

## پروتکل های ارتباطی

1. ارتباط سریال و ارتباط موازی
2. پروتکل
3. پروتکل SPI
4. پروتکل UART
5. پروتکل I2C

تهیه و تنظیم :

باشگاه رباتیک باغ کتاب تهران

پاییز 1398

منابع :

•

[www.sanatbazar.com](http://www.sanatbazar.com)

[www.circuitbasics.com](http://www.circuitbasics.com)

[learn.sparkfun.com](http://learn.sparkfun.com)

• کتاب طراحی مدار های واسط کامپیوتری ( دکتر جواد راستی )

پروتکل های ارتباطی:

زمانی که یک میکروکنترلر را به یک سنسور، نمایشگر و ماژول های مختلف وصل می کنید، آیا تا به حال فکر کردید که این قطعات چگونه با یکدیگر صحبت می کنند؟ دقیقا چه چیزی به یکدیگر می گویند؟ حرف های یکدیگر را چگونه متوجه می شوند؟

ارتباط قطعات الکترونیکی دقیقا مانند ارتباط بین انسان هاست. هردو طرف باید به یک زبان صحبت کنند. در الکترونیک به این زبان ها پروتکل ارتباطی می گویند. خوشبختانه چند مورد از این پروتکل های ارتباطی از اهمیت بیشتری برخوردار بوده و در اکثر پروژه های الکترونیکی قابل پیاده سازی هستند.

Serial Peripheral Interface (SPI)

Inter-Integrated Circuit (I2C)

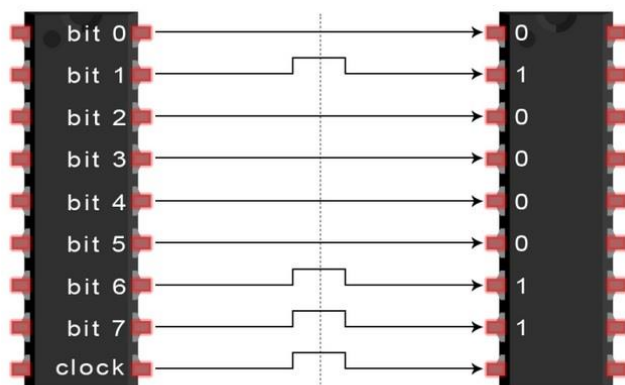
Universal Asynchronous Receiver/Transmitter (UART)

سرعت پروتکل های ارتباطی SPI، I2C و UART به مراتب از پروتکل های دیگری مانند USB، ethernet، Bluetooth و WiFi کمتر است، ولی کار با آنها ساده تر و به سخت افزارها و منابع سیستمی کمتری احتیاج دارند. SPI، I2C و UART برای ارتباط بین میکروکنترلرها و ماژول ها و سنسورهای دیگر که نیاز به انتقال اطلاعات با سرعت بالایی نیست مناسب هستند.

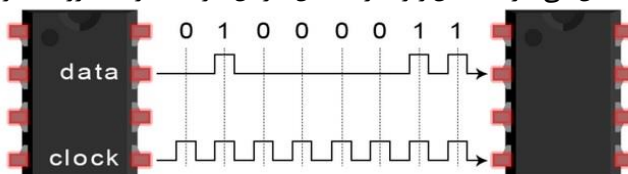
ارتباط سریال در مقابل موازی:

قطعات الکترونیکی از طریق ارسال بیت ها به واسطه سیم هایی که بین شان وجود دارد، با یکدیگر صحبت می کنند. هر بیت در واقع یک حرف از یک کلمه است، اما با 26 حرف زبان انگلیسی متفاوت است. بیت فقط می تواند مقدار 0 یا 1 را اختیار کند و از طریق تغییرات سریعی که در سطح ولتاژ به وجود می آید، از یک قطعه به قطعه ای دیگر منتقل می شوند. برای سیستمی که با منبع 5 ولت کار می کند، بیت صفر از طریق ایجاد یک پالس کوتاه صفر ولت و بیت 1 از طریق ایجاد پالس کوتاه 5 ولت ایجاد می شود.

بیت ها می توانند به صورت سریال و یا موازی بین قطعات منتقل شوند. در حالت موازی تمامی بیت های مربوط به یک داده به صورت یک جا از طریق سیم ها منتقل می شود. شکل زیر نحوه انتقال موازی حرف 'C' به صورت باینری (01000011) نشان می دهد.



در ارتباط سریال بیت ها یکی یکی و از طریق یک سیم منتقل می شوند. شکل زیر نحوه انتقال سریال حرف 'C' را به صورت باینری (01000011) نشان می دهد.



همانطور که مشاهده می کنید، ارتباط سریال سرعت پایین تری نسبت به ارتباط موازی دارد بطوریکه سرعت در ارتباط موازی می تواند تا 100 کیلوبایت در ثانیه نیز باشد، حال آنکه در ارتباط سریال حداکثر سرعت 256000 بیت بر ثانیه ( 32 کیلوبایت ) می باشد، اما در عوض نیاز به سیم های کمتری دارد و مقرون به صرفه تر است، و معمولا در فواصل طولانی از ارتباط نوع موازی استفاده نمی شود؛ بطور عادی فاصله ی دو نقطه با ارتباط موازی حداکثر می تواند تا 5 متر باشد در حالیکه ارتباط از نوع سری در محیط هایی با نویز کم می تواند تا 15 متر نیز وجود داشته باشد.

در ارتباط سریالی اگر نقطه ای از سیم قطع گردد تمام ارتباط از بین می رود اما در ارتباط موازی با قطع یک سیم یک بیت از دست می رود!

از طرف دیگر برای برقراری ارتباط سریال و فرستادن بیت ها بر روی یک خط در فرستنده به مبدل موازی به سریال و در گیرنده به مبدل سریال به موازی نیازمندیم.

در بین این ارسال و دریافت اطلاعات ممکن است اطلاعات به درستی منتقل نشوند و برای جلوگیری از این رویداد از روش هایی برای تشخیص خطا و همچنین تصحیح خطا استفاده می شود، که البته معمولا در صورت تشخیص خطا، گیرنده درخواست ارسال مجدد اطلاعات را می کند تا اینکه بخواهند خطا را تصحیح کنند.

## پروتکل<sup>۲</sup>

به مجموعه قراردادهای مابین فرستنده و گیرنده برای یک ارتباط صحیح، پروتکل گفته می شود. در پروتکل باید

موارد زیر مشخص شود:

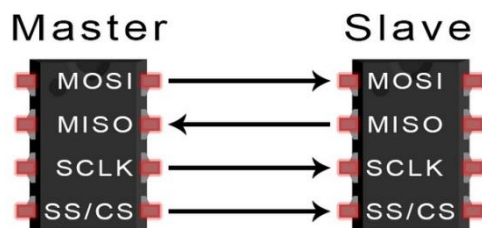
- ✓ نحوه آغاز و پایان ارتباط
- ✓ سرعت ارتباط
- ✓ شیوه کنترل جریان داده
- ✓ نوع ارتباط از دید حجم سیمهای ارتباطی (موازی یا سریال)
- ✓ جهت انتقال داده ها
- ✓ نحوه تشخیص و تصحیح خطا
- ✓ ...

## معرفی پروتکل ارتباطی SPI:

SPI یک پروتکل ارتباطی بوده که توسط بسیاری از ماژول ها استفاده می شود. ماژول SD کارت، ماژول RFID card reader و وایرلس 2.4 گیگاهرتزی گیرنده/فرستنده نمونه هایی از این موارد هستند.

ابتدایی ترین تعریف از SPI انتقال داده ها بدون وقفه می باشد. هر تعداد از بیت ها می توانند به طور پیوسته فرستاده و یا دریافت شوند. در ارتباط I2C و UART داده ها به صورت بسته هایی که به بیت های مشخصی محدود شده، فرستاده می شوند. شرایط شروع و توقف، ابتدا و انتهای بسته ها را مشخص می کند، لذا در طول انتقال داده ها وقفه هایی وجود دارد.

ارتباط قطعات از طریق SPI بر اساس رابطه master - slave (استاد و برده) می باشد. master قطعه کنترل کننده (معمولا میکروکنترلر) و slave (معمولا یک سنسور، نمایشگر یا تراشه حافظه) دستورات را از master دریافت می کند. در ساده ترین روش ارتباط SPI یک master و یک slave وجود دارد، در حالی که این امکان هم هست که یک master چند slave را کنترل کند.



MOSI (Master Output/Slave Input) = خطی برای ارسال داده ها از master به slave

MISO (Master Input/Slave Output) = خطی برای ارسال داده ها از slave به master

SCLK (Clock) = خطی مربوط به سیگنال کلاک

SS/CS (Slave Select/Chip Select) = خطی برای master برای اینکه انتخاب کند داده به کدام slave منتقل شود.

Wires Used	4
Maximum Speed	Up to 10 Mbps
Synchronous or Asynchronous?	Synchronous
Serial or Parallel?	Serial
Max # of Masters	1
Max # of Slaves	Theoretically unlimited*

در عمل تعداد slave ها با توجه به میزان ظرفیت بارهایی که به سیستم متصل هستند، محدود می شود. چون باعث کاهش دقت master در سوئیچ کردن بین سطح های ولتاژ خواهد شد.

## SPI چگونه کار می کند:

### کلاک ساعت:

سیگنال کلاک باعث همزمانی داده های خروجی از master و داده های نمونه از slave می شود. به ازای هر کلاک یک بیت داده منتقل خواهد شد، پس سرعت انتقال اطلاعات به فرکانس سیگنال کلاک بستگی دارد. مقدار دهی اولیه و فرکانس سیگنال کلاک در ارتباط SPI توسط master تعیین می شود.

پروتکل های ارتباطی که از سیگنال کلاک استفاده می کنند، همزمان (synchronous) خوانده می شوند که ارتباط SPI نیز از این نوع می باشد. پروتکل های ارتباطی دیگر که از سیگنال کلاک استفاده نمی کنند، ناهمزمان (asynchronous) خوانده می شوند. برای مثال در ارتباط UART هر دو سمت رابطه از یک نرخ انتقال اطلاعات (baudrate) یکسان برای انتقال داده ها استفاده می کنند.

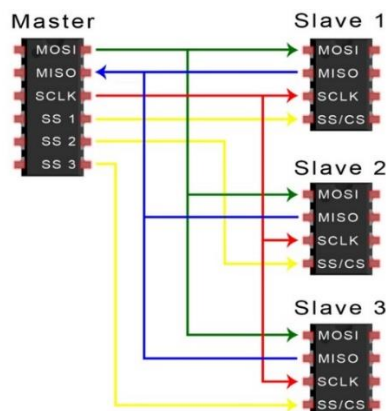
سیگنال کلاک در ارتباط SPI توسط تنظیمات clock polarity و clock phase در دسترس است. از این دستورات برای تعیین اینکه بیت ها از نوع خروجی و یا نمونه ی داده می باشند، استفاده می شوند. Master از دستور clock polarity برای تعیین اینکه بیت ها در لبه بالا رونده و یا پایین رونده به صورت خروجی و داده نمونه باشد، استفاده می کند. از دستور clock phase جهت تعیین اینکه داده خروجی و نمونه در لبه بالا رونده و یا پایین رونده اول و یا دوم، بدون توجه به سطح آنها، اتفاق بیافتند، استفاده می شود.

### انتخاب slave:

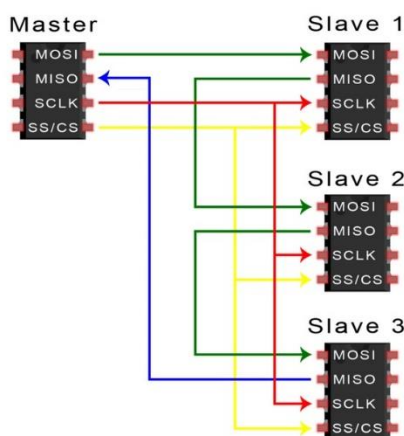
Master از طریق قرار دادن خط انتخاب مربوط به (CS/SS) slave در حالت LOW، می تواند انتخاب کند که با کدام slave می خواهد ارتباط برقرار کند. و اگر این خط در حالت HIGH قرار داشته باشد، هیچ گونه انتقال اطلاعاتی صورت نخواهد گرفت. در صورت وجود چند پین CS/SS تمامی slave ها به صورت موازی به master متصل خواهند شد و در صورتی که فقط یک پین CS/SS وجود داشته باشد، اتصال از طریق یک سیم و به صورت شاخه ای خواهد بود.

### وجود چند slave:

همان طور که گفته شد در پروتکل SPI امکان برقراری ارتباط یک master با یک slave و یا یک master با چند slave وجود دارد. دو راه برای اتصال چند slave به یک master وجود دارد، در صورتی وجود چندین پین انتخاب slave، اتصالات مطابق شکل زیر است:



و در صورتی که فقط یک پین برای انتخاب slave وجود داشته باشد، اتصالات شاخه ای به صورت زیر خواهد بود:



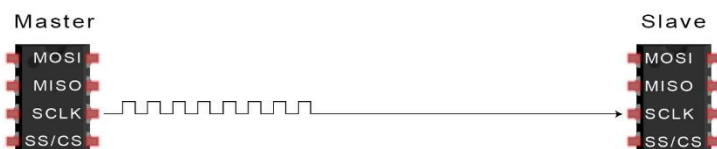
### MOSI و MISO:

داده ها به صورت بیت به بیت و سریال از طریق پین MOSI از master به slave فرستاده می شوند. Slave نیز این داده ها را از طریق پین MOSI دریافت می کند. در این حالت معمولا بیت های با ارزش (MSB) زودتر فرستاده می شوند.

Slave نیز داده های خود را از طریق پین MISO به master می فرستد. در این حالت معمولا بیت های کم ارزش (LSB) زودتر فرستاده می شوند.

### مراحل انتقال داده از طریق پروتکل SPI:

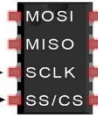
1 - master سیگنال کلاک را ارسال می کند.



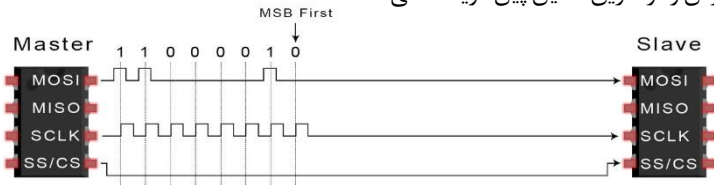
Master



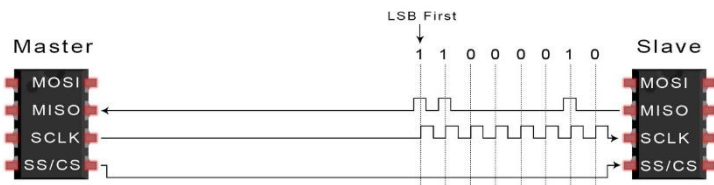
2 - master جهت فعال کردن slave پین SS/CS را در حالت LOW قرار می دهد Slave



3 - master داده ها را به صورت بیت به بیت از طریق پین MOSI ارسال کرده و slave نیز آن را از طریق همین پین دریافت می کند.



4 - در صورتی که به اطلاعات slave احتیاج باشد، داده ها به صورت بیت به بیت از طریق پین MISO به master فرستاده خواهند شد:



### مزایا و معایب SPI :

با توجه به مزایا و معایب این پروتکل و مقایسه با پروتکل های دیگر شما می توانید انتخاب درست را در پروژه های خود داشته باشید:

مزایا:

- بیت شروع و پایانی وجود ندارد، لذا داده ها بدون وقفه منتقل خواهند شد.
- سیستم پیچیده آدرس دادن slave مانند I2C وجود ندارد.
- سرعت بالاتر انتقال اطلاعات نسبت به I2C (تقریباً دو برابر)
- با توجه به دو خط جدا برای MOSI و MISO امکان دریافت و ارسال داده به صورت هم زمان وجود دارد.

معایب:

- استفاده از چهار سیم (در صورتی که I2C و UART از دو سیم استفاده می کنند)
- هیچ علامتی مبنی بر دریافت صحیح داده وجود ندارد (I2C این امکان را دارد)
- هیچ گونه روش تشخیص خطا ندارد مانند پیریتی در UART
- فقط امکان استفاده از یک master وجود دارد.

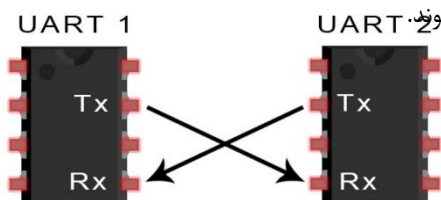
حتماً به خاطر می آورید که پرینترها، موس و مودم ها دارای چه کابل های ضخیم و کانکتورهای بزرگی برای اتصال به کامپیوتر بودند. تمامی این قطعات از روش UART برای برقراری ارتباط با کامپیوتر استفاده می کردند. اگر چه USB تقریباً به طور کامل جایگزین کانکتورهای قدیمی شده است، اما این به معنی کنار گذاشتن روش UART نیست.

از UART در بسیاری از پروژه های الکترونیکی جهت اتصال ماژول های GPS ، بلوتوث و RFID به رزپیری پای، آردوینو و سایر میکروکنترلرها استفاده می شود.

UART یا Universal Asynchronous Receiver/Transmitter یک پروتکل ارتباطی مانند SPI و I2C نمی باشد، بلکه یک مدار فیزیکی است در میکروکنترلر و یا به صورت یک IC. هدف اصلی UART ارسال و دریافت داده ها به صورت سریال می باشد.

### معرفی ارتباط UART:

در ارتباط UART دو دستگاه به طور مستقیم به یکدیگر متصل می شوند. دستگاه ارسال کننده داده های موازی را که از یک بخش کنترلی مانند CPU دریافت می کند، به صورت سریال تبدیل کرده و برای دستگاه دریافت کننده ارسال می کند. سپس در سمت دریافت کننده داده های سریال به صورت موازی تبدیل می شوند. داده ها از پین TX دستگاه ارسال کننده برای پین RX دستگاه دریافت کننده فرستاده می شوند.



UART داده ها را به صورت ناهمزمان ارسال می کند. به این معنی که سیگنال کلاک برای همزمان کردن داده های خروجی دستگاه ارسال کننده با داده های نمونه دستگاه دریافت کننده، وجود ندارد. به جای سیگنال کلاک از بیت های شروع، توازن (parity) و پایان در بسته داده های ارسالی استفاده می کند، لذا دستگاه دریافت کننده متوجه می شود که چه زمانی شروع به خواندن داده ها کند.

زمانی که دستگاه دریافت کننده متوجه بیت شروع شد، داده های ورودی را با فرکانس مشخصی که به آن نرخ انتقال داده (baud rate) می گویند، دریافت می کند. baud rate تعیین کننده سرعت انتقال اطلاعات بوده و برحسب بیت بر ثانیه (bps) سنجیده می شود. هر دو دستگاه UART باید با baud rate یکسان عمل کرده و حداکثر میزان اختلاف آنها می تواند 10 % باشد.

تعداد سیم های مورد استفاده = 2

حداکثر سرعت = baud 115200 و به طور معمول baud 9600

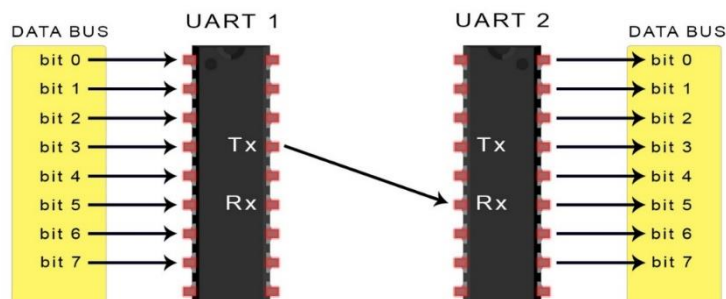
همزمان و یا ناهمزمان = ناهمزمان

حداکثر تعداد master (کنترل کننده) = 1

حداکثر تعداد slave (کنترل شونده مثل سنسورها و ماژول ها و...) = 1

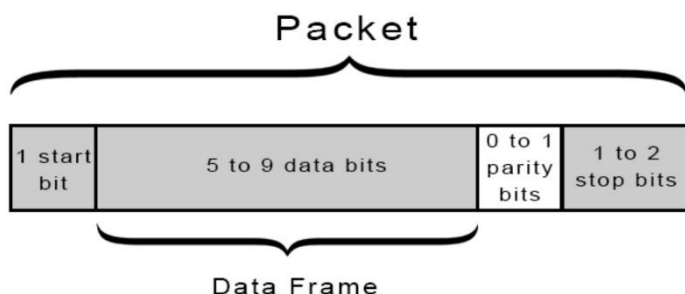
### UART به چه صورت کار می کند:

دستگاه ارسال کننده UART داده ها را از گذرگاه داده (data bus) دریافت می کند. گذرگاه داده برای ارسال داده ها به UART توسط قطعاتی مانند CPU، حافظه و یا میکروکنترلر استفاده می شود. داده ها به صورت موازی از گذرگاه داده به UART فرستاده شده، دستگاه ارسال کننده با اضافه کردن بیت های شروع، توازن و پایان یک بسته داده ایجاد می کند. در ادامه این بسته به صورت سریال و بیت به بیت از طریق پین TX ارسال می شود. دستگاه دریافت کننده داده های سریال را از طریق پین RX دریافت و به حالت موازی تبدیل کرده و بیت های شروع، تعادل و پایان را حذف می کند. در انتها این داده ها به گذرگاه داده در دستگاه دریافت کننده فرستاده می شوند.





همان طور که گفته شد داده ها در ارتباط UART به صورت یک بسته فرستاده می شوند. هر بسته شامل یک بیت شروع، 5 تا 9 بیت داده به UART بستگی دارد، یک بیت توازن قابل تنظیم و یک یا دو بیت پایانی.



### بیت شروع:

خط انتقال اطلاعات از UART زمانی که از آن استفاده نمی شود، در حالت HIGH می گیرد. برای شروع انتقال اطلاعات این خط برای مدت یک کلاک ساعت در وضعیت LOW قرار می گیرد. زمانی که دستگاه دریافت کننده یک تغییر سطح ولتاژ از حالت HIGH به LOW را تشخیص داد، شروع به دریافت داده ها با baud rate مشخص می شود.

### ساختار داده ها (data frame):

در این قسمت داده اصلی قرار گرفته و می تواند از 5 تا 8 بیت، در صورتی که از بیت توازن استفاده شود، را شامل شود. و در صورتی که بیت تعادل به کار نرود، می تواند تا 9 بیت باشد. در اکثر اوقات ابتدا بیت کم ارزش (LSB) فرستاده می شود.

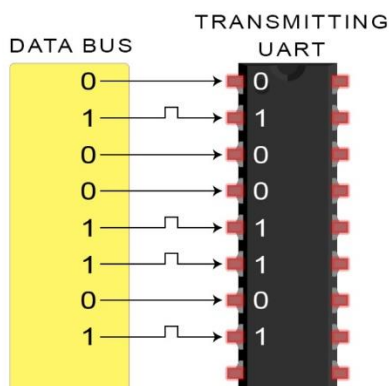
### بیت توازن:

این بیت ها معمولاً توازن زوج و یا فرد در داده ها به وجود می آورند. از بیت توازن برای تشخیص وجود خطا در داده های ارسالی UART استفاده می شود. بیت های ارسالی ممکن است به دلایلی مانند: امواج الکترومغناطیسی، عدم تطابق baud rate و یا فاصله طولانی بین دستگاه ها به وجود آید. زمانی که داده دریافت شد، تعداد یک های موجود در آن با توجه به توازن زوج و یا فرد انتخاب شده، بررسی می شود. در صورتی که بیت توازن صفر باشد (توازن زوج)، در این صورت تعداد یک ها باید زوج، و در صورتی که بیت توازن یک باشد (توازن فرد) تعداد یک ها باید فرد باشد. از این طریق وجود و یا عدم وجود خطا در داده ارسالی بررسی می شود.

### بیت پایانی:

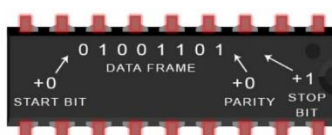
برای مشخص کردن داده پایانی، دستگاه فرستنده خط انتقال داده را برای مدت حداقل دو بیت از حالت LOW به حالت HIGH منتقل می کند.

### مراحل انتقال داده در UART:



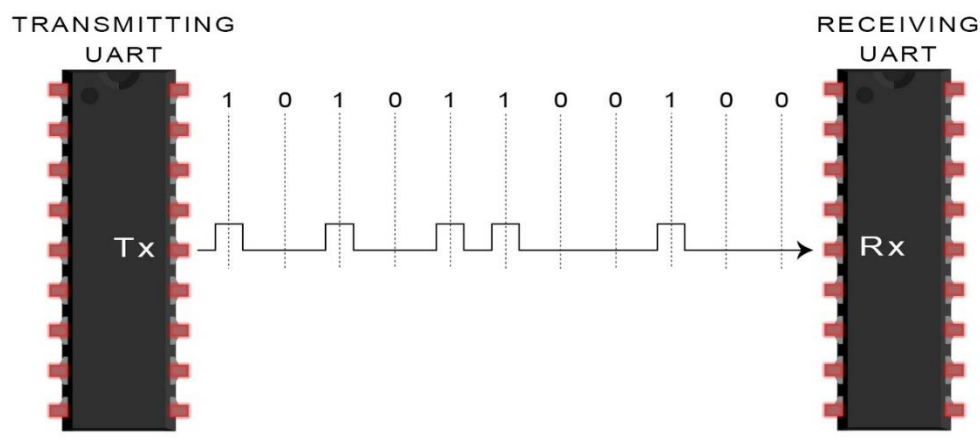
1- دستگاه ارسال کننده داده ها را به صورت موازی از گذرگاه داده دریافت می کند.

2- دستگاه ارسال کننده بیت شروع، توازن و پایانی را به داده اضافه می کند.

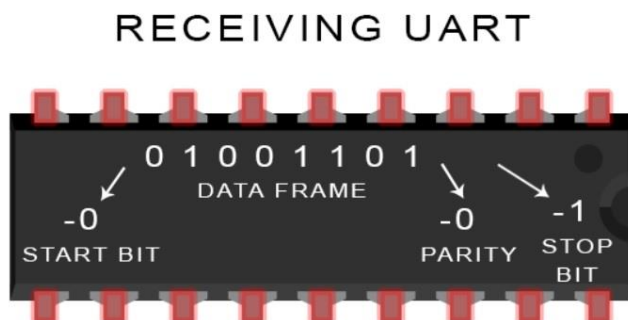




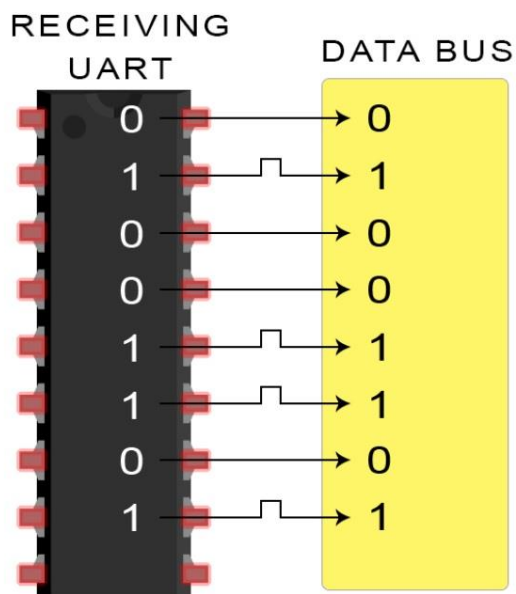
3- بسته داده به صورت سریال و با توجه به سرعت baud rate از دستگاه ارسال کننده فرستاده می شود.



4- دستگاه دریافت کننده بیت شروع، توازن و پایانی را از داده حذف می کند.



5- دستگاه دریافت کننده داده سریال را به موازی تبدیل کرده و به گذرگاه داده ارسال می کند.



## مزایا و معایب روش UART:

عدم وجود پروتکل ارتباطی جالب به نظر می رسد ولی UART با توجه به نحوه عملکردش بسیار مناسب و کاربردی است.

مزایا:

- استفاده از فقط دو سیم
- ضرورتی به استفاده از سیگنال کلاک وجود ندارد
- وجود بیت توازن برای بررسی خطا
- ساختار بسته های داده با توجه به تنظیمات دو سمت قابل تغییر است
- در اختیار بودن مستندات و اطلاعات مفید و روش های گسترده استفاده

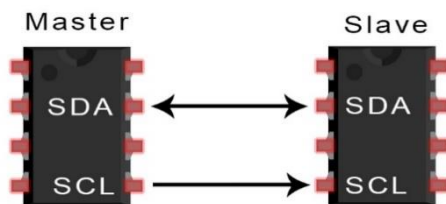
معایب:

- اندازه داده های ارسالی حداکثر به 9 بیت محدود شده
- عدم پشتیبانی از چند master (کنترل کننده) و چند slave (کنترل شونده)
- میزان baud rate دو سمت حداکثر می توانند 10% با یکدیگر اختلاف داشته باشند.

I2C :

بعد از بررسی ارتباطات SPI و UART در این مقاله آخرین پروتکل یعنی I2C (the inter-integrated circuit) مورد بررسی قرار می گیرد. قطعاتی مانند نمایشگرهای قدیمی، سنسورهای فشار بارومتری، ژيروسکوپ و شتاب سنج از این پروتکل استفاده می کنند.

پروتکل I2C در واقع ادغام شده از بهترین ویژگی های SPI و UART می باشد. توسط I2C امکان اتصال چند slave به یک master (مانند SPI) و یا استفاده از چند master برای کنترل یک یا چند slave وجود دارد. این ویژگی زمانی که شما می خواهید از چند میکروکنترلر برای ارسال داده به یک کارت حافظه و یا نمایش بر روی LCD استفاده کنید، بسیار مناسب می باشد.



مانند روش UART در I2C نیز از دو سیم برای انتقال اطلاعات استفاده می شود.

SDA (Serial Data) = خطی برای ارسال و دریافت داده بین master و slave

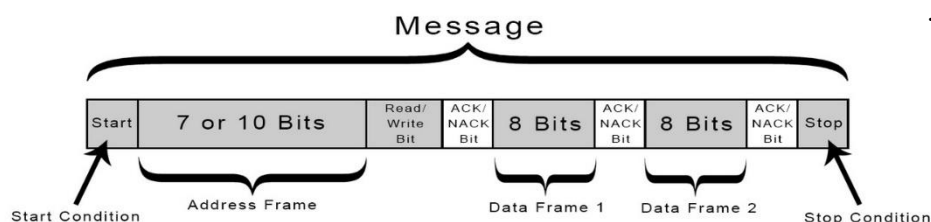
SCL (Serial Clock) = خطی که حامل سیگنال کلاک می باشد.

I2C یک پروتکل ارتباطی سریال می باشد، لذا داده ها به صورت بیت به بیت از طریق خط SDA منتقل خواهند شد.

همانند SPI، پروتکل I2C نیز به صورت همزمان می باشد، لذا بیت های خروجی با بیت های نمونه توسط سیگنال کلاک همزمان می باشند. سیگنال کلاک توسط master کنترل می شود.

## نحوه عملکرد I2C:

در I2C داده ها به صورت پیام هایی فرستاده شده و پیام ها نیز خود به چند بخش تقسیم می شوند. هر پیام شامل یک بخش آدرس بوده، که همان آدرس باینری مربوط به slave است. یک یا دو بخش مربوط به داده نیز در پیام های ارسالی وجود دارد. پیام همچنین شامل شرایط اولیه و پایانی، بیت های read/write و بیت های ACK / NACK بین هر بخش از پیام می شود.



## معرفی پیام، بخش، و بیت های I2C:

شرایط شروع: تغییر خط SDA از حالت high به حالت low، قبل از تغییر خط SCL به همین صورت

شرایط پایان: تغییر خط SDA از حالت low به حالت high، بعد از تغییر خط SCL به همین صورت

بخش آدرس: ارسال 7 تا 10 بیت با توالی منحصر به فرد به هر slave جهت تشخیص توسط master

بیت read / write: وجود یک بیت هنگامی که داده از طرف master برای slave (ولتاژ سطح low) و بر عکس (ولتاژ سطح high) فرستاده می شود.

بیت ACK / NACK: هر بخش پیام شامل یک بیت تصدیق کننده / عدم تصدیق کننده می باشد. در صورتی که بخش آدرس پیام یا بخش داده ی پیام به درستی دریافت شود، یک بیت ACK از طرف دستگاه دریافت کننده به فرستنده ارسال می گردد.

## آدرس دادن:

I2C برخلاف SPI دارای خطی برای انتخاب slave مورد نظر نیست. بنابراین برای اینکه هر slave تشخیص دهد که کدام داده برای آن ارسال شده، از روش آدرس دهی استفاده می شود. بخش آدرس همیشه اولین بخش بعد از بیت شروع در هر پیام می باشد.

Master آدرس slave مورد نظر برای اتصال را ارسال می کند. سپس هر کدام از slave ها این آدرس را با آدرس خود مقایسه می کنند. در صورت همخوانی آدرس، یک بیت سطح پایین ACK برای master فرستاده می شود. در صورتی که آدرس هم خوانی نداشته باشد، اتفاقی نیافتاده و خط SDA در وضعیت high باقی می ماند.

## بیت read / write:

بخش آدرس در انتهای خود شامل یک بیت بوده که به slave اطلاع می دهد، که master قصد خواندن اطلاعات از آن و یا ارسال اطلاعات به آن را دارد. در صورتی که master قصد ارسال اطلاعات داشته باشد، این بیت در وضعیت low و در صورتی که در قصد خواندن داشته باشد، این بیت در وضعیت high قرار خواهد گرفت.

## بخش داده:

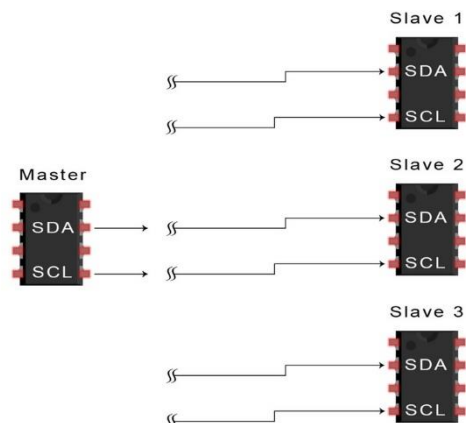
بعد از دریافت بیت ACK توسط master از طرف slave، اولین بخش داده آماده ارسال می باشد.

بخش داده همیشه شامل 8 بیت بوده و بیت های با ارزش تر (MSB) در ابتدا فرستاده می شوند. این بخش ها بلافاصله با بیت ACK / NACK همراه شده تا تشخیص داده شود، که داده به درستی دریافت شده است. بیت ACK قبل از اینکه داده بعدی ارسال گردد، باید توسط master و یا slave دریافت شود (بستگی دارد به اینکه کدام یک داده را ارسال کرده باشند).

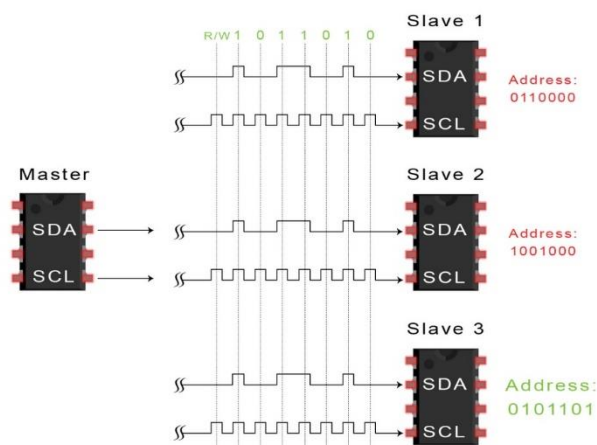
بعد از ارسال تمامی بخش های پیام، master می تواند شرایط توقف را به slave ارسال کند تا انتقال اطلاعات متوقف شود. شرایط پایانی در واقع تغییر ولتاژ خط SDA از حالت low به high، بعد از تغییر خط SCL به همین صورت و باقی ماندن در وضعیت high می باشد.

## مراحل انتقال داده از طریق پروتکل I2C:

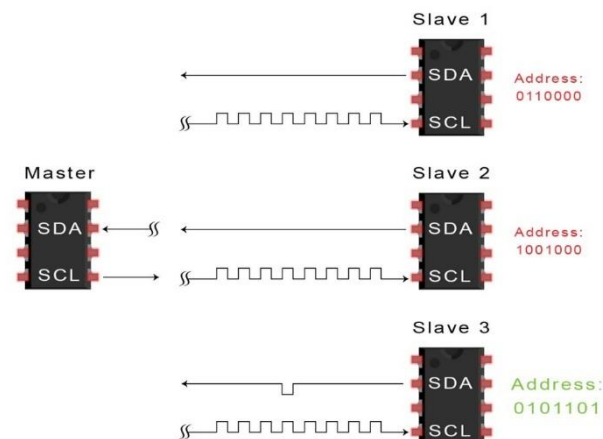
1- شرایط شروع از طرف master به تمامی slave هایی که به آن متصل هستند، ارسال می گردد. به این صورت که ابتدا خط SDA و سپس خط SCL از حالت low به high تغییر می کند.

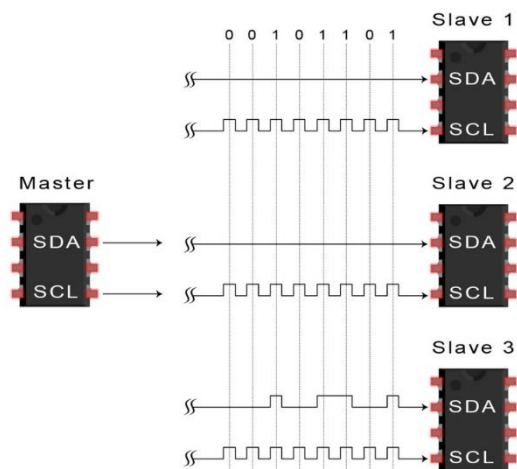


2- از طریق بیت read/write آدرس 7 یا 10 بیتی توسط master برای slave ها فرستاده می شود، تا هدف مورد نظر پیدا تشخیص داده شود.



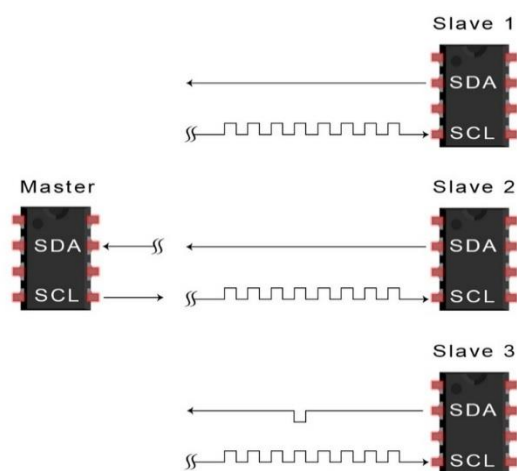
3- هر کدام از slave ها آدرس فرستاده شده را با آدرس خود مقایسه می کنند. در صورت همخوانی آدرس، slave مورد نظر بیت ACK را از طریق قرار دادن خط SDA در حالت low برای مدت بیت، ارسال می کند. و در صورتی که آدرس همخوانی نداشته باشد، خط SDA در حالت high ثابت باقی خواهد ماند.



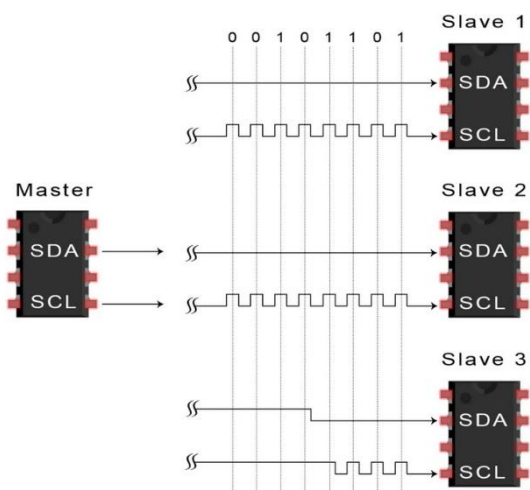


4- Master بسته داده را ارسال و یا دریافت می کند.

5- بعد از ارسال هر کدام از بسته های داده، دستگاه دریافت کننده یک بیت ACK به فرستنده ارسال می کند، تا از دریافت موفق داده اطلاع دهد.

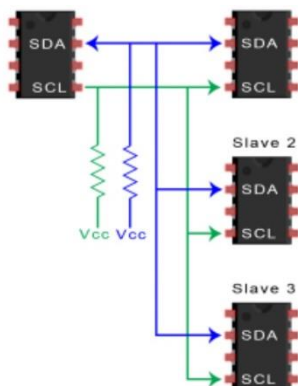


6- برای توقف انتقال داده ها، master شرایط توقف را به slave ارسال می کند، به این صورت که ابتدا خط SCL و سپس خط SDA را از حالت low به حالت high تغییر می دهد.



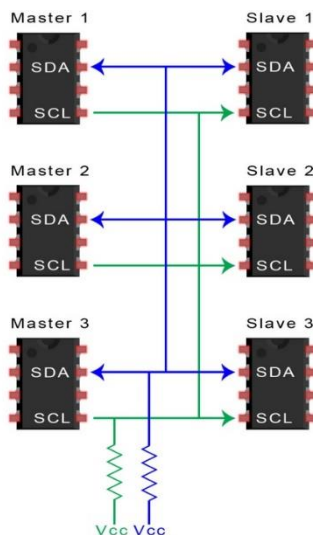
استفاده از یک master جهت کنترل چند slave:

چون I2C از روش آدرس دهی استفاده می کند، امکان کنترل چند slave از طریق یک master وجود دارد. از طریق آدرس 7 بیتی،  $2^7$  (128) آدرس مختلف به وجود می آید. استفاده از آدرس دهی 10 بیتی رایج نمی باشد، ولی در این حالت هم  $2^{10}$  (1024) آدرس متفاوت به وجود می آید. نحوه اتصال در این روش را نیز در شکل زیر مشاهده می کنید، که مقاومت های 4.7 کیلو اهم pull-up از خط های SDA و SCL به VCC متصل می شوند:



استفاده از چند master برای کنترل چند slave:

امکان اتصال چند master به یک و یا چند slave وجود دارد. در این حالت مشکل زمانی به وجود می آید که دو master بخواهند به طور هم زمان از یک خط SDA داده را ارسال و یا دریافت کنند. برای حل این مشکل هر کدام از master ها قبا از ارسال پیام باید تشخیص دهند که خط SDA در حالت low قرار دارد و یا حالت high. در صورتی که خط SDA در حالت low باشد، به این معناست که گذرگاه داده توسط master دیگری در حال استفاده می باشد، لذا باید تا پایان ارسال داده منتظر ماند. و در صورتی که خط SDA در حالت high باشد، به معنای آزاد بودن خط است و امکان ارسال داده وجود دارد. نحوه اتصالات در این روش هم در شکل زیر آمده است که از خط های SDA و SCL با مقاومت 4.7 کیلو اهم به VCC متصل شده اند:



### مزایا و معایب روش I2C:

دلایل زیادی وجود دارد که امکان مقایسه روش I2C را با پروتکل های دیگر سخت می کند، ولی می توان از میان آنها دلایل خوبی برای انتخاب و یا عدم انتخاب I2C پیدا کرد:

مزایا:

- استفاده از دو سیم
- پشتیبانی کردن از چند master و چند slave
- اطلاع از انتقال صحیح داده ها توسط بیت های ACK/NACK
- ساختار سخت افزاری ساده تر نسبت به روش UART
- پروتکل شناخته شده و پرکاربرد

معایب:

- سرعت پایین تر انتقال اطلاعات نسبت به روش SPI
- حداکثر طول داده 8 بیت می باشد.
- تجهیزات سخت افزاری مشکل تر برای پیاده سازی نسبت به روش SPI