می‌دهد که در چه زمانی باید بیت‌های دریافتی را بخواند. به عنوان مثال در شکل زیر، رشته‌ای از پالس‌های الکتریکی شامل عدد  $0x17$  که با سیگنال کلاک همراه شده‌اند نشان داده شده است (با فرض اینکه در لبه پایین رونده‌ی سیگنال کلاک، یک بیت ارسال می‌شود).



### - ارتباط سریال غیرهمزمان

برای فواصل طولانی و یا ارتباطات بی‌سیم، استفاده از ارتباط سریال همزمان مقرون به صرفه نمی‌باشد. چرا لازم است تا سیگنال کلاک نیز منتقل شود و این کار هزینه‌بر خواهد بود. در روش غیرهمزمان، کانال ارتباطی شامل خط کلاک مجزا نبوده و هر یک از دو وسیله، کلاک را ایجاد می‌نمایند. به عنوان مثال در شکل زیر وضعیتی را مشاهده می‌کنید که داده‌ی  $0x17$  به صورت غیرهمزمان منتقل می‌شود. با فرض اینکه وضعیت خط در حالت بی‌کاری (Idle) یک منطقی است، قبل از اولین بیت داده، یک صفر به عنوان سیگنال شروع (Start Bit) درج شده تا گیرنده بتواند از طریق آن کلاک را بازیابی کند. سپس بیت‌های داده ارسال شده و گیرنده با آگاهی از عرض هر بیت، آنها را تفکیک می‌کند. پس از ارسال هشتمین بیت داده، یک منطقی به عنوان بیت پایان (Stop Bit) ارسال می‌شود.



ارتباط سریال اگر چه به تعداد سیم‌های کمتری نیاز دارد، اما گیرنده باید از عرض بیت‌های ارسالی آگاهی داشته باشد.

### آشنایی با مفهوم Simplex، Full-Duplex و Half-duplex

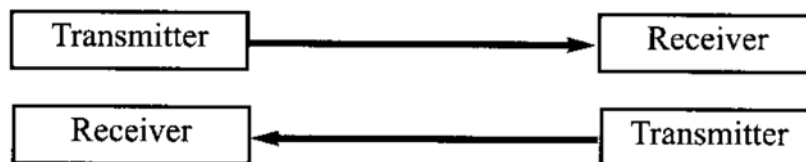
در کاربردهایی، تنها لازم است تا سیگنال‌هایی از فرستنده به گیرنده ارسال شود و فرستنده نیاز به دریافت اطلاعات از گیرنده ندارد. به این نوع ارتباط که در آن جهت تنها یک طرفه است، Simplex می‌گویند.

Simplex



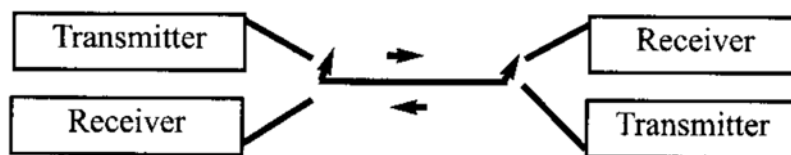
از سوی دیگر، در مواردی نیز لازم است تا ارتباط فرستنده و گیرنده به شکل دو طرفه باشد. بسته به اینکه دو وسیله در یک لحظه بتوانند ارسال و دریافت داشته باشند یا خیر، دو نوع ارتباط Full-Duplex و Half-Duplex خواهیم داشت. مطابق با شکل زیر در ارتباط Full-Duplex از دو کانال ارتباطی مجزا برای انتقال داده استفاده شده و در نتیجه ارسال و دریافت می‌تواند به صورت همزمان انجام شود.

Full Duplex



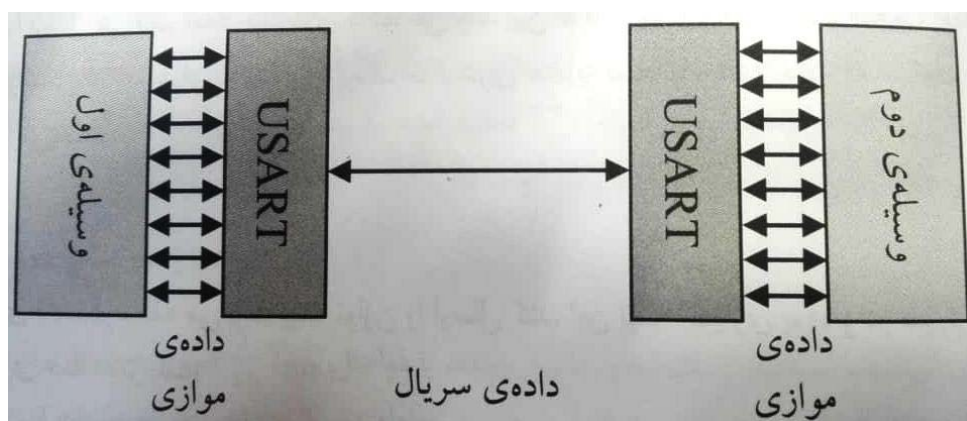
همانطور که در شکل زیر مشاهده می‌شود، در ارتباط Half-Duplex فرستنده و گیرنده امکان ارسال و دریافت را دارند. اما از آنجایی که کانال ارسال و دریافت مشترک است، انتقال نمی‌تواند در یک لحظه انجام پذیرد.

Half Duplex



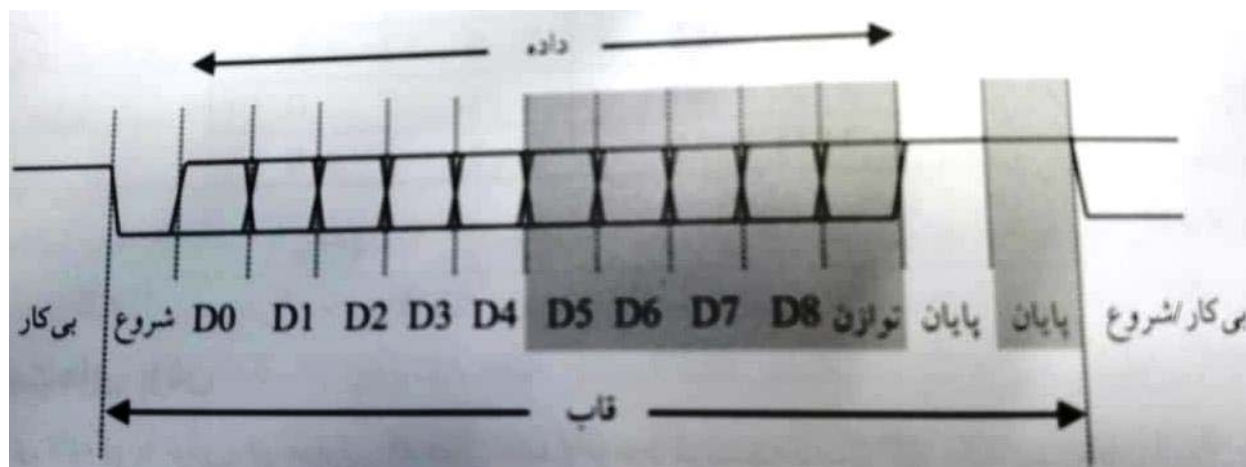
## آشنایی با مفهوم USART

USART وسیله‌ای است که امکان برقراری ارتباط سریال Full-Duplex به صورت همزمان و غیرهمزمان را بین دو وسیله فراهم می‌کند. وظیفه اصلی USART تبدیل داده موازی به سریال و بالعکس است. در طرف فرستنده، USART یک بایت را به صورت موازی دریافت کرده و بیت‌های آن را طبق یک الگوی مشخص، بیت به بیت ارسال می‌کند. در طرف گیرنده، USART دیگری وجود دارد که بیت‌ها دریافت کرده و داده‌ی اصلی را بازسازی می‌کند.



## آشنایی با اجزای یک قاب (Frame) داده

معمولاً برای همزمان سازی فرستنده و گیرنده در ارتباط سریال غیرهمزمان، نیاز به درج یک بیت شروع (Start Bit) و یک یا دوبیت پایان (Stop Bit) به ابتدا و انتهای داده‌ی خام است. همچنین ممکن است برای کنترل خطا، قبل از بیت پایان یک بیت توازن (Parity Bit) نیز درج شود. بدین ترتیب، داده‌ی اصلی با بیت‌های دیگر محصور شده و به عبارت دیگر قاب می‌شود. بنابراین یک قاب ایجاد شده توسط USART مطابق شکل زیر می‌تواند شامل بیت شروع، بیت‌های داده، بیت توازن و بیت یا بیت‌های پایان باشد که در ادامه به بررسی این اجزا می‌پردازیم.



### - بیت شروع (Start Bit)

در وضعیت بیکاری، خط در حالت منطقی یک است. با ایجاد یک لبه‌ی پایین رونده توسط فرستنده، گیرنده از شروع یک قاب اطلاعاتی آگاه شده و آماده‌ی دریافت بیت‌های داده می‌شود.

### - بیت‌های داده (Data Bits)

در برخی ارتباطات سریال ممکن است ابتدا بیت MSB و در برخی ابتدا LSB ارسال شود. اما USART همواره پس از بیت شروع LSB را ارسال می‌کند. تعداد بیت‌های داده، می‌تواند بین ۵ تا ۹ بیت متغیر باشد و انتخاب این بیت‌ها در اختیار کاربر است.

### - بیت توازن (Parity)

پس از ارسال بیت‌های داده، فرستنده می‌تواند بیت توازن را ارسال کند این بیت اختیاری بوده و در صورتی که استفاده شود، می‌تواند در آشکارسازی خطا موثر باشد. توازن بر دو نوع زوج و فرد است. در صورتی که از توازن زوج استفاده شود، باید مجموع تعداد یک‌های داده و بیت توازن عددی زوج باشد و در مورد توازن فرد نیز این مجموع باید عددی فرد باشد. به عنوان مثال در داده‌ی هشت بیتی 0b10110010، تعداد یک‌ها چهار است. بنابراین با توازن زوج، بیت توازن صفر بوده و با توازن فرد، یک خواهد بود. بیت توازن توسط USART فرستنده محاسبه شده و درج می‌گردد. در صورتی که حین انتقال (بر اثر نویز یا عوامل دیگر) یکی از بیت‌ها تغییر کند، تعداد یک‌ها با بیت توازن مغایرت داشته و توسط USART گیرنده، خطا آشکار خواهد شد.

## – بیت یا بیت‌های پایان (Stop Bit)

در امتداد بیت‌های داده (و یا توازن، در صورتی که فعال باشد)، توسط فرستنده دست کم یک بیت پایان ارسال می‌شود. بیت پایان همواره یک است. پس از بیت پایان امکان دارد تا یک قاب دیگر ارسال شود و یا برای مدتی خط در حالت بیکاری باقی بماند. در صورتی که پس از بیت پایان، بیدرنگ قاب بعدی ارسال شود، وجود این یک منطقی باعث می‌شود تا به طور قطع یک لبه‌ی پایین رونده در ابتدای قاب بعدی وجود داشته باشد که این مسئله به همزمان سازی فرستنده و گیرنده کمک می‌کند.

در صورت تمایل می‌توان تنظیمات USART را به شکلی انجام داد تا دو بیت پایان ارسال شود. این وضعیت در مواردی مفید است که لازم است تا در فاصله‌ی بین دو قاب، گیرنده زمانی کافی برای استفاده از داده‌ی قبلی را داشته باشد.

## – آشنایی با مفهوم Baud Rate

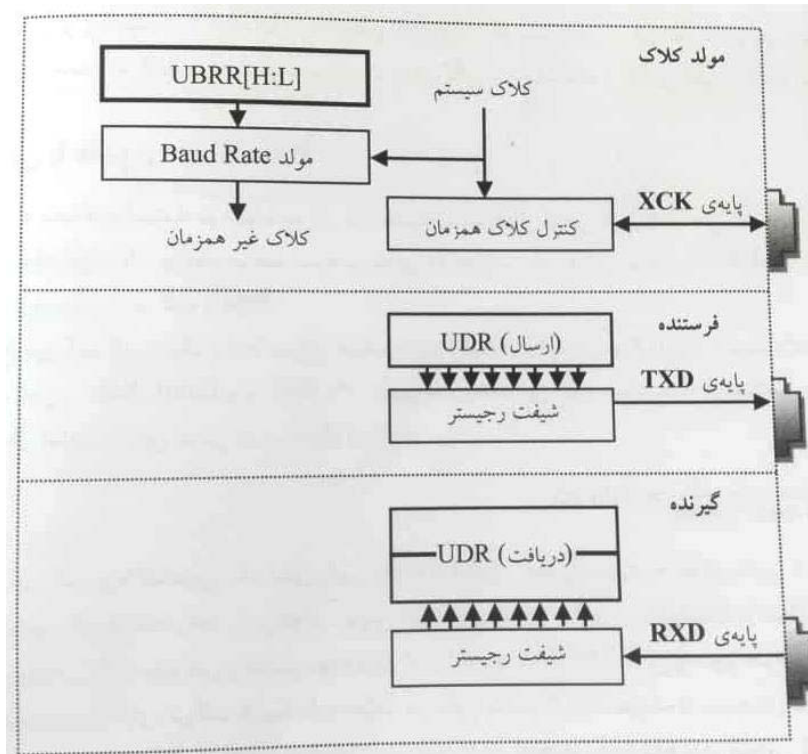
همانطور که اشاره شد تا در ارتباط سریال غیرهمزمان، گیرنده از عرض هر بیت آگاهی داشته باشد. این بدان معناست که دو طرف ارتباط باید بر یک سرعت مشخص توافق داشته باشند که به این سرعت، Baud Rate گفته می‌شود که عددی است بر حسب بیت در ثانیه (BPS یا Bit Per Second) به عنوان مثال فرض کنید که در یک ارتباط سریال غیرهمزمان، USART فرستنده و گیرنده با ۸ بیت داده، بدون بیت توازن، یک بیت پایان و Baud Rate برابر 9600bps تنظیم شده باشند. این بدان معناست که فرستنده باید ۹۶۰۰ بیت در یک ثانیه ارسال نماید، بنابراین عرض هر بیت برابر است با:

$$T_{bit} = \frac{1}{\text{baud Rate}} = \frac{1}{9600} = 104\mu S$$

USART وسیله گیرنده با تشخیص یک لبه‌ی پایین‌رونده با فواصل  $1/16 T_{bit}$  شروع به نمونه‌برداری از خط کرده و تا اتمام ۱۰۴ میکروثانیه، ۱۶ بار مقدار خط را می‌خواند. سپس نمونه‌های ۸، ۹ و ۱۰ بررسی می‌شوند، در صورتی که دو یا سه نمونه آنها صفر منطقی باشند، بیت شروع معتبر بوده و در غیر اینصورت USART آنرا نویز فرض کرده و نادیده می‌گیرد. روند نمونه برداری برای دریافت هشت بیت داده، هشت بار تکرار شده و گیرنده، هر یک از بیت‌های داده را بر اساس نمونه‌های ۸، ۹ و ۱۰ تعیین می‌کند در نهایت به همین روش بیت پایان تشخیص داده شده و در صورتی که نمونه‌های ۸، ۹ و ۱۰ برابر یک نباشند، بیت پایان معتبر نیست و USART اعلام خطای قاب (Frame Error) می‌کند.

## آشنایی با سخت افزار USART در میکروکنترلر ATmega32

در شکل زیر نمودار بلوکی ساده شده‌ی USART میکروکنترلر AVR را مشاهده می‌کنید. این واحد سخت‌افزاری که به صورت مجتمع در میکروکنترلر تعبیه شده است شامل سه بخش مولد کلاک، فرستنده و گیرنده است.



پین **XCK** (که عملکرد دوم **PB0** است) پایه تولید و یا دریافت کلاک است که بخش تولید کلاک را به بیرون ارتباط می‌دهد. این پایه تنها در وضعیت همزمان کاربرد داشته و در حالت غیرهمزمان به صورت **I/O** عمومی خواهد بود.

**USART** شامل دو بخش مجزای فرستنده و گیرنده است که هر کدام دارای یک شیفت رجیستر می‌باشند. در بخش ارسال، داده به شکل موازی از رجیستر **UDR** به شیفت رجیستر منتقل شده و پس از ایجاد یک قاب اطلاعاتی، بیت به بیت از پایه **TXD** به بیرون شیفت داده می‌شود. در بخش دریافت نیز داده‌ی سریال از پین **RXD** دریافت شده و وارد بافر **USART** می‌شود و پس از آن به رجیستر **UDR** منتقل خواهد شد. (دقت کنید که **UDR** گیرنده و فرستنده، متفاوت از یکدیگر می‌باشند.) وجود یک بافر در بخش گیرنده باعث می‌شود تا در صورت مشغول بودن **CPU**، آخرین دو بایت داده از دست نرود.



برای فعال کردن هر یک از بخش‌های فرستنده و گیرنده لازم است تا بیت فعال ساز مربوط به آن در رجیستر UCSRB یک شود.

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

بیت TXEN بیت فعال ساز بخش فرستنده USART است که با یک شدن آن، پایه PD1 از حالت I/O خارج می‌شود. بیت RXEN بیت فعال ساز بخش گیرنده USART است که با یک شدن آن، پایه PD0 ورودی USART شده و I/O نخواهد بود.

از آنجا که ما از Wizard در نرم‌افزار Code Vision استفاده می‌کنیم نیازی به آشنایی با ثباتهای دیگر نداریم و از ذکر آنها در اینجا خودداری می‌کنیم.

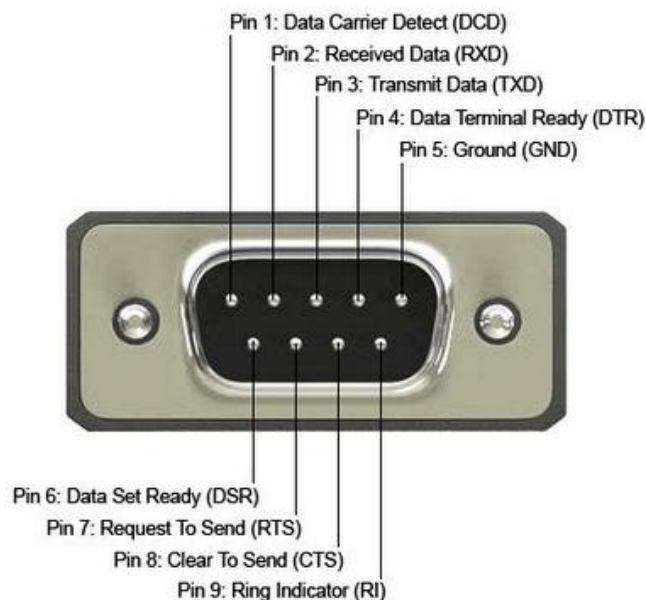
## آشنایی با استاندارد RS-232

همانطور که قبلاً گفته شد می‌توان بین دو وسیله‌ای که مجهز به USART هستند، انتقال داده انجام داد. به دلیل اهمیت چنین سخت‌افزاری، در اغلب میکروکنترلرهای امروزی واحدی به نام USART و یا UART موجود است. تفاوت این دو در این که UART بر خلاف USART تنها قابلیت ارتباط به صورت غیرهمزمان را داراست. بنابراین میکروکنترلرها از این طریق می‌توانند به سادگی با یکدیگر تبادل داده کنند. ارتباط به کمک USART منحصر به میکروکنترلرها نمی‌شود و سیستم‌هایی نظیر رایانه‌ها نیز، از واحد مشابهی برای انتقال داده به صورت سریال استفاده می‌کنند. در واقع رایانه‌ها از واحد UART سود می‌برند. که در نتیجه ارتباط بین میکروکنترلر و رایانه نیز امکان پذیر است. این امکان در بسیاری از کاربردها مفید است. از سوی دیگر به دلیل امکاناتی نظیر صفحه کلید و نمایشگر، استفاده از رایانه برای یادگیری و آزمایش برخی مفاهیم سریال مناسب است. از این رو در ادامه، از ارتباط بین میکروکنترلر و رایانه برای آزمایش کردن مطالب ارائه شده استفاده خواهیم کرد.

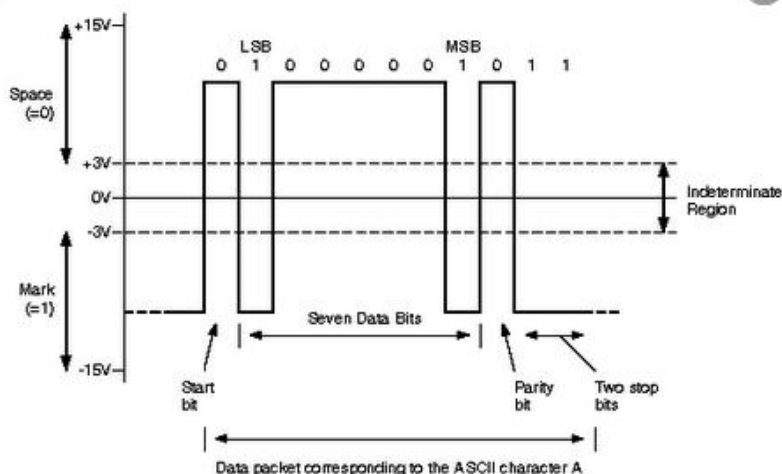
رایانه‌ها برای انتقال داده از طریق UART خود با دنیای خارج از استاندارد RS-232 استفاده می‌کنند. این استاندارد یک استاندارد سریال است. که در آن مشخصات فیزیکی ارتباط مانند سطح ولتاژ، نوع کانکتور، رفتار اتصال کوتاه و غیره تعریف شده است. رایانه‌های شخصی معمولاً دارای یک یا چند پورت سریال مبتنی بر RS-232 هستند. در شکل زیر تصویر یک کانکتور پورت سریال را مشاهده می‌کنید. به این کانکتور DB9 male گفته می‌شود.



RS232 Pinout



کانکتور DB9 دارای نه پین است که پین‌های ۲، ۳ و ۵ به ترتیب RXD، TXD و GND می‌باشند. سایر پین‌ها، مربوط به سیگنال‌های دست‌تکانی (handshaking) و عملکرد مودم می‌باشند و پایه XCK نیز در کانکتور DB9 وجود ندارد، چرا که UART عملکرد همزمان را پشتیبانی نمی‌کند. یکی از ویژگی‌های RS-232 که در ارتباط PC با میکروکنترلر اهمیت دارد سطح ولتاژ آن است. ولتاژ خط RS-232 در حالت صفر و یک منطقی با TTL متفاوت بوده و مطابق با شکل زیر است:



همانطور که در شکل ملاحظه می‌کنید سطوح منطقی RS-232 به صورت زیر است:

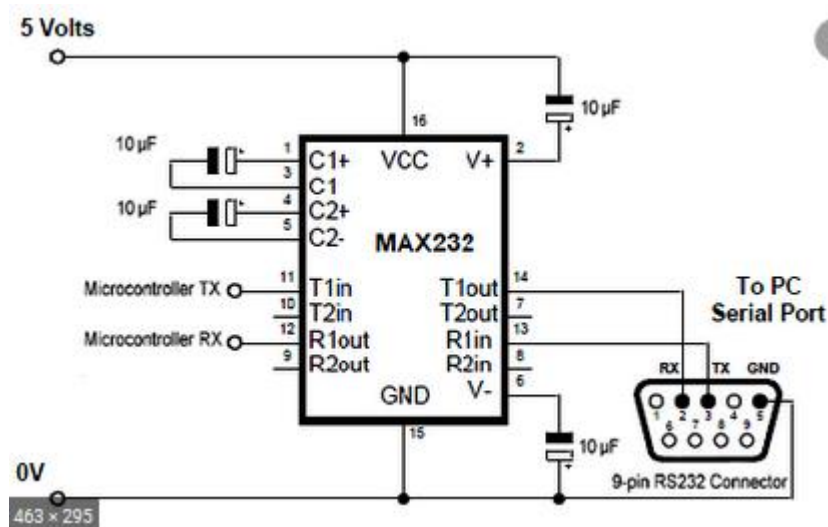
- یک منطقی بین 3- تا 15- ولت، و

- صفر منطقی بین 3+ تا 15+ ولت است

- و بین 3- تا 3+ نا معین است.

لازم به ذکر است که وسیله‌ی گیرنده باید توانایی تحمل ولتاژ اعمال شده از 25- تا 25+ ولت را داشته باشد.

از آنجا که میکروکنترلر توانایی کار با این سطوح ولتاژ را ندارد، برای ارتباط دهی پورت سریال PC (با سطح ولتاژ RS-232) با میکروکنترلر نیاز به یک مبدل ولتاژ است. اگر چه می‌توان این مبدل را به صورت گسسته با قطعات الکترونیکی طراحی کرد اما بدین منظور چندین مدار مجتمع با عملکرد مناسب وجود دارد. یکی از این ICها، قطعه MAX232 از شرکت Maxim است. MAX232 یک IC شانزده پایه است که شامل دو مبدل مجزا است. در شکل زیر یک نمونه از نحوه‌ی اتصال میکروکنترلر به پورت سریال PC را از طریق این قطعه مشاهده می‌کنید:



دقت کنید که نیازی به اتصال منبع ولتاژ منفی به این IC نیازی نیست و خود ولتاژ منفی مورد نیاز را با استفاده از خازنهای به کار رفته ایجاد می‌نماید.

با توجه به اینکه ما برای نوشتن برنامه از محیط CODEVISION استفاده می‌کنیم نیازی به معرفی رجیسترهای دیگر نیست و Wisard بقیه کار را برای ما انجام خواهد داد.

## آزمایشی که باید انجام دهید:

برنامه‌ای به زبان C بنویسید که میکروکنترلر دائماً پورت سریال را بخواند و روی صفحه نمایش LCD نمایش دهد و یک کپی از آن کاراکتر را روی ترمینال بفرستد.

- اگر کاراکتر 'C' وارد شد. صفحه نمایش LCD را پاک کند.
- اگر کاراکتر 'N' وارد شد. نام و نام خانوادگی شخص را روی ترمینال ارسال کند (البته قبل و بعد از نام شخص کاراکتر LF و CR را بفرستد تا به ابتدای خط بعد برود).
- اگر کاراکتر 'c' وارد شد. ۲۴ عدد LF و CR را روی ترمینال بفرستد.
- اگر کاراکتر 'M' وارد شد، صفحه نمایش LCD را پاک کند و نام و نام خانوادگی شخص را روی صفحه نمایش LCD بنویسد.