

دانشگاه تهران  
دانشکده‌ی مهندسی برق و کامپیوتر



## سیستم‌های نهفته‌ی بی‌درنگ

تمرین دوم

آشنایی با پروتکل‌های ارتباطی

و سنسورها

طراحان:

فرزاد حبیبی  
ایمان مجتهد  
حسین سلطانیلو

اساتید:

دکتر کارگهی  
دکتر مدرسی

۱۳۹۸-۱۳۹۹

## ۱. مقدمه

در این تمرین قصد داریم با رابط‌های (interface) آردوینو آشنا شویم و کار با سنسورهای مختلف را یاد بگیریم. به طور کلی شما در این تمرین باید با استفاده 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 ارتباطند و داده‌هایی که از سنسورهای خود دریافت می‌کنند را به آن ارسال می‌کنند. وظیفه‌ی گره مرکزی پردازش اطلاعات دریافتی و نمایش تصمیم اتخاذشده روی یک LCD است.

برای شبیه‌سازی همانند تمرین قبل از Proteus استفاده می‌کنیم اما این بار بجای استفاده از Arduino Studio از [PlatformIO](#) که یک اکوسیستم کامل برای توسعه‌ی پروژه‌های IoT در اختیار ما قرار می‌دهد استفاده می‌کنیم.

هدف از انجام این تمرین یادگیری کار با رابط‌های آردوینو، آشنایی با پروتکل‌های سریال، I2C، 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 بر خلاف 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<sup>۱</sup> (USART) یا غیرهمگام<sup>۲</sup> (UART) انجام شود.

## • نحوه framing داده‌ها

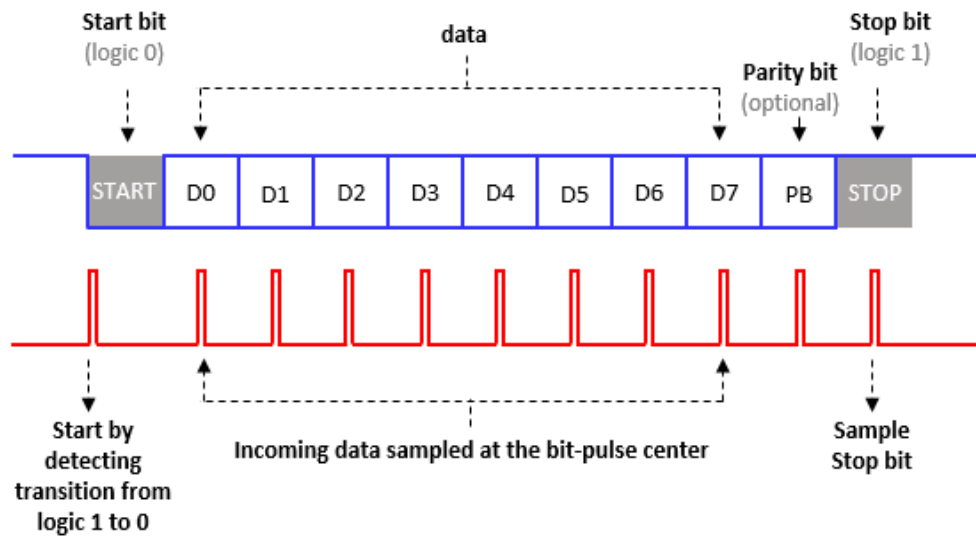
چیدمان، طول داده‌ی رد و بدل شده و نحوه‌ی آغاز و پایان ارسال داده باید از قبل مشخص باشد.

برای درک بهتر، در ارتباط سریال غیرهمگام، 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) ارسال می‌کند و در نهایت دو بیت ۱ ارسال می‌کند که نشان توقف یا پایان است.

---

<sup>۱</sup> Universal Synchronous Serial Receiver And Transmitter

<sup>۲</sup> Universal Asynchronous Serial Receiver And Transmitter



#### • مکانیزم کشف خطا

به صورت استاندارد، برای کشف خطا از یک یا دو بیت parity استفاده می‌شود که در طرف گیرنده با دریافت آن می‌تواند از وجود خطا در خط با خبر شود.

برای اطلاعات بیشتر می‌توانید 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<sup>۳</sup> یا مدار مجتمع یکپارچه می‌پردازیم.

این پروتکل در قطعاتی مانند تایمر، شتاب‌سنج، و ژيروسکوپ استفاده می‌شود؛ در نتیجه یادگیری آن اهمیت زیادی برای پروژه‌های پایانی شما خواهد داشت.

در ارتباط I2C از دو خط به نام‌های SDA<sup>۴</sup> و SCL<sup>۵</sup> استفاده می‌شود. خط SCL حامل سیگنال clock است و خط SDA برای ارسال و دریافت داده‌ها به کار می‌رود. در این پروتکل خط‌های SDA و SCL همگام (synchronous) هستند و خط کلاک توسط یکی از طرفین که پایه (master) نامیده می‌شود تنظیم می‌شود.

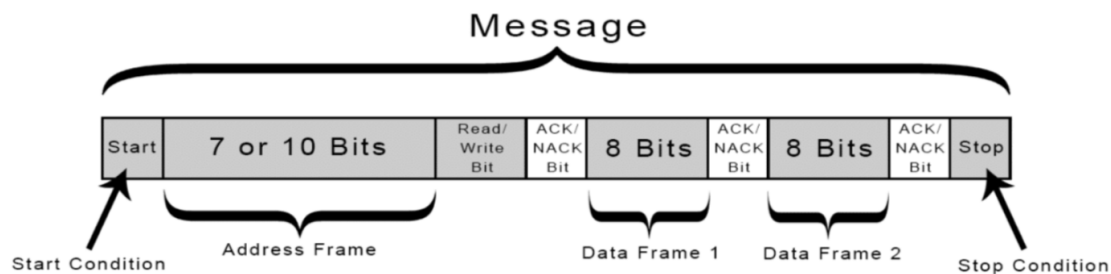
در پروتکل I2C، داده‌های ارسال شده از بخش‌های مختلفی تشکیل شده است. در ابتدای هر بسته، برای شروع انتقال، داده‌ی خط SDA از یک به صفر تغییر می‌کند و بعد از آن خط SCL هم همین تغییر را اعمال می‌کند تا نشانه‌ی شروع ارسال باشد. در قسمت بعدی، آدرس پیرو (slave)، که گره‌ای است که گیرنده داده‌ی ارسالی است، قرار داده می‌شود. با استفاده از این بخش،

<sup>۳</sup> Inter Integrated Circuit

<sup>۴</sup> Serial Data

<sup>۵</sup> Serial Clock

می‌توان برای ارسال داده به **slave**‌های مختلف تنها از یک خط انتقال استفاده کرد. بیت بعدی نشان دهنده‌ی این است که آیا **master** می‌خواهد داده‌ای را ارسال کند و یا داده‌ای از **slave** دریافت کند. مکانیزم تشخیص خطا در این پروتکل به این صورت است که بعد از بیت‌های کنترلی، و بعد از هر یک از **frame**‌های داده که در شکل زیر مشاهده می‌کنید؛ یک بیت برای **acknowledge** وجود دارد. و گیرنده با یک کردن آن، دریافت داده‌ها را تایید خواهد کرد. در نهایت برای نمایش پایان انتقال، خط **SCL** از صفر به یک و **SDA** هم به همین صورت تغییر می‌کند و نشان‌دهنده‌ی پایان بسته‌ی پیام است.



چیدمان سیگنال‌ها در ارتباط I2C

## ۳. شرح تمرین

در این تمرین قرار است یک شبکه از سه آردوینو را شبیه‌سازی کنید. شرح هرکدام 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 بلوتوث 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 بر حسب 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 ال‌سی‌دی 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 نرخ انجام دهد. برنامه‌ی شما باید تحت شرایط زیر، خروجی‌های متناسب را روی ال‌سی‌دی 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 پروژه حتما نحوه کارکرد پروتکل [I2C](#) و [Serial](#) را به صورت دقیق مطالعه کنید. (تسلط شما بر این دو پروتکل در زمان تحویل پروژه 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، نقش پارامتر baud rate چیست؟ و به چه دلیل این پارامتر اهمیت پیدا می‌کند؟
۶. هرکدام از روش‌های I2C و UART مزایا و معایبی دارند، این دو را با هم مقایسه کنید و برتری‌های هر یک را برشمارید.
۷. در مورد سریال نرم‌افزاری و نحوه‌ی کار آن توضیح مختصری دهید. محدودیت‌های آن نسبت به سریال سخت‌افزاری چیست؟
۸. نحوه کارکرد سنسور نورسنج گفته شده را توضیح دهید. دقت این سنسور در چه حدودی است؟ حداقل شدت نور لازم برای دریافت داده‌ی صحیح از این سنسور و بیشترین شدت نوری که می‌تواند پوشش دهد تقریباً چقدر است؟
۹. اگر نیازمند به اتصال چند سنسور مختلف که با پروتکل I2C استفاده می‌کنند بشویم آیا در پیاده‌سازی با مشکلی روبرو می‌شویم؟ حال اگر دو سنسور از یک نوع (مثلاً 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)
۱۰. طراحی مفهومی این تمرین را مانند نمونه‌ای که در تمرین اول دیدید رسم کنید.



## ۵. نرم‌افزارهای مورد نیاز

ابتدا به نحوی از شر **Arduino Studio** خلاص شوید! برای این کار، با استفاده از دستورالعمل‌های وبسایت [PlatformIO](#) اقدام به نصب این برنامه کنید. (این برنامه به صورت یک **plugin** برای ادیتور **VSCode** در دسترس می‌باشد).

برای هر یک از سه گره، یک پوشه ایجاد کنید و آن‌ها را در پوشه‌ی اصلی پروژه‌ی خود قرار دهید. **git**. شما باید در این پوشه باشد تا کدهای هر سه گره را در بر بگیرد.

نکته: برای ایجاد کردن ساختار اولیه هر یک از گره‌ها می‌توانید از دستور زیر استفاده کنید:

```
platformio init --board uno
```

در ادامه، جزئیات مربوط به هر یک از ۳ بورد را خواهید دید:

• **TH-Board**

این گره با استفاده از پروتکل **I2C** به سنسور **SHT25** متصل است، برای این کار تنها کافیسیت دو پورت **SDA** و **SCL** را به پورت‌های متناظر سنسور متصل کنید (در آردوینو **Uno** پورت‌های **SCL** و **SDA** با پورت‌های ورودی آنالوگ مشترک است)

سپس می‌توانید با نحوه‌ی کار با پروتکل **I2C** در آردوینو که توسط [کتابخانه Wire](#) انجام می‌گیرد آشنا شوید.

همچنین می‌توانید با اتصال یک **Virtual Terminal** به پورت **debug** سنسور از درست بودن کدهای خود اطمینان حاصل کنید. (پورت دیباگ در واقع داده‌ی سنسور را به صورت سریال خروجی می‌دهد)

• **Light-Board**

این گره باید به سنسور نورسنج متصل شود. از لیست کتابخانه‌های **Proteus** ماژول **LDR** را به پروژه اضافه کنید. با توجه به دیتاشیت‌های موجود در اینترنت این دو سنسور را به آردوینو متصل کنید.

برای استفاده از این سنسور می‌توانید به صورت مستقیم با ورودی‌ها کار کنید تا داده‌های آن را به دست آورید؛ اما پیشنهاد اکید ما این است که از کتابخانه‌های در دسترس برای آن استفاده کنید. برای این کار می‌توانید از قسمت **Quick Access -> Libraries -> Registry** و جستجوی نام کتابخانه، اقدام به نصب کتابخانه مورد نظر خود کنید. (امکان دارد برای سنسورها چند کتابخانه نوشته شده باشد؛ با توجه به تعداد دانه‌ها و یا نمونه کدها بهترین را انتخاب کنید)

می‌توانید با ترکیب نمونه کدهای مربوط به سنسورها، دیتای سنسور را گرفته و به شکلی مشخص (مثلا جدا کردن آن‌ها با یک حرف مناسب) از طریق پروتکل سریال برای گره اصلی ارسال کنید.

## • Main-Board

گره اصلی وظیفه‌ی گرفتن داده‌های ارسال شده از دو گره و پردازش و تصمیم‌گیری بر اساس آن‌ها و نمایش اطلاعات و تصمیم مرتبط با آن‌ها روی یک LCD را دارد. برای شروع ماژول LM041L با اندازه‌ی ۴×۲۰ را از لیست کتابخانه‌های Proteus به پروژه اضافه کنید. با استفاده از مستندات موجود در اینترنت، LCD را به آردوینو متصل کنید. (پایه‌ی En را حتماً به GND متصل کنید)

کتابخانه‌ی LiquidCrystal را مانند قسمت قبلی نصب کنید. با استفاده از تکه‌کدی از درستی اتصال خود مطمئن شوید. حال با اتصال یک ماژول بلوتوث به ورودی سریال آردوینو، دیتای خروجی TH-Board را دریافت و در اولین خط LCD نمایش دهید، حال باید داده‌های مربوط به گره دوم را نیز دریافت کنید. اگر دقت کنید آردوینو تنها یک پورت RX دارد! برای اینکار می‌توانید از کتابخانه SoftwareSerial استفاده کنید. این کتابخانه دوتا از پایه‌های دیجیتال آردوینو را به ورودی و خروجی مجازی سریال تبدیل می‌کند، با استفاده از این کتابخانه یک پورت RX دیگر به آردوینو اضافه کنید و بعد از گرفتن داده‌ی سنسور نورسنج، آن را در خط دوم LCD چاپ کنید و نهایتاً نرخ آبیاری قطره‌ای را در خط سوم چاپ کنید.

نکته: کتابخانه SoftwareSerial کتابخانه پایه‌ای برای این کار است. می‌توانید از کتابخانه‌های جایگزین مانند AltSoftSerial که امکانات بیشتری را در اختیار شما قرار می‌دهند استفاده کنید.

زمانی که مقدار سریال را می‌خوانید امکان دارد در میانه‌ی ارسال داده بوده باشد. مثلاً فرض کنید گره اول برای شما عدد ۱۲۳۴ را ارسال کند ولی زمانی که شما داده را می‌خوانید داده به صورت کامل خوانده نشده باشد و تنها مقدار ۱۲ را دریافت کنید! این مشکل را در نظر بگیرید و آن را رفع کنید. (راهکار شما نباید به صورت blocking باشد)

در پیاده سازی ارتباطات بین گره‌ها به نکات زیر توجه داشته باشید:

- ارتباط دو ماژول بلوتوث داده شده در محیط شبیه‌سازی به صورت مجازی برقرار می‌شود. برای اینکار روی این ماژول‌ها دوبار کلیک کنید و physical port یکی از ماژول‌ها را برابر COM۳ و دیگری را برابر COM۴ قرار دهید. سپس با استفاده از یکی از برنامه‌های ساخت ارتباط مجازی مثلاً com0com این دو پورت را به هم متصل کنید)

- خواندن اطلاعات مربوط از دو گره نباید به صورت blocking باشد. چراکه سرعت ارسال داده‌ها از گره‌ها می‌تواند متفاوت باشد و برای نمایش دیتای مربوط به یک گره نباید منتظر گره دیگر بمانیم. (برای پیاده‌سازی این قسمت متد available سریال را مطالعه کنید)

## ۶. نکات مهم:

- تمرکز این پروژه در کنار یادگیری و استفاده از پرتکل‌های ارتباطی، بر روی کد زدن صحیح به‌عنوان یک مهندس کامپیوتر برای این اجزا می‌باشد، در نتیجه بخشی از نمره این پروژه به تمیزی کد (استفاده از نام‌گذاری قابل‌فهم، استفاده درست از توابع و ...) تعلق دارد.
- استفاده از [Git](#) برای این پروژه اجباری می‌باشد و شما به‌جای آپلود پروژه ملزم به گذاشتن hash آخرین کامیت خود در محل آپلود می‌باشید. (حتماً پوشه‌ی `.vscode` را در `.gitignore` قرار دهید)
- با توجه به سنگین‌تر بودن این تمرین نسبت به تمرین قبل، پیشنهاد می‌شود قسمت‌های مختلف این تمرین بین افراد گروه تقسیم شود. البته همه‌ی افراد باید تسلط کافی به مباحث تمامی بخش‌های پروژه داشته باشند.
- این تمرین باید در قالب گروه‌های سه نفره انجام شود.
- تمرین تحویل حضوری خواهد داشت.
- برای ما مهم است که حاصل کار خود را تحویل دهید به همین دلیل به شدت با تقلب برخورد خواهد شد و به طرفین نمره ۱۰۰- تعلق خواهد گرفت.
- گزارش کار باید شامل جواب سؤالات و تصاویر کافی برای هر بخش باشد. همچنین نوشتن موارد اضافی و بیش‌از‌حد مشمول نمره‌ی منفی خواهد شد.
- سؤالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. توجه داشته باشید که دیگر شبکه‌های اجتماعی مانند تلگرام راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.