

به نام خدا

گزارش کار پروژه ۲

اعضای گروه:

۸۱۰۱۹۶۶۰۶ کیمیا خبیری

۸۱۰۱۹۶۴۹۱ طاهّا شعبانی

۸۱۰۱۹۶۶۲۹ زهرا موسوی موحد

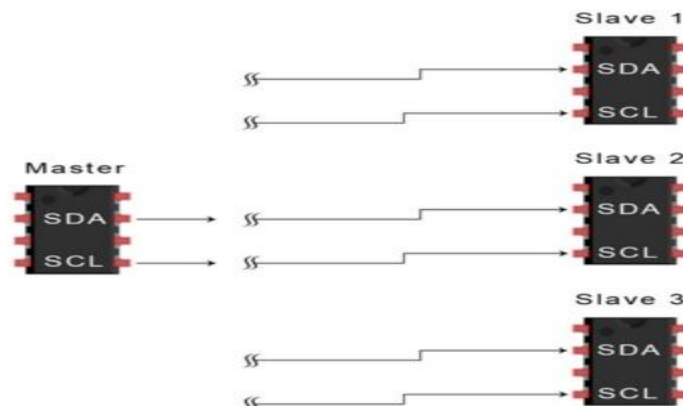
۱- پروتکل I2C قابلیت اتصال multi masters - multi slaves را دارد. توضیح دهید مشکلات پیش روی این

حالت چیست؟ و این پروتکل چگونه آن را حل کرده است؟

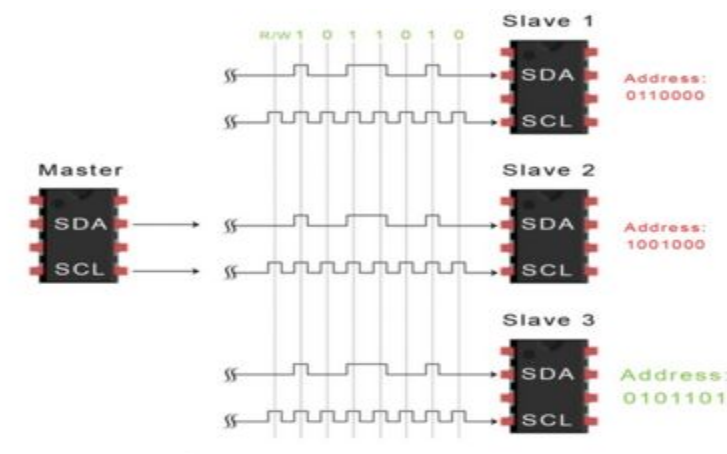
مشکل زمانی به وجود می آید که دو master بخواهند به طور همزمان از یک خط SDA داده را ارسال و یا دریافت کنند. برای حل این مشکل هر کدام از master ها قبل از ارسال پیام باید تشخیص دهند که خط SDA در حالت low قرار دارد و یا حالت high. در صورتی که خط SDA در حالت low باشد، به این معناست که گذرگاه داده توسط master 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.

۲- مراحل ارتباط دو دستگاه به یکدیگر از طریق این پروتکل (I2C) را بیان کنید.

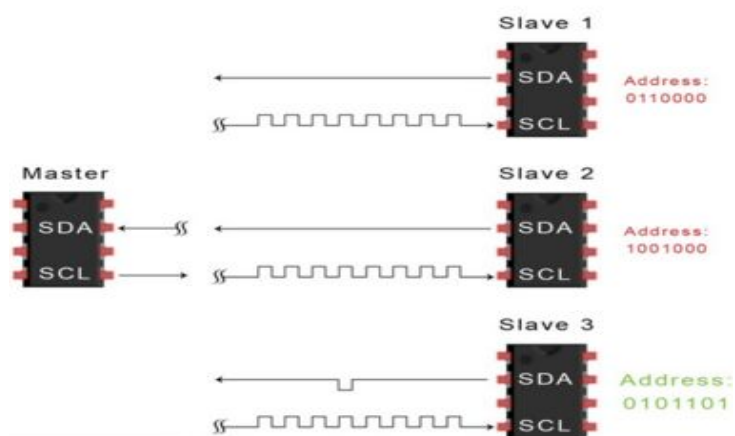
۱- شرایط شروع از طرف master به تمامی slave هایی که به آن متصل هستند، ارسال می گردد. به این صورت که ابتدا خط SDA و سپس خط SCL از حالت low به high تغییر می کند.



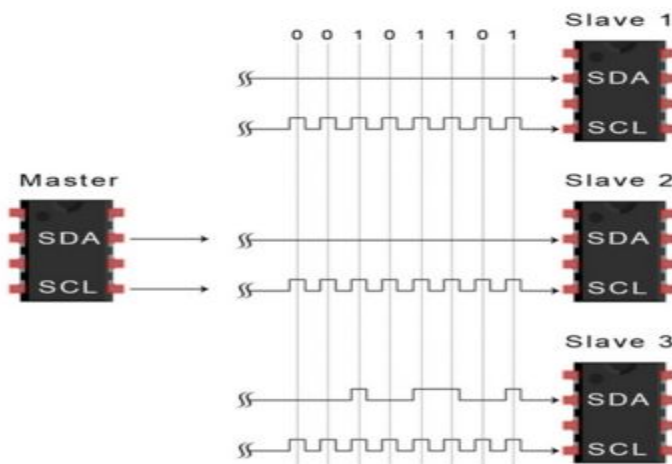
2- از طریق بیت read/write 10 یا 7 بیتی توسط master برای slave ها فرستاده می شود، تا هدف مورد نظر تشخیص داده شود.



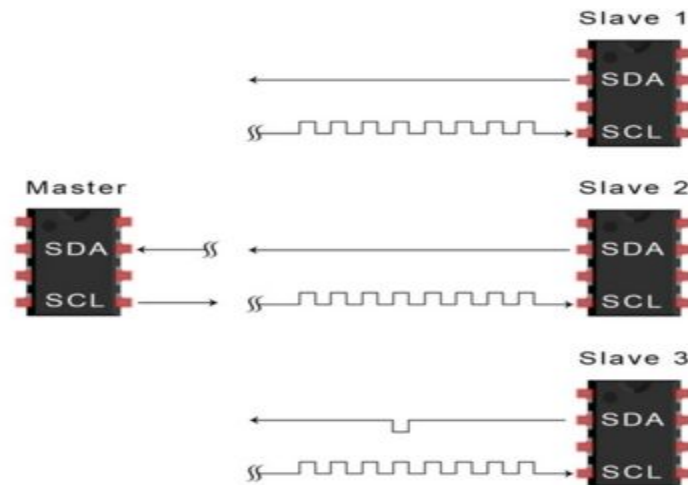
3- هر کدام از slave ها آدرس فرستاده شده را با آدرس خود مقایسه می کنند. در صورت همخوانی آدرس، slave مورد نظر بیت ACK را از طریق قرار دادن خط SDA در حالت low برای مدت بیت، ارسال می کند. و در صورتی که آدرس همخوانی نداشته باشد، خط SDA در حالت high ثابت باقی خواهد ماند.



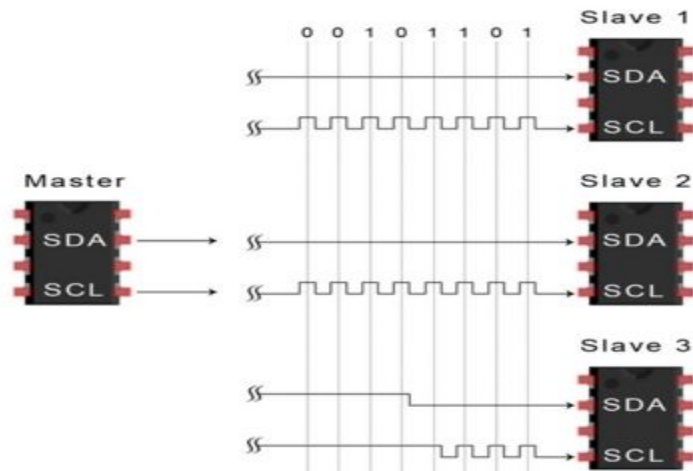
4- Master بسته داده را ارسال و یا دریافت می کند.



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.



۳- در اتصال **UART**، نقش پارامتر **baud rate** چیست؟ و به چه دلیل این پارامتر اهمیت پیدا میکند؟

در ارتباط سریال از آنجایی که انتقال داده به صورت بیت به بیت روی یک سیم انجام می گیرد، لازم است سرعت ارسال داده توسط فرستنده با سرعت دریافت داده در گیرنده برابر باشد. در نتیجه باید در ابتدای برقراری ارتباط، سرعت ارسال بین طرفین توافق شود.

در اتصال **UART** این توافق به این صورت است که پیش از آغاز ارتباط نرخ ارسال داده بین فرستنده و گیرنده مورد توافق قرار می گیرد. این سرعت معمولاً با نرخ **bps** نشان داده میشود که نشان دهنده ی تعداد بیت فرستاده شده در ثانیه است.

نرخ باود (**baud rate**) در واقع تعداد تغییرات سیگنال داده در ثانیه است که لزوماً در مخابرات با تعداد بیت‌های ارسال یکسان نیست. اما در ارتباط سریال، چون تنها دو ولتاژ وجود دارد که هرکدام نشان دهنده یک بیت است، تعداد تغییرات در ثانیه با تعداد بیت در ثانیه یکسان است. بنابراین در این بحث می توان نرخ بیتی و نرخ باود را یکسان در نظر گرفت.

۶- هر کدام از روشهای **I2C** و **UART** مزایا و معایبی دارند، این دو را با هم مقایسه کنید و برتریهای هر یک را برشمارید.

مقایسه ی **UART** و **I2C**:

I2C	UART	
زنجیر کردن دستگاه ها به یکدیگر ساده است	ساده	پیچیدگی
از UART سریعتر است	پایین	سرعت
حداکثر 127 دستگاه اما هر چه تعداد دستگاه ها بیشتر شود پیچیده تر میشود	حداکثر 2 دستگاه	تعداد دستگاه
2 سیم	1 سیم	تعداد سیم
پشتیبانی از چند master و چند slave	از master و slave پشتیبانی نمیکند	تعداد master و slave

مزایای روش **UART** :

- پیاده سازی آن ساده است.
- نیازی به کلاک ندارد.
- استفاده از **parity bit** به عنوان مکانیزم کشف خطا

مزایای روش **I2C** :

- انعطاف پذیرتر است زیرا میتواند از ارتباط **multi master** و **multi slave** پشتیبانی کند.
- استفاده از آن ساده است زیرا برای برقراری ارتباط میان دستگاه های مختلف فقط از دو سیم دو طرفه استفاده میکند.
- با نیاز های دستگاه های **slave** متنوع سازگار است

- as a low pin/signal count even with numerous devices on the bus

۷- در مورد سریال نرم افزاری و نحوه ی کار آن توضیح مختصری دهید. محدودیتهای آن نسبت به سریال سخت افزاری

چیست؟

هدف از کتابخانه‌های سریال نرم افزاری امکان استفاده از دیگر پین‌های دیجیتال از آردوینو برای برقراری ارتباط سریال است. ممکن است چند پین از این پین‌های دیجیتال برای ارتباط سریال در نظر گرفته شود. ماکسیمم سرعت این ارتباط به **115200 bps** می‌رسد. این کتابخانه ارتباط سریال را بر روی آن پین دیجیتال شبیه سازی می‌کند و به دلیل این که شبیه سازی شده است محدودیت‌هایی نسبت به سریال واقعی (سخت افزاری) دارد.

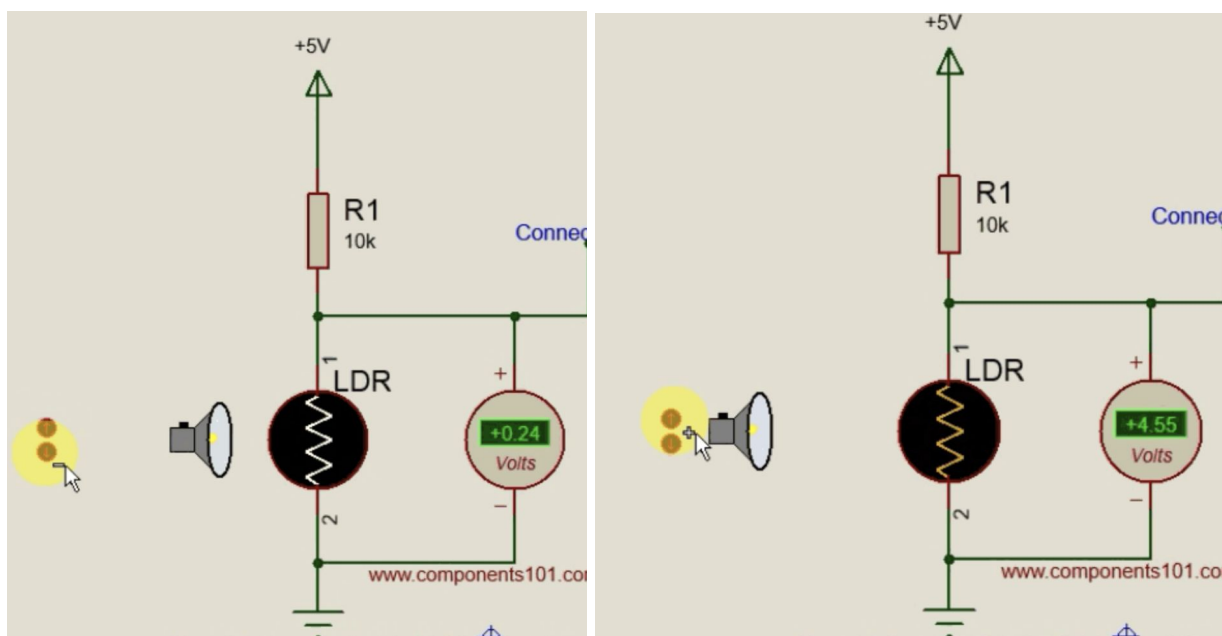
اگر از چندین سریال نرم افزاری استفاده شود فقط یکی از آنها می‌تواند در یک لحظه داده دریافت کند. پین‌های استفاده‌شده در ارتباط سریال باید از بتوانند مدیریت وقفه کنند. تمام پین‌های دیجیتال این امکان را ندارند و به همین دلیل، از برخی از آنها استفاده می‌شود. ماکسیمم سرعت گرفتن داده در ارتباط سریال نرم افزاری برابر **57600bps** است.

۸- نحوه کارکرد سنسور نورسنج گفته شده را توضیح دهید. دقت این سنسور در چه حدودی است؟ حداقل شدت نور لازم برای دریافت داده ی صحیح از این سنسور و بیشترین شدت نوری که میتواند پوشش دهد تقریباً چقدر است؟

LDR مخفف **Light Dependant Resistance** نوعی مقاومت است که بر اساس نوری که به آن می‌تابد مقاومتش تغییر می‌کند. هنگامی که این سنسور در تاریکی کامل است، مقاومتی در حدود چند مگا اهم دارد و هنگامی که در نور شدید قرار می‌گیرد، مقاومتی در حد چند اهم پیدا می‌کند.

هنگامی که این سنسور در یک مدار مقاومتی مانند شکل زیر قرار بگیرد، با مقادیر مختلف مقاومتی جریان متفاوتی از خود عبور می‌دهد و اختلاف پتانسیل دو طرف آن متفاوت می‌شود. این ولتاژ را می‌توان به شدت نور نگاشت داد.

کمترین مقدار نوری که می‌تواند اندازه گیری کند، ۱۰ لوکس و بیشترین مقدار آن ۱۰۰۰ لوکس است. دقت آن لوکس است.



۹- اگر نیازمند به اتصال چند سنسور مختلف که با پروتکل I2C استفاده میکنند بشویم آیا در پیاده سازی با مشکلی روبرو میشویم؟ حال اگر دو سنسور از یک نوع (مثلا دو تا سنسور MPU6050) داشته باشیم چطور؟ اگر سه تا از این سنسور داشته باشیم چطور؟ (جواب به ترتیب خیر، خیر، بله است! توضیح دهید چرا)

هر کدام از سنسور ها یک آدرس متفاوت دارند که با استفاده از این آدرس با آن ها ارتباط برقرار می شود. این آدرس یک آدرس ۷ یا ۸ بیتی است. پس به تعداد آدرس های مختلفی که می توان ساخت می توان از سنسور های مختلف استفاده کرد.

این سنسور دارای یک بیت است که با تغییر دادن آن به ۰ یا ۱ می توان آدرس های متفاوتی ایجاد کرد. پس اگر دوتا سنسور از این نوع داشته باشیم همچنان مشکلی پیش نمی آید.

چون تنها یک بیت متغیر وجود دارد نمی توان سه سنسور از این نوع با آدرس های متفاوت بسازیم پس با مشکل رو به رو می شویم.

۱۰- طراحی مفهومی این تمرین را مانند نمونه ای که در تمرین اول دیدید رسم کنید.

