

نکات و قوانین آزمایشگاه ریزپردازنده:

- فقط از طریق تکلیف مربوطه در سامانه VU و لینک‌های اعلام شده مجاز به ارسال هستید.
- فایل‌های پروژه خود را در یک فایل zip قرار دهید و آن را به شکل زیر نام‌گذاری کنید:

1) Core (Folder)

2) Project\_name.ioc (CubeMX Project)



Name\_StudentNumber\_MinPrj\_T#.zip

بعد از T شماره گروه خود را قرار دهید که معادل با شماره پک دریافتی شماست.

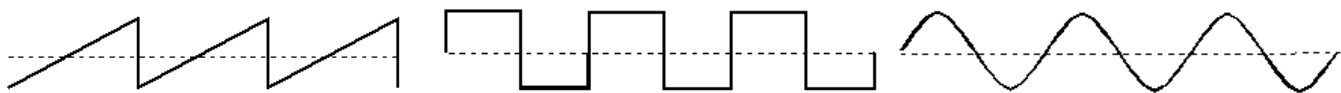
فایل‌های بالا در دایرکتوری Workspace که در CubeIDE ساختید قرار دارند و به صورت پیش فرض در آدرس زیر قرار دارد:

C:\Users\{Username}\STM32CubeIDE\workspace\_{Version}\{Project\_name}

- در صورت مشاهده و اثبات هرگونه **تقلب** و شباهت در کدها نمره طرفین **100%-** در نظر گرفته خواهد شد.
- ارسال توسط یکی از اعضا گروه کافی می‌باشد.
- تحویل بر اساس کد آپلود شده است و در صورت مشاهده **مغایرت** در کد تحویلی و کد آپلود شده نمره 0 به آن تسک تعلق خواهد گرفت.

### مینی پروژه خانه هوشمند:

در این مینی پروژه قصد داریم امکانات دماسنج و نورسنج محیط یک خانه هوشمند را شبیه سازی کنیم. شدت نور و دمای خانه نیز بر روی seven segment نشان داده می‌شود. دو رقم سمت راست seven segment بیانگر شدت نور (از ۰ تا ۹۹) و دو رقم سمت چپ نیز بیانگر دمای محیط به درجه سلسیوس می‌باشد. در این خانه، اگر نور از آستانه ای که برای شما ارسال می‌شود کمتر شد، یا آنکه دمای خانه نیز از آستانه ای که برای شما ارسال می‌شود بیشتر شد، buzzer شروع به پخش پیغام خطر می‌کند. نوع پیغام خطر نیز یک بوق با شدت ثابت می‌باشد که فرکانس آن متغیر بوده و به سه شکل مثلثی، مربعی یا سینوسی می‌باشد.



اطلاعات باید از طریق uart به کمک فریم هایی شبیه فریم های پروتکل ethernet (که در ادامه در موردشان توضیح داده خواهد شد) به کامپیوتر ارسال شده و به کمک برنامه ای که در اختیار شما قرار خواهد گرفت، شدت دما و نور را نشان بدهد. همچنین شما می توانید آستانه حساس برای روشن شدن led های خارجی با توجه به LDR و پیغام خطا برای سنسور LM35 را مشخص کنید. این اطلاعات نیز با فریم هایی با پروتکل شبیه ethernet برای میکرو ارسال می شوند. لذا شما باید هم توانایی دیکود هم انکود کردن فریم ها را داشته باشید.

1 byte (0xAA)	1 byte	1 byte	1 byte	2 – 255 byte	1 byte
Preamble	Dest. Addr.	Src. Addr.	Data len.	Data	CRC

شروع فریم با preamble است که نشان دهنده ی آغاز فریم است و مقدار آن برابر با 0xAA می باشد. پس از آن آدرس مقصد و سپس آدرس مبدأ قرار می گیرند. بایت بعدی نیز length یا طول بخش data است که نشان دهنده ی تعداد بایت های موجود در بخش data می باشد و در نتیجه می تواند عددی بین 0 تا 255 باشد. اما از آنجا که تعداد 0 برای بایت های داده کاربردی نداشته و همچنین داده های ارسالی ما حداقل باید دارای دو بایت داده باشند، این عدد باید چیزی بین 2 تا 255 باشد. بایت آخر بعد از data نیز CRC می باشد که برای کنترل خطا استفاده می گردد. در نتیجه طول مجموع پکت ها باید بین 7 تا 260 بایت باشد. چند نکته در مورد فریم ما وجود دارد؛ یکی آنکه Destination Address و Source Address در این مینی پروژه عملاً هر مقداری می توانند داشته باشند ولی مقدار داشتن آنها اهمیت دارد. دیگر آنکه برای پکت هایی که از میکرو به سیستم ارسال می شود، در بخش data بایت اول بایت کنترل می باشد که نشان دهنده نوع داده (دما یا نور) است که برای آن مقدار 0x21 برای نور و مقدار 0x22 برای دما استفاده شده و بایت دوم بیانگر مقدار دما یا نور است که برای دما همان درجه سلسیوس و برای نور درصد آن یا عددی بین 0 تا 99 قرار می گیرد. دوره ارسال این پکت ها را هر یک ثانیه در نظر بگیرید. به شکلی که هر یک ثانیه دو پکت نور و دما پشت سر هم ارسال شوند.

پیاده سازی CRC در ارسال و دریافت (محاسبه و قرار دادن crc در انتهای پکت در زمان ارسال و چک کردن و بررسی خطا با کمک crc در زمان دریافت) دارای نمره اضافه است. در صورت عدم پیاده سازی مقدار آن را در پکت خود تمام صفر قرار دهید.

برای محاسبه CRC از divisor یا generator معادل با 0x19A که فرم باینری آن 110011010 است استفاده کنید. و آن را برای تمام پکت به غیر از بخش preamble محاسبه کنید. (برای dest addr و src addr و len و data)

برای آشنایی با نحوه محاسبه CRC می توانید از لینک زیر استفاده کنید:

[https://en.wikipedia.org/wiki/Cyclic\\_redundancy\\_check#Computation](https://en.wikipedia.org/wiki/Cyclic_redundancy_check#Computation)

همچنین پکت هایی از سیستم به میکرو ارسال میشود که تنها یک نوع دارد و آن هم برای تنظیم نحوه عملکرد بوق است. بدین شکل که باید کنترل آن باید مقدار 0x11 بوده و بایت بعدی نشان دهنده نوع بوق می باشد که مقدار 0x31 برای بوق با موج مربعی، 0x32 برای بوق با موج مثلثی و مقدار 0x33 برای بوق با موج سینوسی می باشد. بایت بعدی نشان دهنده حد پایین فرکانس پخش شده و بایت بعد از آن نیز

حد بالای فرکانس پخش شده می باشد که هر دو می توانند عددی بین 200 تا 4000 باشد. در نتیجه این نوع از پکت داری بخش data با طول 4 بایت است. زمان پخش هر دوره از بوق نیز دلخواه است.

برای تعیین آستانه های نور و دما نیز پکت هایی برای شما ارسال می گردد که بدین صورت است؛ بایت کنترل 0x41 برای دما و 0x42 برای نور می باشد. بایت دوم نیز تعیین کننده آستانه است که برای دما به درجه سلسیوس و برای نور به درصد می باشد.

دقت داشته باشید که برنامه شما باید طوری نوشته شود که هر زمان هر کدام از این پکت ها را دریافت کردید، تغییرات مربوط به آن را در میکرو اعمال کنید و تنظیم شدن موارد مختلف از قبیل حالت بوق و آستانه های نور و دما باید امکان تغییر به هر تعداد و هر زمان را داشته باشند.

پیاده سازی بافر برای دریافت و ارسال پکت ها دارای نمره اضافه می باشد. در غیر این صورت می توانید پردازش پکت های دریافتی را بدون صف بندی و بلافاصله بعد از دریافت انجام دهید و به طور مشابه برای پکت های ارسالی می توانید آن ها را در موعد ارسال ساخته و بلافاصله ارسال کنید و صف بندی پکت ها اجباری نمی باشد.

برای محاسبه دما از مقدار خروجی سنسور LM35 از فرمول زیر استفاده کنید:

$$\text{Voltage\_mv} = \text{raw\_value} * 3300 / 4095$$

$$\text{Temperature\_c} = \text{Voltage\_mv} / 10$$

برای اسکیل کردن دقیق تر مقدار خام LDR به بازه 0 تا 99 می توانید از فرمول زیر استفاده کنید:

$$\text{scaled\_value} = (\text{raw\_value} - \text{min\_raw\_value}) * 99 / (\text{max\_raw\_value} - \text{min\_raw\_value})$$

در فرمول بالا، مقادیر max\_raw\_value و min\_raw\_value به ترتیب بیشترین و کمترین مقادیر خام قابل دستیابی با سنسور نور است. از آنجا که لزوماً خروجی سنسور تمامی اعداد موجود در بازه ۰ تا ۴۰۹۵ را شامل نمی شود، از این فرمول استفاده می کنیم تا با استفاده از مقادیر بیشینه و کمینه ی عملی و نه تئوری، عدد نهایی اسکیل شده که به صورت درصد است را نمایش دهیم. این درصد نسبتی واقعی تر با شرایط محیط دارد و می توانید با فراهم کردن بیشترین و کمترین نور محیطی این مقادیر را بیابید.

طبعاً باید برای اندازه گیری های خود نرمال سازی نیز انجام بدهید تا مقادیر شما کمتر تحت تأثیر نویز قرار گیرند. برای نرمال سازی از روش های میانگین گیری یا دیگر روش ها می توانید استفاده کنید. استفاده از روش های بهینه و خوب برای نرمال سازی شامل نمره اضافه می باشد.

برای بحث نرمال سازی می توانید از دو لینک زیر کمک بگیرید.

[https://www.st.com/resource/en/application\\_note/an1711-software-techniques-for-compensating-st7-adc-errors-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an1711-software-techniques-for-compensating-st7-adc-errors-stmicroelectronics.pdf)

<https://www.electronicdesign.com/technologies/analog/article/21778422/use-software-filters-to-reduce-adc-noise>

## نکات مهم:

۱) در حلقه while در تابع main کدی ننویسید.

۲) دریافت UART می بایست به صورت وقفه ای راه اندازی شود.

۳) پایه VCC بازر، مقاومت حساس به نور، volume و سنسور LM35 را به ۳.۳ ولت وصل کنید.

۴) segment-7 را با کمک آیسی راه اندازی کنید.

۵) برای راه اندازی چند ماژول آنالوگ، برای هر کدام از آنها از یک واحد adc جدا استفاده کنید و از چند کانال یک واحد adc استفاده نکنید.