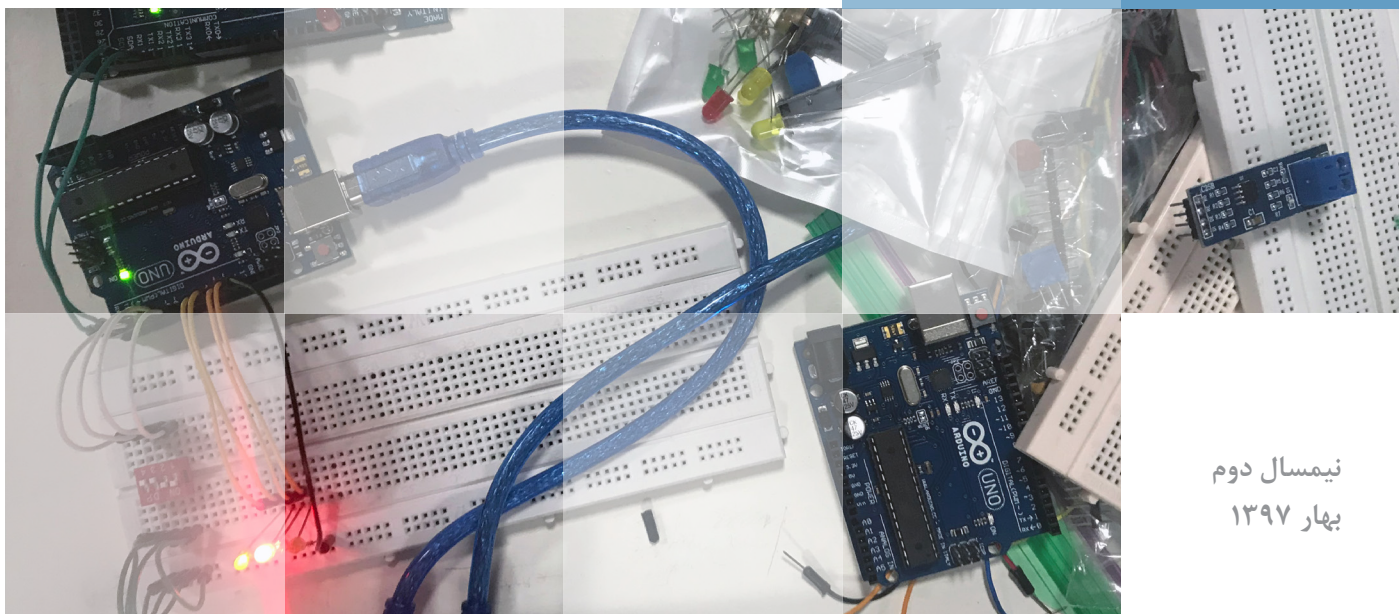


## مبانی سیستمهای نهفته و بی درنگ

طراحی و پیاده سازی بخشی از یک سیستم مدیریت ساختمان برای کنترل کلیدهای هوشمند



نیمسال دوم  
بهار ۱۳۹۷



#### دانشگاه شهید بهشتی

دانشجو: مهسا لایقی

استاد: دکتر سید حسین عطارزاده نیاکی

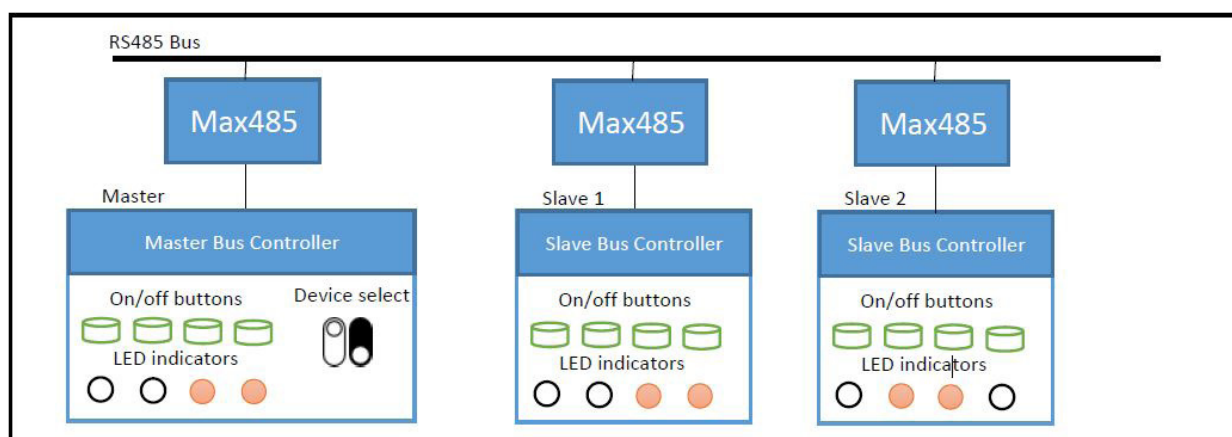
دستیار آموزشی: نسترن جان نثار

## فهرست

- ۰۴ مقدمه
- ۰۵ طرح مسئله
- ۰۶ روند پیشرفت پروژه
- ۰۷ ماشین حالت

## مقدمه

## مبانی سیستمهای نهفته و بی درنگ

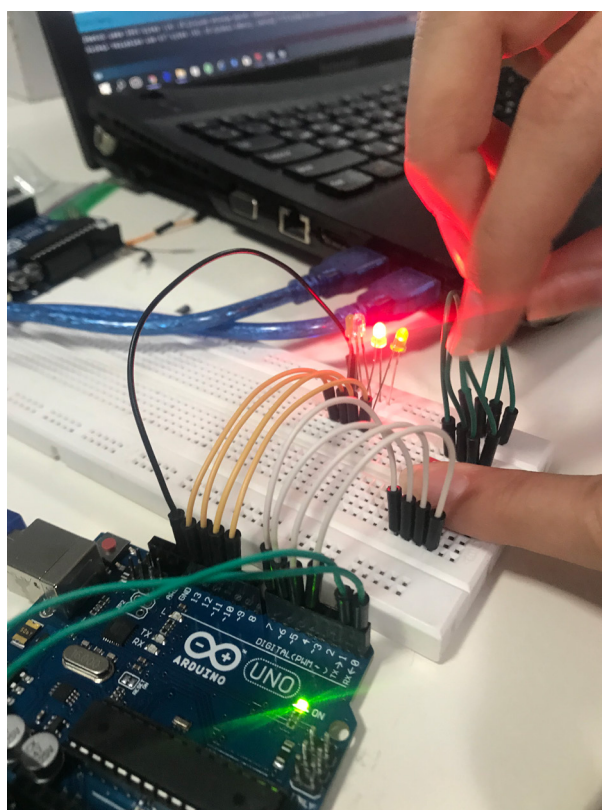


شمای کلی سیستم به این صورت است که یک گره مرکزی اصلی (Master) تعدادی کلید هوشمند (Slave) را که از طریق یک باس (RS485) به هم متصل شده اند کنترل می کند. کلیدهای هوشمند میتوانند تک پل و یا دارای تعداد بیشتری پل باشند. پل های مختلف این کلیدها را می توان جداگانه از طریق دکمه های محلی و یا توسط گره مرکزی از طریق باس روشن و خاموش کرد. گره مرکزی دارای سوئیچ هایی است که می توان با آن کلید هوشمند مورد نظر را انتخاب کرد (Device select). پس از انتخاب کلید هوشمند مورد نظر وضعیت کلید پیوسته از طریق باس خوانده شده و به صورت محلی نمایش داده می شود. به علاوه، پس از انتخاب یک کلید خاص در گره مرکزی میتوان با استفاده از کلیدهای محلی وضعیت روشن و خاموش پل های کلید انتخاب شده را تغییر داد.

باس (RS485) مورد استفاده بین گرهها (Multi-dropped) و یک طرفه است به این معنی که تمام ادوات به باس متصل هستند و در هر لحظه یک گره باس را در اختیار دارد (Half-duplex). چپ هایی نظیر Max 485 و ماژول های آماده ای که بر اساس آنها ساخته شده مانند (+ و +) امکان تبدیل سطح ولتاژهای عادی TTL UART را به سیگنال های باس RS485x در هر گره متصل به باس دارند.

## طرح مسئله

پروژه تعریف شده برای این درس شامل طراحی و پیاده‌سازی بخشی از یک سیستم مدیریت ساختمان برای کنترل کلیدهای هوشمند است که در گروه‌های دو نفری انجام و تحویل داده می‌شود. در پیاده‌سازی فرض بر این است که از بردهای **Arduino** استفاده می‌شود. بخشی از پروژه شامل مدلسازی، شبیه‌سازی و پیاده‌سازی سیستم پایه و نوشتن گزارش اجباری بوده که مکمل نمره‌ی نهایی است ولی بخش‌های اختیاری دیگری نیز مشخص شده که به عنوان نمره‌ی اضافه در نظر گرفته شده است.



## روند پیشرفت پروژه

در این پروژه قصد داشتیم که دکمه های محلی یک Slave یا کلید هوشمند را از طریق خودش و نیز از طریق Master روشن و خاموش کنیم.

در ابتدا برای شروع سعی کردم یک ارتباط UART ساده را پیاده سازی کنم به این ترتیب که از Master داده ای را به Slave می فرستادم و از آن نیز جوابی دریافت می کردم.

در ابتدای کار باسی را برای ارتباط Master و Slave در نظر نگرفتم و تنها RX مستر را به TX اسلیو و TX مستر را به RX اسلیو وصل کردم.

چالش هایی که در این مرحله با آن مواجه شدم این بود که داده دریافتی `int` بود و من می خواستم آن را بایت بایت بفرستم. ابتدا همان `int` را فرستادم که مشکل ساز می شد. پس از گذراندن این مرحله شروع به پیاده سازی پروتکل ذکر شده در صورت پروژه شدم.

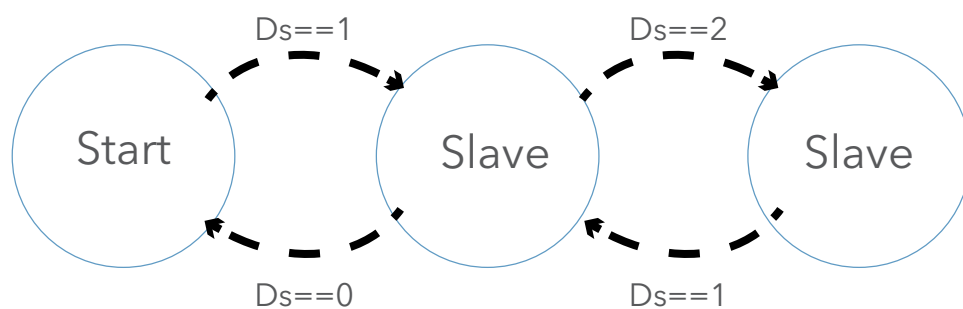
در این مرحله در Master ۳ فریم را برای Slave ارسال کردم. همانطور که در پروتکل توضیح داده شده بود فریم اول آدرس دستگاه را مشخص می کرد. من برای اسلیو آدرس ۵۰ را در نظر گرفتم در صورتی که در سریال مانیتور عدد ۵۰ انتخاب می شد مستر باید از طریق سریال ۱ چیزی برای اسلیو ارسال می کرد در غیر اینصورت اگر این آدرس عدد ۱۰ یعنی همان آدرس مستر می بود، مستر کلید های خودش را کنترل می کرد.

در فریم بعدی مشخص می شد که مستر عمل `write` انجام می دهد یا `Read`. (از طریق چک کردن بیت پر ارزش) اگر عدد دریافتی کوچکتر از ۱۲۷ می بود یعنی مستر قصد `write` کردن یا به عبارت دیگر تغییر دکمه های اسلیو را دارد. در غیر اینصورت اسلیو باید وضعیت کلید های خودش را اطلاع می داد.

در فریم آخر اگر مستر قصد `write` داشت باید مشخص می کرد که کدام دکمه ها را می خواست روشن یا خاموش کند.

در تمامی این مراحل مشکلی که با آن بیشتر از همه مواجه شدم این بود که نمیتوانستم وضعیت کلید ها را وقتی که مستر درخواست `Read` داشت برگردانم.

پس از اینکه این مراحل را به درستی پشت سر گذاشتم سعی کردم که `Max 485` را وصل کنم و دیگر مستر و اسلیو مستقیماً با هم در ارتباط نباشند. که در اینجا پس از اتصالات دیگر انگار پیامی از مستر به اسلیو ارسال نمی شد.



`Serial_available() < 6`

