نکات و قوانین تکالیف آزمایشگاه ریزپردازنده:

- فقط از طریق تکلیف مربوطه در سامانه VU و لینک‌های اعلام شده مجاز به ارسال هستید.
- آپلود تکلیف توسط یکی از اعضای گروه کافی می‌باشد.
- فایل‌های پروژه گروه خود را در یک فایل rar قرار دهید و آن را به شکل زیر با مشخصات یکی از اعضای گروه نام‌گذاری کنید:

1) Core (Folder)

2) Project_name.ioc (CubeMX Project File)

⇩

Name_StudentNumber_S#.rar

مثلاً برای آپلود تکلیف سوم:

AminGhasempour_9612111111_S3.rar

فایل‌های بالا در دایرکتوری Workspace که در CubeIDE ساختید قرار دارند و به‌صورت پیش‌فرض در آدرس زیر قرار دارد:

C:\Users\{Username}\STM32CubeIDE\workspace_{Version}\{Project_name}

- یک کلیپ تا ۱۵ دقیقه از عملکرد برد و توضیح مختصر کد، اتصالات و پیاده‌سازی تهیه کنید که در آن هر کدام از اعضای گروه قسمت‌هایی را که خودش انجام داده توضیح دهد و آن را هم در فایل آرشیو قرار دهید.
- فاز اول تکلیف تحویلی نمی‌باشد و تنها فرستادن موارد خواسته شده برای فاز دوم مورد نیاز است.
- توجه کنید که حداکثر حجم مجاز برای کلیپ 100 MB است و حتماً حجم کلیپ را با نرم‌افزاری مانند Advanced Video Compressor کاهش دهید.
- در صورت مشاهده و اثبات هرگونه تقلب و شباهت در کدها نمره طرفین ۱۰۰%- در نظر گرفته خواهد شد.

فاز ۱:

نام خانوادگی خود را به فارسی روی LCD کاراکتری چاپ کنید.

فاز ۲ (تحویلی):

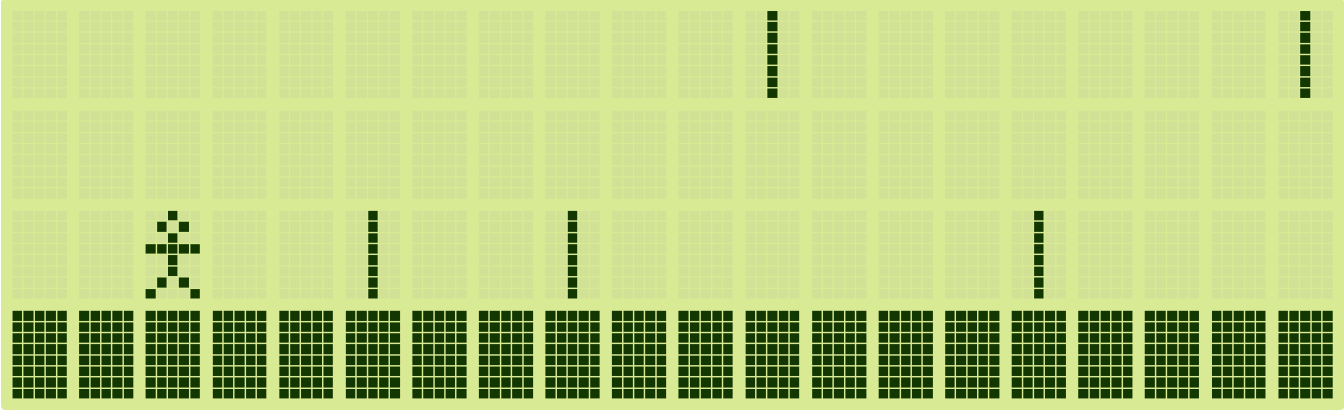
در این فاز قصد داریم نسخه ساده‌تری از بازی دایناسور گوگل کروم را طراحی کنیم. در این بازی دو نوع مانع وجود دارد. مانع اول در ردیف اول قرار دارد و مانع دوم در ردیف سوم از بالا (می‌توانید موانع را به هر شکلی که دوست دارید طراحی کنید مانند درخت، پرنده و ...). ردیف چهارم از بالا نیز همواره تیره است و زمین را نشان می‌دهد. بازیکن در ردیف سوم از بالا و ستون سوم از چپ قرار دارد و با هر بار فشار داده شدن دکمه آبی روی برد دوخانه به سمت بالا می‌پرد. در این بازی موانع به‌صورت تصادفی تولید می‌شوند و لازم است توجه کنید که نحوه تولید موانع به شکلی نباشد که رد شدن از آن‌ها غیرممکن شود و یا فواصل بین آن‌ها آن‌قدر زیاد گردد که بازی چالشی نداشته باشد. برای شمردن امتیاز هر خانه پیشروی بازیکن را یک امتیاز در نظر بگیرید و آن را بر روی 7-Segment نمایش دهید. سرعت پریدن را بر اساس صلاح دید خود، مقدار مناسبی در نظر بگیرید که بازی ممکن و چالش‌برانگیز باشد. سرعت حرکت موانع را ثابت و با فرکانس ۱ (هر یک ثانیه یک خانه به سمت چپ) در نظر بگیرید.

قبل از شروع بازی بر روی LCD با پیغام مناسبی نام فارسی بازی را نمایش دهید و با پایان یافتن بازی امتیاز نهایی را به همراه عبارت مناسب نمایش دهید. در هر دو این حالات که پیغامی بر روی صفحه آمده است با فشار دادن دکمه بازی باید شروع گردد.

توجه داشته باشید که در این بازی باید به موارد زیر توجه کنید:

۱. لازم است برای بازیکن و موانع کاراکترهای جدید طراحی گردد و از آن‌ها استفاده شود.
۲. این بازی تا زمانی که بازیکن نباخته است ادامه دارد و باید دائما به صورت تصادفی موانع مناسبی تولید کند.
۳. نحوه به‌روزرسانی LCD باید به شکلی باشد که تصویر پریدگی نداشته باشد و به‌صورت بهینه‌ای به‌روزرسانی‌ها انجام گردد.

در شکل زیر یک نمونه از صفحه بازی را مشاهده می‌کنید (دقت کنید که این تصویر یک نمونه است و بازیکن و موانع باید متناسب با سلیقه خودتان طراحی گردند):

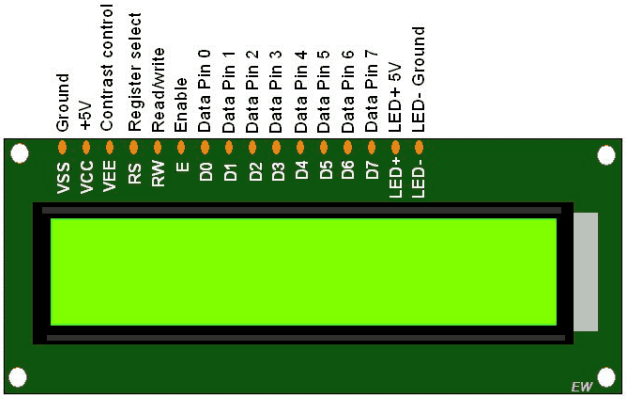


- تولید صدای پریدن با استفاده از بازر و تولید کاملاً تصادفی[1] موانع هر کدام دارای ۵% نمره اضافه می‌باشد.
- برای نوشتن فارسی و ساخت کاراکتر جدید می‌توانید از ابزار Custom Character Generator در وب کمک بگیرید.
- توجه کنید که هم‌زمان حداکثر ۸ کاراکتر جدید می‌توانید روی LCD ذخیره کنید.
- درون حلقه while(۱) در تابع main کدی ننویسید.
- ماژول‌ها را به‌صورت وقفه‌ای راه‌اندازی کنید.
- از Delay و روش‌های Busy waiting استفاده نکنید.

[1] https://en.wikipedia.org/wiki/Random_number_generation#%22True%22_vs._pseudo-random_numbers

خلاصه نحوه راه‌اندازی LCD کاراکتری:

۱. اتصال پین‌های LCD به برد (پیش‌فرض پین‌های D8 تا D14)



۲. پروژه را بدون تغییر حالت پیش‌فرض پین‌های بالا در CubeMX ایجاد کنید.

- اگر از وقفه‌ای استفاده می‌کنید که در آن توابع کتابخانه LiquidCrystal فراخوانی می‌شوند نباید این وقفه اولویت صفر داشته باشد و باید اولویت ۱ یا بیشتر را برای آن تنظیم کنید.

۳. افزودن فایل LiquidCrystal.c به پوشه src پروژه

۴. افزودن فایل LiquidCrystal.h به پوشه inc پروژه

۵. در فایل main.c کتابخانه LiquidCrystal را include کنید.

```
#include "LiquidCrystal.h"
```

۶. در ابتدای تابع main و بعد از کد فراخوانی توابع initialization تابع LiquidCrystal را در begin2 و end2 فراخوانی کنید و پین‌هایی که برای LCD در نظر گرفتید را به‌عنوان پارامترهای آن پاس دهید.

```
LiquidCrystal (GPIOD, GPIO_PIN_8, GPIO_PIN_9, GPIO_PIN_10, GPIO_PIN_11, GPIO_PIN_12,
GPIO_PIN_13, GPIO_PIN_14);

begin(20,4);
```

- اگر از توابع کتابخانه LiquidCrystal در تایمر یا تابع ISR واحد دیگری استفاده کرده‌اید، تابع start مربوط به این واحدها را بعد از فراخوانی تابع initialize مربوط به LCD (تابع LiquidCrystal) بیاورید.