



دانشگاه تهران
دانشکده‌ی مهندسی برق و کامپیوتر



سیستم‌های نهفته‌ی بی‌درنگ

تمرین دوم

آشنایی با پروتکل‌های ارتباطی انتقال داده و سنسورها

طراحان:

فرزاد حبیبی

ایمان مجتهد

حسین سلطانلو

هدی مهدیانی

استاد:

دکتر مهدی کارگهی

دکتر مهدی مدرسی

۱۳۹۹-۱۴۰۰

آشنایی با پروتکل های ارتباطی انتقال داده و سنسورها

هدف از انجام این تمرین یادگیری کار با رابط های^۱ Arduino، آشنایی با پروتکل های انتقال داده به روش سریال (RS232 و I2C) و آشنایی با برخی از انواع سنسورها و نحوه استفاده از آنها است. برای انجام این تمرین از برد Arduino، شبیه ساز Proteus و بستری جهت توسعه سیستم های IOT با عنوان PlatformIO (به جای Arduino IDE) استفاده می شود. در این تمرین باید با استفاده از دو برد Arduino، یک شبکه شامل یک گره مرکزی و یک گره دیگر که به یک سنسور متصل است و داده های دریافتی از سنسور خود را به گره مرکزی ارسال می کند را ایجاد و شبیه سازی کنید. وظیفه ی گره مرکزی پردازش اطلاعات دریافتی و نمایش تصمیم اتخاذ شده روی یک LCD است.

➤ مفاهیم مقدماتی

روش های انتقال داده- به طور کلی برای انتقال داده ها از دو روش می توان استفاده کرد: انتقال موازی و انتقال سریال.

- **انتقال موازی:** در انتقال موازی تمامی بیت های یک داده، به طور همزمان طی یک پالس ساعت بر روی خطوط انتقال مختلف ارسال می شوند. این روش برای انتقال داده در مسیرهای کوتاه میان مبدا و مقصد مناسب بوده و با توجه به دریافت همزمان بیت ها در مقصد سرعت انتقال اطلاعات بسیار زیاد است. اما اگر فاصله مبدأ و مقصد زیاد باشد، هزینه ناشی از ایجاد ارتباط بسیار زیاد خواهد شد.

- **انتقال سریال:** در انتقال سریال طی هر پالس ساعت یک بیت انتقال می یابد و برای برقراری ارتباط باید حداقل یک خط انتقال از مبدأ به مقصد وجود داشته باشد. در این روش لازم است در ابتدا بر روی سرعت ارسال داده میان فرستنده و گیرنده توافق شود. همچنین نحوه فریم بندی، طول داده و نحوه ی آغاز و پایان ارسال داده باید از قبل مشخص باشد. نسبت به روش موازی هزینه ارتباط در این روش کمتر بوده اما سرعت انتقال داده نیز پایین تر است. امروزه اکثر ارتباطات راه دور برای کاهش هزینه و کاهش نویز به صورت سریال پیاده سازی می شود. ارتباط سریال به دو روش سنکرون و آسنکرون انجام می شود:

- **ارتباط سریال به روش آسنکرون^۲ (غیرهمگام)-** در این روش برای انتقال اطلاعات از فرستنده به گیرنده تنها از یک خط ارتباطی استفاده شده و به منظور ارسال و دریافت صحیح و به موقع اطلاعات، علاوه بر بیت های اصلی، بیت های شروع و پایان در ابتدا و انتهای آن ارسال می شوند. همچنین به عنوان مکانیزمی جهت کشف خطا و اطمینان از صحت اطلاعات دریافتی توسط گیرنده معمولاً از بیت توازن (Parity) استفاده می شود.

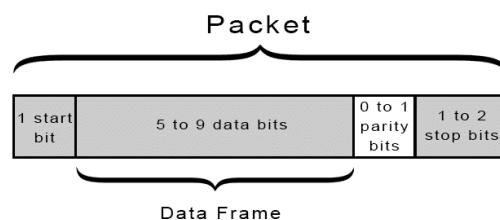
^۱ Interface

^۲ Asynchronous

○ ارتباط سریال به روش سنکرون^۳ (همگام) - در این روش، حداقل دو خط ارتباط بین فرستنده و گیرنده لازم است. یک سیم حاوی داده‌ای است که به طور سریال ارسال می‌شود و سیم دیگر حاوی پالس ساعت است که توسط فرستنده ارسال شده و گیرنده خود را با آن همگام می‌کند تا داده‌ها را از روی خط به درستی بخواند (به در این روش چند فریم داده می‌توانند در قالب یک پیام ارسال شوند و دیگر نیازی به بیت شروع و پایان برای هر داده ۸ بیتی نیست. البته فقط در ابتدای انتقال بسته داده‌ها، یک کاراکتر سنکرون، ارسال می‌شود و بلافاصله بیت‌های داده اصلی فرستاده خواهد شد که این موضوع باعث بالا رفتن سرعت انتقال داده‌ها در روش سنکرون می‌گردد.

پروتکل های ارتباط سریال - در این بخش دو پروتکل ارتباطی سریال UART به عنوان روش غیرهمگام و I2C به عنوان روش همگام معرفی می‌شوند.

- **پروتکل ارتباطی UART^۴ :** پروتکل ارتباطی UART یک پروتکل ارتباطی سریال غیر همگام است که هر فریم از داده را در قالب یک بسته ارسال می‌کند. در حالت بیکار (idle)، مقدار ۱ بر روی خط ارتباطی قرار دارد و 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 و فرستنده هم پس از ارسال بیت صفر، به ترتیب از بیت کم ارزش شروع به ارسال فریم داده (۸ بیت) می‌کند. برای نشان دادن پایان ارسال نیز از یک یا دو بیت با مقدار ۱ استفاده می‌شود. همچنین برای کشف خطا بیت توازن در انتهای فریم داده ارسال می‌شود. (شکل ۱)



شکل ۱. بسته و نحوه فریم بندی داده در پروتکل ارتباطی UART به روش غیرهمگام

- **پروتکل ارتباطی I2C^۵ :** پروتکل ارتباطی سریال I2C یک پروتکل ارتباطی دو سیمه سریال همگام به روش تعریف Master/Slave است. از ویژگی‌های این پروتکل می‌توان به قابلیت اتصال چند Slave به یک Master و قابلیت استفاده از چند Master برای کنترل یک یا چند Slave اشاره کرد. این ویژگی‌ها سبب شده که این پروتکل از پر استفاده ترین پروتکل‌ها برای برقراری ارتباط بین المان‌های دیجیتال مثل

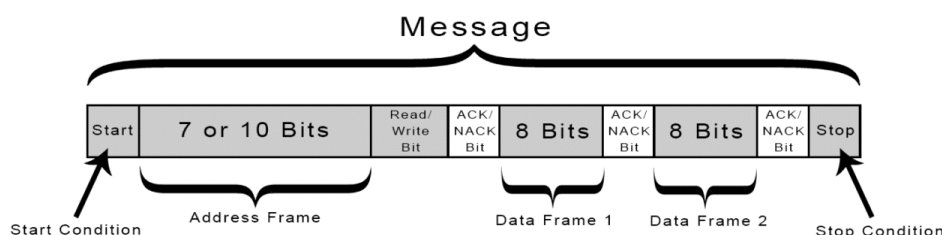
^۳ Synchronous

^۴ Universal Asynchronous Serial Receiver And Transmitter

^۵ Inter Integrated Circuit

میکروکنترلرها با یکدیگر و ارتباط آنها با انواع آی سی ها، حافظه ها، ماژول های فرستنده و گیرنده، سنسورها، نمایشگرها و غیره (بنابراین آشنایی با آن اهمیت زیادی برای پروژه های پایانی شما دارد).

در ارتباط I2C از دو خط به نام های SDA^۶ برای ارسال / دریافت داده و SCL^۷ برای سیگنال کلاک استفاده می شود. این دو خط همگام بوده و کلاک توسط Master تنظیم می شود. در ابتدای ارسال هر پیام، 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. سپس آدرس Slave به عنوان طرف گیرنده روی خط 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) و می توان یک یا چند فریم داده را به آن ارسال نمود. بیت Read/Write نشان دهنده ی این است که آیا گره Master میخواهد داده ای را ارسال کند و یا داده ای از Slave دریافت کند. همچنین برای تشخیص خطا بعد از هر یک از فریم های داده یک بیت ACK/NACK قرار گرفته است که گیرنده با یک کردن آن، دریافت داده ها را تایید خواهد کرد. در نهایت برای نمایش پایان انتقال، خط SCL از صفر به یک و SDA هم به همین صورت تغییر می کنند و نشان دهنده ی پایان پیام است. (شکل ۲)



شکل ۲. بسته داده در پروتکل ارتباطی همگام I2C

➤ **ابزارهای توسعه کد و شبیه سازی** - در این تمرین برای شبیه سازی همانند تمرین قبل از Proteus استفاده می شود اما به جای Arduino IDE، PlatformIO در VSCode^۸ مورد استفاده قرار می گیرد.

- **معرفی PlatformIO : PlatformIO** یک بستر مشارکتی حرفه ای متن باز^۹ مبتنی بر پایتون برای توسعه اینترنت اشیا^{۱۰} (IoT) و سیستم های تعبیه شده^{۱۱} است. PlatformIO یک پلتفرم IDE متقابل با یک دیباگر یکپارچه^{۱۲} است که بر روی ویندوز، مک و لینوکس راه اندازی می شود. این بستر دارای ویژگی هایی از جمله

^۱ Serial Data

^۷ Serial Clock

^۸ Visual Studio Code

^۹ Open Source

^{۱۰} Internet Of Things

^{۱۱} Embedded System

^{۱۲} Integrated Debugger

امکان توسعه سریع کد (مانند کد های اجرایی C / C++) و همچنین ابزار هوشمند بررسی و آنالیز کد برای توسعه سریع و حرفه ای است که در آردوینو IDE وجود ندارد و همچنین مجهز به امکان موقعیت یابی و پیمایش کد ها^{۱۳} و قالب بندی کد است. از جمله ویژگی های اصلی آن می توان به ساختار چندین پلتفرمی، مدیریت کتابخانه، مانیتور پورت سریال و غیره اشاره نمود. PlatformIO از مجموعه متنوعی از [پلتفرم](#)، [فریم ورک](#) و [بورد](#) از جمله Arduino پشتیبانی می کند و با تعدادی [مثال](#) و [کتابخانه](#) همراه است که لیست آن ها در هر لینک قابل مشاهده است.

- **افزونه PlatformIO در VSCode :** همان طور که می دانید ویژوال استودیو کد (VSCode) یک ویرایشگر متن شناخته شده با تعدادی افزونه است که امکان توسعه و پیشرفت کد با زبان های مختلف برنامه نویسی را فراهم می کند. قابلیت استفاده از PlatformIO به عنوان یک افزونه^{۱۴} در ویژوال استودیو کد وجود دارد که با استفاده از [لینک](#) زیر می تونید آن را به سادگی در محیط VSCode نصب نمایید.

➤ شرح تمرین

به طور کلی شما در این تمرین باید با استفاده از دو بورد Arduino Uno، یک شبکه شامل یک گره مرکزی و یک گره دیگر که به یک سنسور متصل است را ایجاد و شبیه سازی کنید. این گره غیر مرکزی با گره مرکزی در ارتباط است و داده هایی که از سنسورهای خود دریافت می کند را به آن ارسال می کند. وظیفه ی گره مرکزی پردازش اطلاعات دریافتی و نمایش تصمیم اتخاذ شده روی یک LCD است.

شرح هر کدام از گره ها که با استفاده از یک بورد آردوینو ایجاد می شوند به صورت زیر است:

- **گره غیر مرکزی (TH-Board) -** این گره در داخل یک گلدان قرار دارد و اطلاعات رطوبت خاک و دما را تولید میکند. برای این کار، بورد مربوطه از طریق ارتباط I2C به یک [سنسور اندازه گیری دما و رطوبت \(SHT25\)](#) متصل می شود و اطلاعات سنسور را بعد از پردازش از طریق ارتباط سریال UART به [ماژول بلوتوث \(HC-05\)](#) ارسال می کند. این ماژول بلوتوث در ادامه اطلاعات را به ماژول بلوتوث گره مرکزی ارسال خواهد کرد. ارسال داده هر ۵ ثانیه یکبار انجام میشود.

- **گره مرکزی (Main-Board) -** این گره که گره مرکزی در این پروژه است، تصمیم می گیرد که بر اساس شرایط، آبیاری قطره ای صورت بگیرد یا خیر و اگر جواب مثبت است، این کار را با چه نرخی انجام دهد. این گره به یک LCD مجهز است و اطلاعات سنسور را دریافت می کند و بسته به شرایط، پیام مناسب را روی LCD چاپ می کند. همچنین این گره به یک موتور DC متصل است که وظیفه کنترل ورودی آب به گلدان را دارد. دقت کنید که این گره نیاز به یک ماژول بلوتوث هم دارد که داده ها را از گره دیگر دریافت کند.

برنامه‌ی شما باید دما و رطوبت دریافتی را روی LCD نمایش داده و تحت شرایط زیر، خروجی‌های متناسب را تولید کند:

- اگر رطوبت بالای ۵۰ درصد بود، آبیاری صورت نگیرد.
- اگر رطوبت کمتر از ۲۰ درصد بود، آبیاری با نرخ ۲۰ سی‌سی بر دقیقه انجام گیرد. برای اینکار باید پالس‌هایی با $duty\ cycle$ ۲۵٪ به موتور ارسال گردد.
- اگر رطوبت بین ۲۰ تا ۵۰ درصد بود نیز دو حالت رخ می‌دهد. اگر دما کمتر از ۲۵ درجه‌ی سلسیوس بود، آبیاری لازم نیست. اگر دما بیش‌تر از ۲۵ درجه‌ی سلسیوس بود، آبیاری با نرخ ۱۰ سی‌سی بر دقیقه صورت گیرد ($duty\ cycle$ پالس‌ارسالی به موتور: ۱۰٪).

➤ نکته‌های پیاده‌سازی

- قبل از شروع پیاده‌سازی پروژه حتما نحوه کارکرد پروتکل‌های [I2C](#) و [UART](#) را به صورت دقیق مطالعه کنید. (تسلط شما بر این دو پروتکل در زمان تحویل پروژه سنجیده می‌شود).
- برای شبیه‌سازی همانند تمرین قبل از Proteus استفاده می‌کنیم اما این‌بار به جای استفاده از Arduino IDE از PlatformIO استفاده می‌کنیم. به منظور نصب PlatformIO بر اساس دستورالعمل‌های لینک [PlatformIO](#) عمل کنید. دقت کنید که این برنامه به صورت یک plugin برای ویرایشگر VSCode در دسترس است.
- برای ایجاد هر گره از بورد Arduino Uno استفاده می‌شود. در PlatformIO برای هر یک از دو گره، یک پوشه با نام آن ایجاد کنید و آن‌ها را در پوشه‌ی اصلی پروژه‌ی خود قرار دهید.
- در ادامه، جزئیات مربوط به هر یک از دو بورد بیان شده است:
 - **TH-Board:** این گره با استفاده از پروتکل I2C به سنسور SHT25 متصل است، برای این‌کار تنها کافی است دو پورت SCL و SDA را به پورت‌های متناظر سنسور متصل کنید. در بورد Arduino Uno پورت‌های SCL و SDA با پورت‌های ورودی آنالوگ مشترک است. برای آشنایی با نحوه‌ی کار با پروتکل I2C در آردوینو که توسط [کتابخانه Wire](#) انجام می‌گیرد می‌توانید به لینک مراجعه کنید.
 - 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 به خروجی می‌دهد.
 - این بورد باید دو نوع داده دما و رطوبت را برای گره مرکزی ارسال کند. برای این‌کار پروتکل مشخصی وجود ندارد. می‌توانید با استفاده از خلاقیت خود راهکاری برای این‌کار پیشنهاد دهید. برای مثال یک ایده می‌تواند استفاده از delimiter برای جداسازی انواع داده باشد.
 - **Main-Board:** گره اصلی وظیفه‌ی گرفتن داده‌های ارسال شده از گره و پردازش و تصمیم‌گیری بر اساس آن‌ها و نمایش اطلاعات و تصمیم مرتبط با آن‌ها روی یک LCD را دارد. برای شروع مازول LM041L با

اندازه‌ی ۴x۲۰ را از لیست کتابخانه‌های Proteus به پروژه اضافه کنید. با استفاده از مستندات موجود در اینترنت، LCD را به Arduino متصل کنید (پایه‌ی En را 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 LCD نمایش دهید. 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.

دقت کنید که برای آبیاری باید از شیرهای (valve) قابل کنترل با ولتاژ استفاده شود، ولی برای راحت تر شدن تمرین، می‌توانید فرض کنید که PWM ساخته شده به یکی از موتورهای DC تمرین قبل ارسال میشود.

- 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).

- ارتباط دو ماژول بلوتوث داده شده در محیط شبیه‌سازی Proteus به صورت مجازی برقرار می‌شود. برای این کار 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.

➤ پرسشی‌ها

۱. در مورد بلوتوث، از چه فرکانسی برای ارتباط بی‌سیم استفاده میشود؟ در صورت وجود چند دستگاه بلوتوث در اطراف هم، چگونه از تداخل داده‌های ارسالی دستگاه‌ها جلوگیری میشود؟ نیازی به آرایه جزییات پروتوکل ارتباطی بلوتوث نیست. بیان مفاهیم کلی کافی است.

۲. اگر نیاز به اتصال چند سنسور مختلف که از پروتکل I2C استفاده می‌کنند باشد آیا می‌توان همه را به تنها پرت موجود I2C در AVR متصل کرد؟ در این صورت، چگونه تضمین میشود که داده‌های ارسال آنها با هم تداخل نمیکند؟

۳. طراحی مفهومی این تمرین را مانند نمونه‌ای که در تمرین اول دیدید رسم کنید و تنها اتصالات و اجزای اصلی را نمایش دهید.

➤ نکته های تحویل تمرین

- این تمرین تحویل حضوری دارد و باید در قالب گروه های چهار نفره انجام شود. موعد تحویل تمرین در صفحه درس اعلام شده است.
- در کنار یادگیری مفاهیم بیان شده ، نحوه کدنویسی خوانا (ماژولار بودن کد، استفاده از نامگذاری مناسب، استفاده درست از توابع و ...) در این تمرین اهمیت بالایی دارد.
- گزارش کار کاملی از مراحل انجام تمرین ، نتایج شبیه سازی، تصاویر و نکات هر بخش و پاسخ به سوالات تهیه شود.
- پیشنهاد می شود هر چه زودتر پیاده سازی ها را شروع کنید و قسمت های مختلف این تمرین بین افراد گروه تقسیم شود. البته همه ی افراد باید تسلط کافی به مباحث تمامی بخش های پروژه داشته باشند.
- اگر پروژه تان موفقیت آمیز بود، میتوانید روی تجاری سازی آن فکر کنید!! (لیست گلدانهای هوشمند دیجیکالا)

موفق باشید :