

به نام خدا

گزارش پروژه دوم

کامران حسینی ۸۱۰۱۹۵۵۵۰

امیر محمد رنجبر پازکی ۸۱۰۱۹۵۴۰۲

علی عدالت ۸۱۰۱۹۵۴۲۷

سیستم‌های نهفته بی‌درنگ

استاد: دکتر کارگهی و دکتر مدرسی

دانشکده مهندسی برق و کامپیوتر دانشگاه تهران

زمستان ۱۳۹۷

● سوالات:

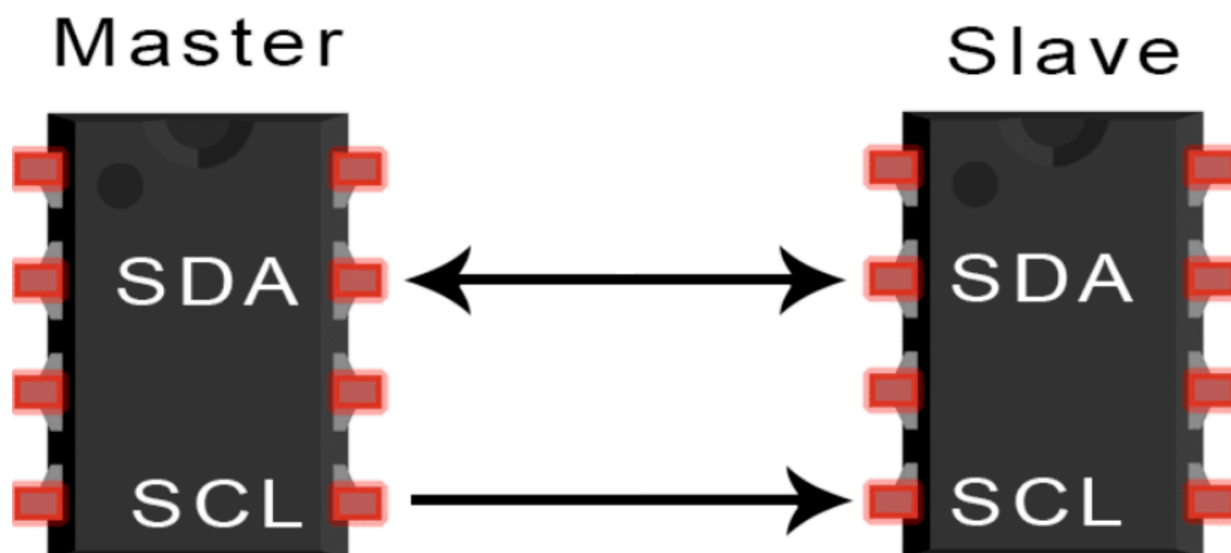
۱. پروتکل I²C قابلیت اتصال **multi masters – multi slaves** را دارد. توضیح دهید مشکلات پیش روی این حالت چیست؟ در این پروتکل از چه مکانیزم داوری ای استفاده می شود و چگونه مشکل تداخل را حل کرده است؟

در این حالت اینکه چند **master** داریم ممکن است که چند **master** با هم در حال نوشتن و گذاشتن کلاک باشند که باعث میشود اطلاعات از دست برود. برای حل این مشکل هر **master** برای اینکه کنترل **bus** را در دست بگیرد ابتدا چک میکند که آیا **bus** آزاد هست یا خیر (چک میکند که ولتاژ روی **bus** پایین است (چون در این حالت **bus** مشغول است) یا خیر) و اگر آزاد بود کنترل **bus** را در دست میگیرد و همه دیگر دستگاه ها در این حالت به عنوان **slave** به کار خود ادامه می دهند.

مکانیزم داوری (**Arbitration mechanism**) نیز به این صورت است که اگر دو **master** همزمان قصد نوشتن **data** روی **bus** را داشته باشند، دستگاهی مجوز نوشتن دارد که کلاک کمتری را در نظر گرفته باشد.

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

در I²C ویژگی های خوب **SPI** و **UART** با هم وجود دارند. در این پروتکل می توان چندین **slave** را به یک **master** متصل کرد و می توانند با چند **master** یک یا چند **slave** را کنترل کرد. در این ارتباط از دو سیم برای انتقال داده استفاده می شود که نمونه در زیر آمده است.



خط SDA برای انتقال داده بین master و slave استفاده می شود. SCL خطی هست که کلاک را دارد و از master کلاک مشخص می شود و slave ها با این کلاک کار می کنند. داده بر روی SDA با کلاک در SCL همگام است. در پروتکل I²C داده های ارسال شده از بخش های مختلفی تشکیل شده است. در ابتدای هر بسته برای شروع انتقال داده خط SDA از یک به صفر تغییر می کند و بعد از آن خط SCL نیز این تغییر را انجام می دهد تا شروع انتقال مشخص شود. در قسمت بعد آدرس slave قرار می گیرد که بوسیله ی آن می توان برای ارسال داده به slave های مختلف تنها از یک خط انتقال استفاده شود. بیت بعدی نشان دهنده این است که آیا master می خواهد داده ای را ارسال کند یا داده ای از slave در یافت کند. مکانیزم تشخیص خطا در این پروتکل این است که بعد از بیت های کنترلی و بعد از هر یک از فریم های داده یک بیت به عنوان acknowledge وجود دارد که گیرنده با یک کردن آن دریافت داده را تایید می کند. در آخر برای اطلاع پایان انتقال خط SCL از صفر به یک و SDA هم به همین صورت تغییر می کند که نشان دهنده پایان بسته پیام است.

۳. در اتصال UART (که معروفترین پروتکل ارتباط سریال می باشد)، نقش پارامتر baud rate چیست؟ و به چه دلیل این پارامتر اهمیت پیدا می کند؟

این نرخ مشخص کننده سرعت انتقال در روش سریال است. به بیانی دیگر، این نرخ مشخص می کند که فرستنده چه مدت زمانی خط را روی مقدار ۰ یا ۱ نگه می دارد و گیرنده با چه سرعتی باید نمونه برداری انجام بدهد. این پارامتر بسیاری کلیدی است چراکه همگامی تنها توسط این سرعت مشخص می شود. هر دو ماژول فرستنده و گیرنده باید با یک baud rate کار کنند. اگر بسیار سریع نمونه برداری شود، داده تکراری خوانده می شود و اگر کند این اتفاق رخ دهد، داده ها از بین می رود.

۴. هر کدام از روش های I²C و UART مزایا و معایبی دارند. این دو را با هم مقایسه کنید و برتری های هر یک را برشمارید.

روش **uart** به صورت **p2p** است در حالی که **i2c** برای **general broadcast** استفاده میشود. کلاک در **i2c** همیشه بوسیله **master** تولید میشود و دستگاه های **slave** وابسته به کلاک **master** هستند. در حالی که در **uart** هر دستگاه کلاک مخصوص خود را دارد.

انتقال اطلاعات در **uart**، غیرهمگام (**asynchronous**) است در حالی که در **i2c** این انتقال همگام (**synchronous**) است.

مزیت **uart** سادگی آن در اتصال و استفاده و وجود داشتن آن در اکثر دستگاه های با اتصال ۹ پایه است. از طرفی در **i2c** با افزایش تعداد **master** و **slave** ها پیچیدگی ارتباط افزایش میابد و نیازمند این است که قسمتی از برنامه نرم افزاری به کنترل ارتباط بپردازد در حالی که در **uart** این مشکل وجود ندارد. انتقال دیتا در **uart** قابلیت تغییر سرعت ندارد. یعنی اطلاعات تنها با سرعتی که قبل اتصال دو دستگاه با هم تعیین کرده اند قابل انتقال است در حالی که در **i2c** با تغییر **clock** این سرعت قابل تغییر است.

۵. در مورد سریال نرم افزاری و نحوه کار آن توضیح مختصری دهید. محدودیت های آن نسبت به سریال سخت افزاری چیست؟

کتابخانه های سریال نرم افزاری برای این طراحی شده است تا بتوان از طریق دیگر پین های دیجیتال از آردوینو ارتباط سریال برقرار کرد. ممکن است چند پین برای ارتباط سریال در نظر گرفته شود که ماکسیمم سرعت این ارتباط به **۱۱۵۲۰۰ bps** می رسد. این کتابخانه ها و این گونه برنامه ها برای ایجاد نرم افزاری ارتباط سریال محدودیت هایی دارند که در ادامه آمده است.

اگر از چندین **software serial port** استفاده شود فقط یکی از آنها می تواند در یک زمان دیتا دریافت کند. پین های مورد استفاده در این ارتباط باید بتوانند تغییرات **interrupt** ها را پشتیبانی کنند که تمام پین های مثلا آردوینو این خاصیت را ندارند و نمی توان از تمام پین های آن برای این کار استفاده کرد. ماکسیمم سرعت **RX** ارتباط سریال نرم افزاری برابر **۵۷۶۰۰ bps** است.

۶. نحوه کارکرد سنسور فاصله‌سنج گفته شده را توضیح دهید. دقت این سنسور در چه حدودی است؟ حداقل فاصله لازم برای دریافت داده‌ی صحیح از این سنسور و بیشترین فاصله‌ای که می‌تواند پوشش دهد تقریباً چقدر است؟

این سنسور یک صوت با فرکانس بالا را به سمت مقصدی که فاصله تا آن را می‌خواهیم می‌فرستد. زمان رفت و برگشت موج اندازه گرفته می‌شود و سپس، با تقسیم بر سرعت صوت فاصله به صورت دقیق به دست می‌آید.

این سنسور فاصله بین ۲ تا ۴۰۰ سانتی‌متر را اندازه می‌گیرد و دقت آن ۰.۳ سانتی‌متر است.

۷. اگر در پروژه‌های نیاز به استفاده از دو مازول فاصله‌سنج داشته باشیم چه مشکلی می‌تواند به وجود بیاید؟

ممکن است به دو مشکل برخورد کنیم:

اول اینکه این دو سنسور چون هر دو با امواج صوت کار میکنند و طبیعتاً فرکانس یکسانی نیز دارند (چون سنسور های یکسانی هستند) امواج صوتی که تولید میکنند با هم تداخل خواهد داشت و موجب اختلال در عملکرد یکدیگر میشود.

و دوم اینکه خواندن دیتا به صورت همزمان از این دو سنسور ممکن نیست یعنی باید ابتدا اولی را پینگ کرد، سپس پس از گذشت مدت زمان کافی سنسور دیگر را پینگ کرد و همینطور به ترتیب تا از اختلال در دریافت اطلاعات سنسورها جلوگیری کنیم.

۸. اگر نیازمند به اتصال چند سنسور مختلف که از پروتکل I2C استفاده می‌کنند بشویم، آیا در پیاده‌سازی با مشکلی روبرو می‌شویم؟ حال اگر دو سنسور از یک نوع (مثلاً دو تا سنسور MPU۶۰۵) داشته باشیم چگونه؟ اگر سه تا از این سنسور داشته باشیم چگونه؟ (جواب به ترتیب خیر، خیر، بله است! توضیح دهید چرا)

هر سنسور آدرس خود را برای ایجاد ارتباط با پروتکل I²C دارد و آدرس این سنسورها متفاوت است پس با قرار دادن آدرس خود در میان داده ارسالی می توانند در I²C شرکت کنند و آدرس ها ۷ بیتی است پس به این تعداد سنسور با شماره متفاوت بدون مشکل می تواند در I²C شرکت کند.

برخی سنسورها مانند MPU۶۰۷۰ این امکان را به برنامه نویس می دهند که به کمک تغییر دادن یک پین دیجیتال (در این جا این پین AD۰ نام دارد) آدرس دیگری به عنوان آدرس MPU۶۰۷۰ تعیین شود در نتیجه به کمک یک بیت می توان ۲ آدرس مختلف به دو ماژول MPU۶۰۷۰ نسبت داد. همان طور که گفته شد، چون AD۰ یک پین دیجیتال است و تنها دو حالت دارد بنابراین نمی توان بیشتر از دو آدرس برای این ماژول داشته باشیم.

۹. طراحی مفهومی این تمرین را رسم کنید.

