

# آزمایش ۸

آزمایشگاه ریزپردازنده  
نیم سال دوم ۱۴۰۱-۱۴۰۰

## هدف

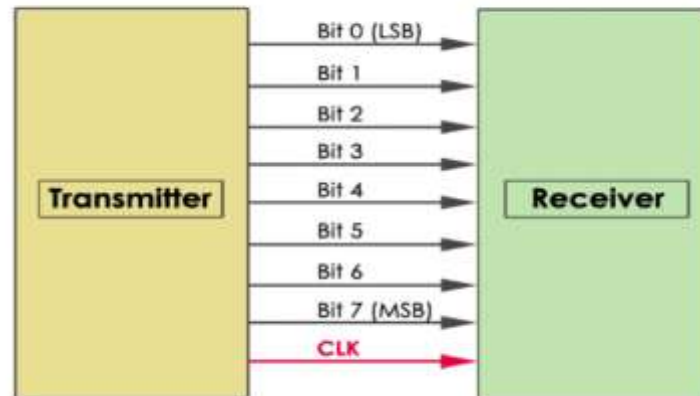
هدف از این آزمایش آشنایی با مفهوم ارتباط سریال از طریق درگاه UART و شیوه راه اندازی آن در میکروکنترلر STM32F401RE است.

## پیش نیاز و مطالعه

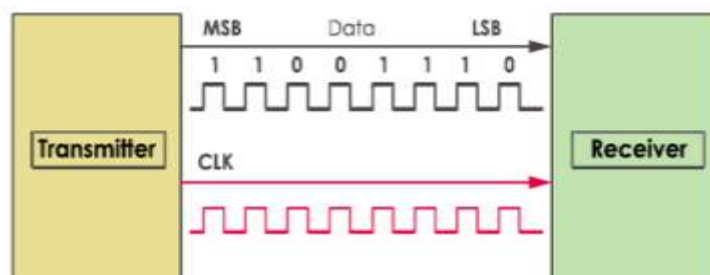
آشنایی با مفهوم ارتباط سریال از طریق USART/UART

در علم کامپیوتر فرآیند ارسال اطلاعات به دو روش سریال و موازی صورت می گیرد. در ارسال موازی دیتای ارسالی از طریق چندین کابل ارتباطی به صورت همزمان ارسال می شود که مزیت آن سرعت بالای انتقال داده است اما عیب آن هزینه کابل بسیار بالای آن است. در ارسال سریال اطلاعات به صورت بیت به بیت ارسال می شود.

### Parallel Communication



### Serial Communication



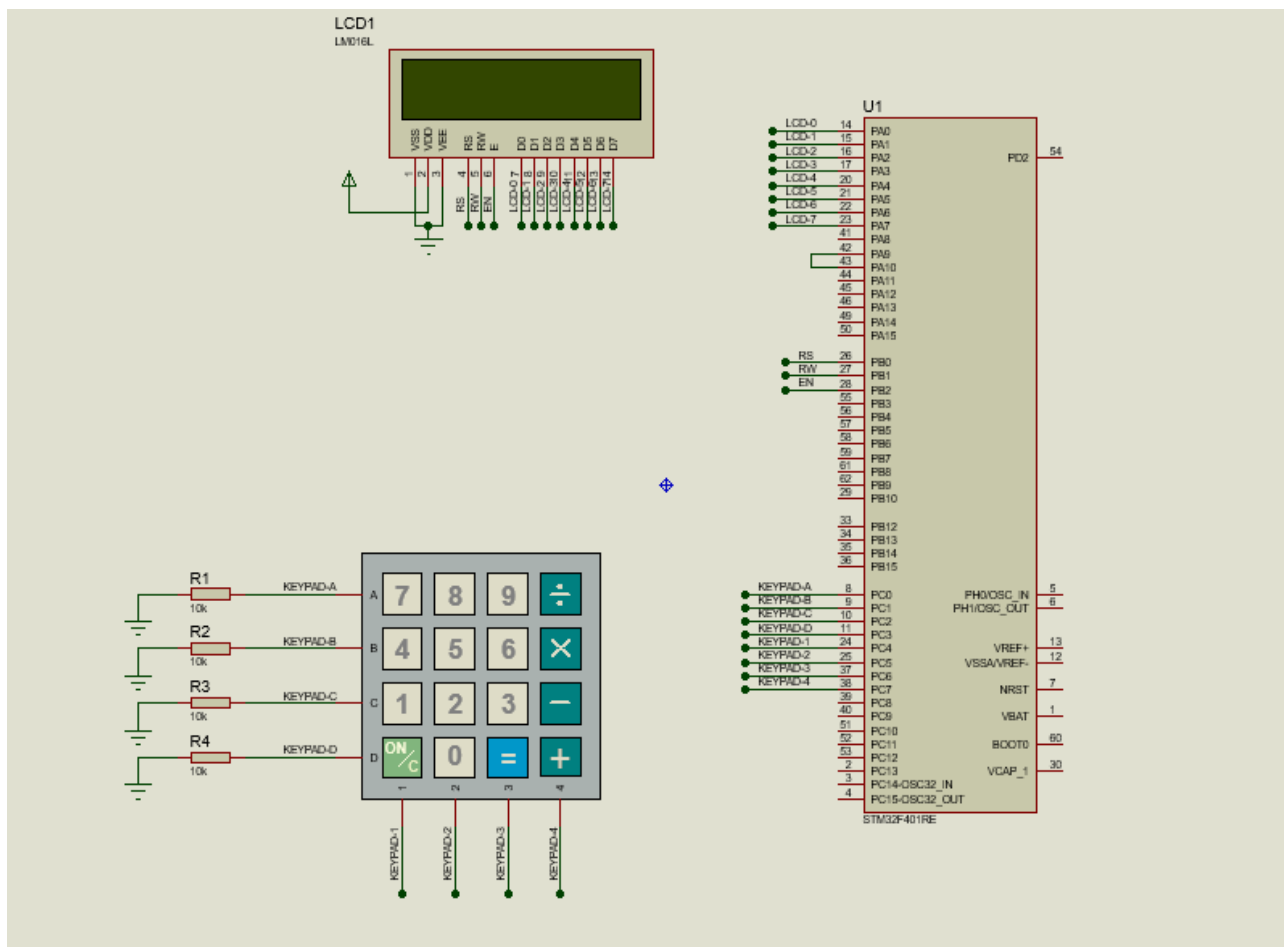
برخی از باس‌های سریال که در میکروکنترلرها مورد استفاده قرار می‌گیرند عبارتند از، SPI، I2C، DC\_BUS، UIN/O و USART. یکی از رابط‌های سریالی که در این آزمایش به آن پرداخته می‌شود درگاه UART(Universal Asynchronous Receiver-Transmitter) است. با به‌کارگیری این درگاه می‌توان داده‌ها را به‌صورت سریال برای یک میکروکنترلر دیگر ارسال و یا از آن دریافت کنیم. همچنین می‌توان با راه‌اندازی پورت سریال کامپیوتر، از طریق پایانه پورت سریال به تبادل اطلاعات با میکروکنترلر پرداخت.

## سوالات تحلیلی

۱. مفهوم ارتباط سریال همزمان (سنکرون) و غیر همزمان (آسنکرون) را شرح دهید.
۲. منظور از Baud Rate چیست؟ و مقدار آن در میکروکنترلر STM32F401 به چه پارامترهایی بستگی دارد و چگونه محاسبه می‌شود؟
۳. پایداری در برابر نویز در خطوط انتقال موازی بیشتر است یا انتقال سریال؟ چرا؟ برای مشکل نویز در درگاه UART چه راه‌حل‌هایی اندیشیده شده است؟

## دستور کار

۱. در این آزمایش قصد داریم با به‌کارگیری درگاه UART موجود در میکروکنترلر STM32F401RE یک دماسنج طراحی کنیم.  
سیستم از بخش‌های زیر تشکیل شده است:
  - میکروکنترلر STM32F401RE
  - ترمینال پورت سریال (virtual terminal در پروتئوس)
  - سنسور دما (Im35 در پروتئوس)سیستم به این صورت عمل می‌کند که میکروکنترلر STM32F401RE مقدار دما را که به یکی از پایه‌های مرتبط به مبدل آنالوگ به دیجیتال آن متصل است را می‌خواند، سپس آن را از طریق درگاه UART برای ترمینال پورت سریال ارسال می‌کند.
۲. در این آزمایش قصد داریم با به‌کارگیری قابلیت UART موجود در میکروکنترلر STM32F401RE یک ماشین حساب طراحی کنیم.  
سیستم از بخش‌های زیر تشکیل شده است:
  - میکروکنترلر STM32F401RE
  - LCD کاراکتری ۱۶\*۲ (LM016L در پروتئوس)
  - Keypad ماتریسی ۴\*۴عملکرد سیستم به‌صورت زیر است.
  - ❖ دستورات محاسباتی دریافت شده از Keypad، از طریق UART ارسال می‌شوند.
  - ❖ دستورات دریافت شده از طریق UART همراه با نتیجه محاسبات بر روی LCD نمایش داده می‌شوند.
  - ❖ دریافت داده را با وقفه دریافت انجام دهید. (به‌کارگیری polling مجاز نیست).
  - ❖ برای اتصال Keypad به میکروکنترلر می‌توانید از هر یک از روش‌های polling و یا وقفه استفاده کنید.
  - ❖ توجه شود که باید پایه‌های RX و TX میکروکنترلر STM32F401RE به هم متصل شوند.
  - ❖ شکل زیر شمای کلی سیستم را در شبیه‌ساز پروتئوس نشان می‌دهد که از USART1 استفاده شده است که در آن PA9 پایه TX و PA10 پایه RX است.



برخی تعاریف و توابع مورد نیاز به قرار جدول زیر است:

UART_HandleTypeDef huart1	ساختمان داده مربوط به UART که به طور خودکار توسط نرم افزار Cube تولید شده و به عنوان پارامتر ورودی به توابع مربوط به UART داده می شود.
HAL_UART_Transmit(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size, uint32_t Timeout)	این تابع برای ارسال اطلاعات ذخیره شده در pData مورد استفاده قرار می گیرد. آرگومان size تعداد المان های ارسالی یا همان المان های ذخیره شده در pData را مشخص می کند. آرگومان Timeout نیز حداکثر زمان برحسب میلی ثانیه است که صرف ارسال داده ها می شود.
HAL_UART_Transmit_IT(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size)	مشابه تابع قبلی است با این تفاوت که مکانیزم عملکرد این تابع non-blocking است و پس از اتمام ارسال با فعال سازی وقفه پایان عملیات را اعلام می کند.

HAL_UART_Receive(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size, uint32_t Timeout)	به تعدادی که توسط آرگومان Size مشخص شده است، المان دریافت کرده و آن‌ها را در pData ذخیره می‌کند. آرگومان Timeout حداکثر زمانی است که صرف دریافت داده‌ها می‌شود.
HAL_UART_Receive_IT(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size)	مشابه حالت قبل است اما از مکانیزم وقفه استفاده می‌کند.
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart) { if(huart->Instance == USART1) { } }	هنگامی که به تعداد مشخص شده، داده توسط UART دریافت شود، این تابع فراخوانی می‌شود. سرویس‌دهنده وقفه دریافت را می‌توان در قالب این تابع پیاده‌سازی کرد.

## الزامات

- تنها به کارگیری کتابخانه CMSIS در بخش یک مجاز است و به کارگیری توابع HAL مجاز نیست.
- به کارگیری هر دو کتابخانه CMSIS و HAL در بخش دوم مجاز است.

## موارد تحویل دانی

- سورس کد تمام بخش‌های ذکر شده را به صورت کامل تحویل دهید. برای خوانایی بیشتر باید بخش‌های مختلف کد کامنت‌گذاری شود.
- پروژه ساخته شده در Keil، Proteus و STM32CubeMX را باید تحویل دهید.
- گزارشی کامل و روشن از بخش‌های مختلف انجام شده در طی اجرای دستور کار تحویل شود. اگر در بخشی قطعه کدی توضیح داده می‌شود، باید یک کپی از آن بخش کد در گزارش آورده شود.
- شماره پین‌ها و پورت‌های به کار گرفته شده به همراه نوع تنظیماتی که برای آن لحاظ شده است در گزارش بیان شود.
- تمام موارد بالا در قالب یک فایل فشرده در سامانه درس‌افزار بارگذاری شود.
- علاوه بر آماده‌سازی شبیه‌سازی برای این آزمایش، دانشجویان موظف‌اند برای پیاده‌سازی عملی بخش ۲ نیز به صورت حضوری آمادگی داشته باشند.

## نکات حائز اهمیت

- بخش‌های مختلفی که باید تحویل داده شوند همگی در یک فایل فشرده باشند و نام فایل فشرده به فرمت زیر باشد:  
<گروه درسی-نام-نام خانوادگی-شماره دانشجویی>
- به ازای هر روز تأخیر، روز اول ۱۵٪، روز دوم ۲۵٪ و روزهای سوم و چهارم ۳۰٪ از نمره کسر خواهد شد و در روز پنجم نمره‌ای تخصیص نمی‌گردد.
- دقت شود که در گزارش نام اعضا، شماره دانشجویی و گروه درسی ذکر گردد.
- آزمایش‌های ریزپردازنده به صورت گروه‌های دوفره انجام داده شده و تحویل می‌شوند.
- نکته مهم این است تمامی افراد گروه باید به همه جوانب و جزئیات آزمایش‌ها مسلط باشند که این نکته توسط مدرسین هنگام تحویل به دقت بررسی خواهد شد.

- هر گروه باید به صورت مجزا آزمایش را انجام دهد و کپی نتایج آزمایش گروه‌های دیگر تحلف است.
- به منظور ایجاد شرایط یکسان برای تمامی گروه‌ها و فاصله داشتن زمان آپلود و تحویل، به هنگام تحویل، اعضای گروه، در همان زمان پاسخ آزمایش خود را از درس‌افزار داندلود کرده و روی سیستم خود تحویل می‌دهند.

موفق باشید

گروه آزمایشگاه‌های ریزپردازنده