

بسمه تعالی



دانشگاه صنعتی شریف

دانشکده مهندسی برق

تمرین سری سوم درس سیستم های نهفته بی درنگ

دستیاران آموزش مربوطه:

ابوالفضل رجائیان

مهدی بخشی

استاد درس:

دکتر ایمان غلامپور

اردیبهشت ۱۴۰۳

۱. آردوینو

۱. مقدمه:

در این تمرین قصد داریم با رابط‌های (interface) آردوینو بیشتر آشنا شویم و کار با سنسورهای مختلف را یاد بگیریم. به طور کلی شما در این تمرین باید از سه آردوینو، یک شبکه شامل یک گره مرکزی و دو گره دیگر که هر کدام یک سنسور روی خود دارند را ایجاد کنید. هر کدام از این گره‌ها با گره مرکزی در ارتباط هستند و داده‌هایی که از سنسورهای خود دریافت می‌کنند را به آن ارسال می‌کنند. وظیفه‌ی گره مرکزی پردازش اطلاعات و نمایش تصمیم اتخاذ شده روی یک LCD است.

برای شبیه‌سازی از Proteus استفاده کنید اما به جای استفاده از Arduino Studio که معمولاً استفاده می‌شود از PlatformIO که یک اکوسیستم کامل برای توسعه‌ی پروژه‌های IoT در اختیار ما قرار می‌دهد، استفاده کنید.

هدف از انجام این تمرین یادگیری کار با رابط‌های آردوینو، آشنایی کامل با پروتکل‌های سریال، I2C و سنسورها و یادگیری استفاده از آنها می‌باشد.

۲. پیش‌زمینه:

✓ رابط سریال RS232

یکی از مواردی که در این تمرین با آن آشنا می‌شوید ارتباط سریال است که با نام RS232 نیز شناخته می‌شود. ارتباط سریال برخلاف ارتباط موازی، از یک سیم برای ارسال داده‌ها استفاده می‌کند و داده‌ها در فاصله‌های زمانی پشت سر هم روی خط ارسال می‌شوند. امروزه اکثر ارتباطات راه دور برای کاهش هزینه و کاهش نویز به صورت سریال پیاده سازی می‌شود. در این پروتکل چند موضوع مهم باید مورد توجه قرار بگیرند:

* نحوه هماهنگی سرعت ارسال و دریافت داده‌ها بین فرستنده و گیرنده

از آنجایی که انتقال داده به صورت بیت به بیت روی یک سیم انجام می‌شود؛ 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. این توافق می‌تواند به صورت همگام^۱ (USART) و یا غیر همگام^۲ (UART) انجام شود.

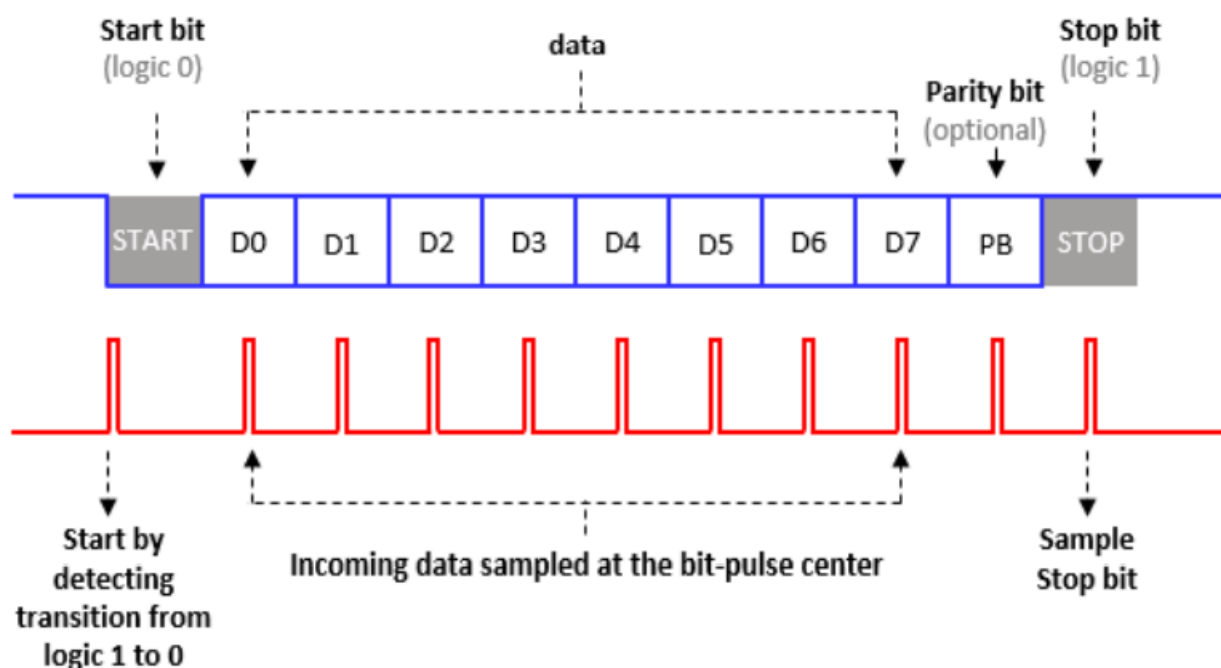
* نحوه framing داده‌ها

چیدمان، طول داده‌ی رد و بدل شده و نحوه‌ی آغاز و پایان ارسال داده باید از قبل مشخص باشد.

برای درک بهتر، در ارتباط سریال غیر همگام، در حالت بیکار (idle)، مقدار ۱ بر روی خط ارتباطی قرار دارد. برای شروع ارسال داده، یک بیت صفر بر روی خط قرار داده می‌شود تا طرف گیرنده متوجه شروع انتقال داده شود. گیرنده با دریافت این بیت، بیت‌های بعدی را به عنوان داده در نظر می‌گیرد. فرستنده هم پس از ارسال بیت صفر، به ترتیب از بیت کم ارزش داده شروع می‌کند، تعدادی بیت (حداکثر ۸ بیت) ارسال می‌کند و در نهایت دو بیت ۱ ارسال می‌کند که نشان توقف یا پایان است.

¹ Universal Synchronous Serial Receiver And Transmitter

² Universal Asynchronous Serial Receiver And Transmitter



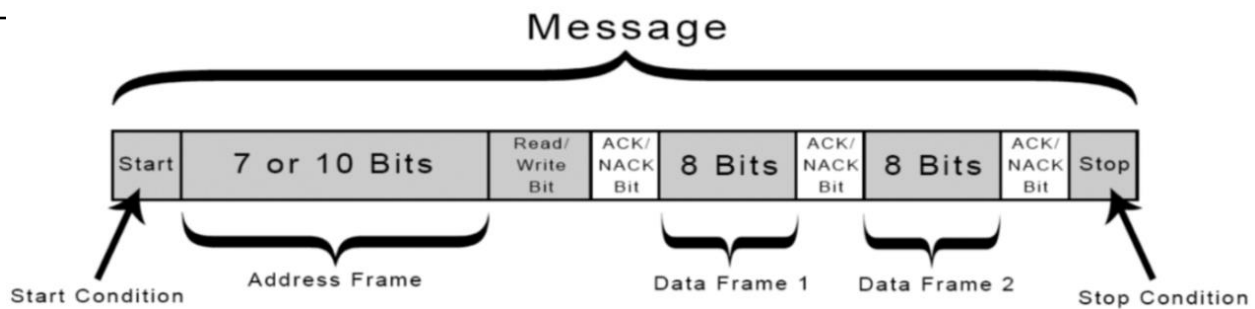
* مکانیزم کشف خطا

به صورت استاندارد، برای کشف خطا از یک یا دو بیت parity استفاده می شود که در طرف گیرنده با دریافت آن می تواند از وجود خطا در خط با خبر شود. برای اطلاعات بیشتر می توانید از این [لینک](#) که به صورت کامل و با جزئیات بیشتر این ارتباط را شرح داده است، استفاده کنید.

✓ ارتباط I2C

در ادامه به توضیح یک مدل از ارتباط سریال به نام I2C می پردازیم. این پروتکل در قطعاتی مانند تایمر، شتاب سنج، و ژيروسکوپ استفاده می شود.

در ارتباط I2C از دو خط به نام های SDA و SCL استفاده می شود. خط SCL حامل سیگنال clock است و خط SDA برای ارسال و دریافت داده ها به کار می رود. در این پروتکل خط های SDA و SCL همگام (synchronous) هستند و خط کلاک توسط یکی از طرفین که پایه master نامیده می شود، تنظیم می شود. در پروتکل I2C، داده های ارسال شده از بخش های مختلفی تشکیل شده است. در ابتدای هر بسته، برای شروع انتقال، داده ی خط SDA از یک به صفر تغییر می کند و بعد از آن خط SCL هم همین تغییر را اعمال می کند تا نشانه ی شروع ارسال باشد. در قسمت بعدی، آدرس پیرو (slave)، که گره ای است که گیرنده داده ی ارسالی است، قرار داده می شود. با استفاده از این بخش، می توان برای ارسال داده به slave های مختلف تنها از یک خط انتقال استفاده کرد. بیت بعدی نشان دهنده ی این است که آیا گره master می خواهد داده ای را ارسال کند و یا داده ای از slave دریافت کند. مکانیزم تشخیص خطا در این پروتکل به این صورت است که بعد از بیت های کنترلی، و بعد از هر یک از frame های داده که در شکل زیر مشاهده می کنید؛ یک بیت برای acknowledge وجود دارد. و گیرنده با یک کردن آن، دریافت داده ها را تایید خواهد کرد. در نهایت برای پایان انتقال، خط SCL از صفر به یک و SDA هم به همین صورت تغییر می کند و نشان دهنده ی پایان بسته ی پیام است.



چیدمان سیگنال‌ها در ارتباط I2C

۳. شرح تمرین:

در این تمرین قرار است یک شبکه از سه آردوینو را شبیه‌سازی کنید. شرح هر کدام از گره‌ها به صورت زیر است:

TH-Board: این گره از طریق ارتباط I2C به یک سنسور اندازه‌گیری دما و رطوبت (SHT25) متصل می‌شود و اطلاعات سنسور را بعد از پردازش از طریق ارتباط سریال به ماژول بلوتوث (HC-05) ارسال می‌کند. این ماژول بلوتوث در ادامه اطلاعات را به ماژول متصل به گره مرکزی ارسال خواهد کرد.

Light-Board: این گره به یک سنسور نورسنج (LDR) متصل است و باید اطلاعات این سنسور را دریافت کند و شدت نور برحسب Lux را با استفاده از ارتباط سریال برای ماژول اصلی ارسال کند. برای تبدیل خروجی سنسور به Lux می‌توانید از این لینک کمک بگیرید. (ارتباط این گره با گره مرکزی از طریق یک سیم برقرار می‌شود).

Main-Board: این گره که گره مرکزی ماست، به یک LCD مجهز است و اطلاعات سنسورها را دریافت می‌کند و بسته به شرایط، پیام مناسب را روی LCD چاپ می‌کند. (دقت کنید که این گره نیاز به یک ماژول بلوتوث هم دارد که داده‌ها را از گره اول دریافت کند). این گره تصمیم می‌گیرد که بر اساس شرایط، آبیاری قطره‌ای صورت بگیرد یا خیر و اگر جواب مثبت است، این کار را با چه نرخی انجام دهد. برنامه‌ی شما باید تحت شرایط زیر، خروجی‌های متناسب را روی LCD نمایش دهد:

اگر رطوبت بالای ۸۰ درصد بود، آبیاری صورت نگیرد.

اگر رطوبت کمتر از ۵۰ درصد بود، آبیاری با نرخ ۱۵ سی‌سی بر دقیقه انجام گیرد

اگر رطوبت بین ۵۰ تا ۸۰ درصد بود نیز سه حالت رخ می‌دهد. اگر دما کمتر از ۲۵ درجه‌ی سلسیوس و شدت نور کمتر از ۶۰۰ لوکس بود، آبیاری با نرخ ۱۰ قطره بر دقیقه صورت گیرد. اگر دما کمتر از ۲۵ درجه‌ی سلسیوس و شدت نور بیشتر از ۶۰۰ لوکس بود، آبیاری با نرخ ۵ قطره بر دقیقه صورت گیرد. نهایتاً اگر دما بیشتر از ۲۵ درجه‌ی سلسیوس بود، آبیاری با نرخ ۱۰ قطره بر دقیقه صورت گیرد.

در پیاده‌سازی این تمرین، به نکات زیر دقت کنید:

TH-Board باید دو نوع داده را برای گره مرکزی ارسال کند. برای این کار پروتکل مشخصی وجود ندارد. می‌توانید با استفاده از خلاقیت خود راهکاری برای این کار پیشنهاد دهید. برای مثال یک ایده می‌تواند استفاده از delimiter برای جداسازی انواع داده باشد.

ارتباط بلوتوث تفاوتی با ارتباط سریال توسط سیم ندارد. در واقع ارسال بی‌سیم داده‌ها در لایه‌های پایین‌تر مدیریت شده و شما تفاوتی بین این دو انتقال حس نخواهید کرد. (برای ارسال داده‌ها بین دو آردوینو از طریق بلوتوث می‌توانید از این لینک استفاده کنید).

قبل از شروع تمرین حتماً نحوه کارکرد پروتکل **I2C** و **Serial** را به صورت دقیق مطالعه کنید. (تسلط شما بر این دو پروتکل در زمان تحویل تمرین سنجیده می‌شود).

۴. سؤالات:

- 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؟
- ۲- مراحل ارتباط دو دستگاه به یکدیگر از طریق این پروتکل I2C را بیان کنید.
- ۳- در اتصال UART، نقض پارامتر baud rate چیست؟ و به چه دلیل این پارامتر اهمیت پیدا می کند؟
- ۴- هرکدام از روش های I2C و UART مزایا و معایبی دارند، این دو را با هم مقایسه کنید و برتری های هر یک را برشمارید.
- ۵- سریال نرم افزاری و نحوه ی کار آن را توضیح مختصری دهید. محدودیت های آن نسبت به سریال سخت افزاری چیست؟
- ۶- 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 می تواند پوشش دهد تقریباً چقدر است؟
- ۷- اگر نیازمند به اتصال چند سنسور مختلف که با پروتکل I2C استفاده می کنند، بشویم، آیا در پیاده سازی با مشکلی روبه رو می شویم؟ حال اگر دو سنسور از یک نوع (مثلاً دو تا سنسور MPU6065) داشته باشیم چطور؟ اگر سه تا از این سنسور داشته باشیم چطور؟ (جواب به ترتیب خیر، خیر، بله است! توضیح دهید چرا)
- ۸- طراحی مفهومی این تمرین را رسم کنید.

۲ آشنایی با زبان اسمبلی پردازنده ARM

رمزگشایی پیام با استفاده از الگوریتم RSA

- توضیح:** در این مسئله، یک پیام رمزنگاری شده با الگوریتم RSA داده شده است. شما باید برنامه ای بنویسید که این پیام رمزنگاری شده را با استفاده از کلید عمومی (e, n) که در اختیار شما قرار دارد، رمزگشایی کند و پیام اصلی را نمایش دهد.
- وظیفه ی شما:** برنامه ای بنویسید که با استفاده از زبان اسمبلی ARM، الگوریتم RSA را برای رمزگشایی پیام اجرا کند. برنامه باید پیام رمزنگاری شده، کلید عمومی ورودی (e, n) و همچنین پیام رمزگشایی شده را نمایش دهد.
- فرض کنید پیام رمزنگاری شده با کلید عمومی $(e = 17, n = 3233)$ به شکل زیر باشد:
- 1829 2760 2418 2935 2760 588 2418 588 2760 2935 1528 2760 1452 2760 1829
- شما باید این پیام را رمزگشایی کنید و پیام اصلی را نمایش دهید.

۳ آشنایی با برنامه نویسی میکروکنترلرهای STM32

۳.۱ تحقیق

در مورد کتابخانه HAL تحقیق کنید. مزایا و معایب آن را بیان کنید.

۳.۲ برنامه نویسی و شبیه سازی

سناریو: شما به عنوان یک مهندس مسئول طراحی یک ربات حرکتی هوشمند با استفاده از میکروکنترلر STM32 هستید. این ربات باید قادر به حرکت در محیط‌های مختلف باشد و با استفاده از سنسورها و مازول بلوتوث، از موانع جلوگیری کند و به هدف مورد نظر برسد.

طراحی سخت‌افزار: طراحی مدار الکترونیکی برای ربات حرکتی شامل میکروکنترلر STM32، موتورهای، سنسورهای فاصله، سنسورهای خط، سنسورهای ارتفاع و مازول بلوتوث است.

برنامه‌نویسی میکروکنترلر: نوشتن کدهای کنترلی برای خواندن داده‌های سنسورها و کنترل حرکت ربات. پیاده‌سازی الگوریتم‌های کنترلی برای جلوگیری از برخورد با موانع.

شبیه‌سازی در پروتئوس: ساخت مدل شبیه‌سازی شامل میکروکنترلر STM32 و سنسورهای مختلف و همچنین اتصال مدارهای ربات و سنسورها در محیط پروتئوس، و در نهایت تنظیم شبیه‌سازی برای شبیه‌سازی حرکت ربات و واکنش به داده‌های سنسورها.

تست و بهینه‌سازی: ارزیابی عملکرد الگوریتم‌های کنترلی در شبیه‌سازی و تنظیمات بهینه‌سازی آنها. تست رفتار ربات در شرایط مختلف شبیه‌سازی شده و اصلاح الگوریتم‌های کنترلی.

۴ تمرین زبان اسمبلی برج هانوی (Tower of Hanoi)

هدف:

هدف از این تمرین پیاده سازی مسئله برج هانوی در زبان اسمبلی با استفاده از رویکرد تکراری است. برنامه باید تعداد دیسک ها را از کاربر بگیرد، مشکل را به صورت تکراری حل کند و مراحل حل را به همراه تعداد کل حرکت های مورد نیاز خروجی بگیرد.

شرح مشکل:

برج هانوی یک مشکل کلاسیک است که در آن شما سه میخ و تعدادی دیسک با اندازه های مختلف دارید. دیسک ها به ترتیب صعودی بر روی یک میخ، با کوچکترین دیسک در بالا و بزرگترین دیسک در پایین شروع می شوند. هدف این است که همه دیسک ها را از اولین میخ به میخ سوم با پیروی از این قوانین منتقل کنید:

- فقط یک دیسک را می توان در یک زمان جابجا کرد.
- دیسک فقط در صورتی قابل جابجایی است که بالاترین دیسک روی پشته باشد.
- هیچ دیسکی را نمی توان روی دیسک کوچکتر قرار داد.

وظیفه:

یک برنامه به زبان اسمبلی بنویسید تا مشکل Tower of Hanoi را برای n دیسک حل کند، جایی که n توسط کاربر ارائه شده است. برنامه شما باید یک راه حل تکراری را پیاده سازی کند و مراحل مورد نیاز برای جابجایی همه دیسک ها از اولین میخ به میخ سوم را به همراه تعداد کل حرکت ها خروجی دهد.

الزامات:

- برنامه باید از کاربر بخواهد که تعداد دیسک ها را وارد کند.
- یک راه حل تکراری برای مشکل برج هانوی اجرا کنید.
- خروجی هر حرکت مرحله به مرحله.
- تعداد کل حرکات مورد نیاز برای حل مشکل را خروجی بگیرید.