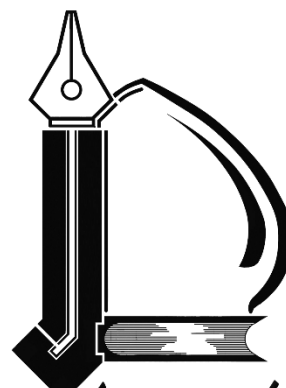




دانشگاه مهندسی



دانشگاه فنی و حرفه‌ای

پروژه نهایی آزمایشگاه ریزپردازنده و زبان اسمبلی

بازی Super Mario Run

بهار ۱۴۰۰



## 1 موضوع

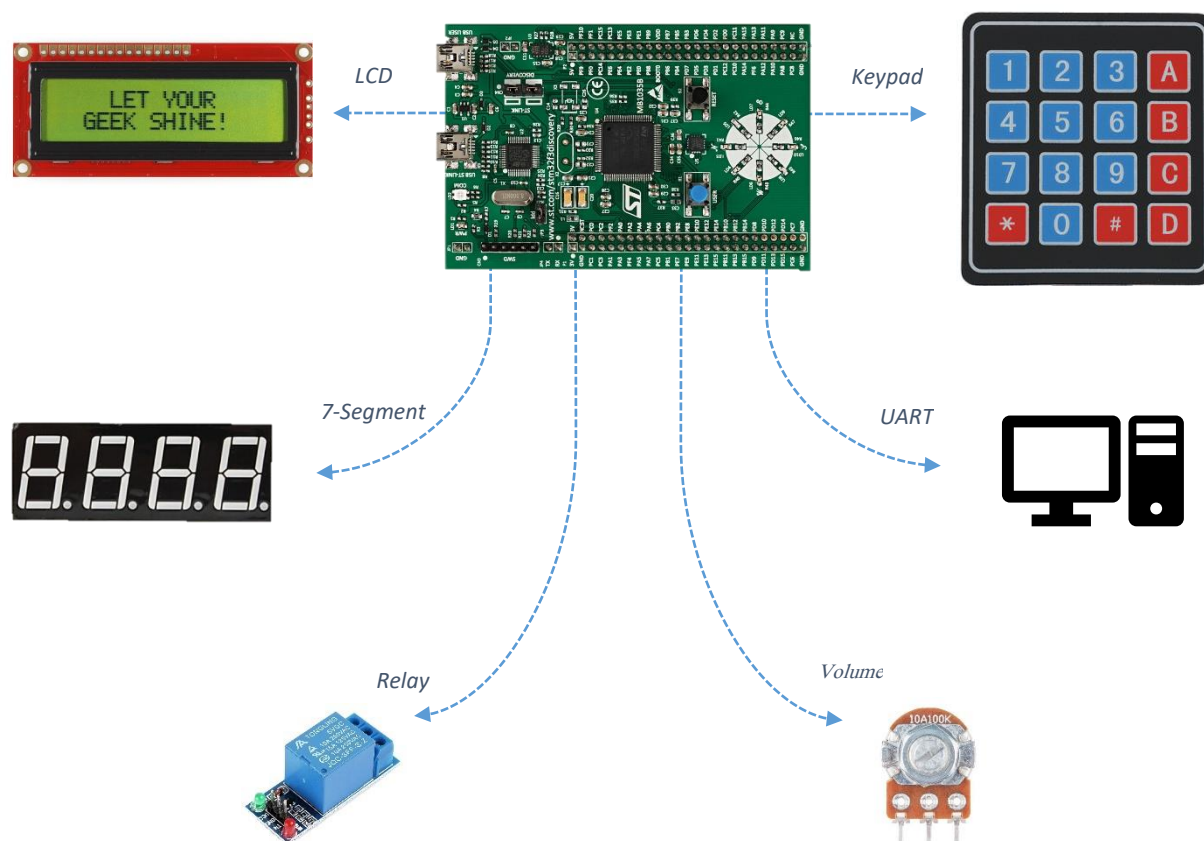
در این پروژه، طراحی و پیاده سازی نسخه ساده ای از بازی **Super Mario Run** مطلوب است.

## 2 زمان بازگردانی

زمان آپلود کد و تحویل مجازی پروژه **۱۵ تیر ساعت ۸ صبح** می باشد.

تحویل مجازی پروژه تنها از روی **کد آپلود شده** می باشد و در صورت مشاهده هرگونه **مغایرت نمره 0** به پروژه تعلق خواهد گرفت.

## 3 نمای کلی اجزا سیستم



در این پروژه طراحی و پیاده‌سازی نسخه ساده‌ای از بازی **Super Mario Run** با استفاده از مباحثی که آموخته‌اید مدنظر است. ابتدا صفحه شروع شامل نام و نمایی از بازی نمایش داده می‌شود و با فشردن دکمه شروع توسط کاربر بازی شروع می‌شود.

نقشه بازی ۶۰ خانه طول دارد و با سرعتی بین ۰.۵ تا ۲ خانه در ثانیه که با Volume کنترل می‌شود، از سمت راست به چپ حرکت می‌کند. بازیکن باید از روی موانع و چاله‌هایی که سر راه او قرار گرفته عبور کند و با جمع‌آوری سکه‌ها به پایان مرحله برسد.

دستورات کنترلی شامل شروع/متوقف کردن بازی و ۲ نوع حرکت روبه‌جلو و پرش، از کیپد و UART دریافت می‌شوند. بعد از توقف حین انجام بازی با فشردن مجدد دکمه، بازی از همان جایی که متوقف شده ادامه می‌یابد. حرکت پرش باعث می‌شود اگر ماریو روی زمین باشد، دو خانه به سمت بالا برود و سپس به پایین بازگردد. صدای پرش با تغییر وضعیت رله ایجاد می‌شود و انیمیشن آن در ۴ حرکت یک خانه‌ای با سرعتی متناسب با سرعت فعلی بازی نمایش داده می‌شود. حرکت روبه‌جلو باعث می‌شود اگر مانعی در خانه جلوی ماریو نباشد او یک خانه به جلو حرکت کند و در حین پرش هم قابل انجام است.

موانع و اجسامی که در مسیر ماریو پدیدار می‌شوند کاملاً تصادفی<sup>۱</sup>، به صورتی که همیشه راهی برای عبور ماریو وجود داشته باشد تولید می‌شوند و در ۴ نوع زیر هستند:

- ❖ آجر: در سطر اول ظاهر می‌شود و حرکت ماریو را محدود می‌کند.
- ❖ صندوق: مانند آجر است اما در صورتی که ماریو از خانه پایین صندوق بپرد و از زیر به آن بخورد صندوق تبدیل به سکه شده و یکی به تعداد سکه‌های بازیکن اضافه می‌شود.
- ❖ لوله بسته: در سطر سوم پدیدار می‌شود و ماریو برای عبور باید از روی آن پرش کند.
- ❖ چاله: در سطر چهارم قرار گرفته و در صورتی که بازیکن با موفقیت از روی آن پرش نکند به درون چاله سقوط کرده و می‌بازد.



شکل ۱ نقشه نمونه

در هر لحظه، سرعت حرکت نقشه بر روی دو رقم سمت چپ و تعداد سکه‌های جمع‌آوری شده بر روی دو رقم سمت راست 7-Segment که با علامت "." از یکدیگر جدا شده‌اند با کمک آیسی مربوطه نشان داده می‌شود.



شکل ۲ نمایش تعداد سکه و سرعت روی 7-Segment

در صورتی که ماریو درون چاله بیفتد یا از حرکت بایستد و به تیغه ستون اول برسد یکی از ۳ جان او که با روشن بودن همان تعداد LED روی برد نشان داده شده، کم می‌شود و دوباره روی زمین ظاهر شده و قادر به ادامه حرکت خود خواهد بود. اگر ماریو به پرچم انتهای مرحله برسد، بازیکن پیروز می‌شود و در صورتی که ۳ جان ماریو تمام شود بازیکن می‌بازد. در هنگام پایان بازی پیام مناسب با امتیاز نهایی بازیکن روی LCD و ترمینال کامپیوتر نمایش داده می‌شود. امتیاز نهایی بازیکن بر اساس زمان مصرف شده و تعداد سکه‌های جمع‌آوری شده محاسبه می‌شود.

## 5 نکات قابل توجه

- تمام قسمت‌های پروژه باید به صورت **وقفه‌ای** پیاده‌سازی شود.
- داخل حلقه **while** در تابع main نباید دستوری نوشته بشود.
- پروژه به صورت انفرادی می‌باشد و در صورت مشاهده و اثبات هرگونه **تقلب** و شباهت در کدها نمره طرفین **100%-** در نظر گرفته خواهد شد. در زمان تحویل شما باید آمادگی انجام تغییرات خواسته شده را در برنامه خود را داشته باشید.
- جزئیاتی که در صورت پروژه تعیین نشده‌اند را خودتان می‌توانید به صورت دلخواه تعریف نمایید.
- فایل‌های پروژه خود را در یک فایل zip یا rar قرار دهید و آن را به شکل زیر نام‌گذاری کنید:

Name\_StudentNumber\_Pr.zip

## 6 توصیه‌ها برای پیاده‌سازی

- باتوجه به متصل بودن ماژول‌های متفاوت به میکروکنترلر و وجود وقفه‌های متعدد، پیشنهاد می‌شود که فرکانس کاری میکروکنترلر را روی 72 مگاهرتز تنظیم نمایید.
- باتوجه به وجود وقفه‌های متعدد در برنامه نهایی و سرعت کاری پایین واحد ADC، توصیه می‌شود تا Sampling Time در واحد ADC را روی 601.5 سیکل قرار دهید تا میکروکنترلر زمان رسیدگی به تمامی وقفه‌ها را داشته باشد.
- برای تغییر کاراکترها در صفحه سعی کنید تا حد ممکن از تابع clear که باعث ایجاد تأخیر نسبتاً زیادی در اجرا می‌شود استفاده نکنید. همچنین دستورات LCD و اندازه آن‌ها را به حداقل برسانید و به صورت بهینه LCD را به روزرسانی کنید.
- استفاده از مقاومت برای LEDها و نقطه دسیمال 7-Segment فراموش نشود.
- ابتدا تمام ماژول‌ها و پین‌های ورودی و خروجی مورد استفاده را راه اندازی و تست کنید تا در میان کار متوجه مشکلات نشوید.

- LCD کاراکتری ای که در اختیار دارید، حداکثر ۸ کاراکتر تعریف شده توسط کاربر را به صورت همزمان می تواند در حافظه ذخیره کند اما در حین اجرای برنامه این کاراکترها می توانند تغییر کنند تا بیشتر از ۸ کاراکتر به صورت کلی پوشش داده شود.
- این LCD علاوه بر توانایی نمایش کاراکترهای ASCII، توانایی پشتیبانی از زبان ژاپنی را نیز دارد که می توانید از کاراکترهای این زبان هم به شکل مستقیم استفاده کنید. (برای توضیحات بیشتر این [لینک](#) را مشاهده کنید)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	CC0			0	@	P	`	p				-	タ	ミ	α	p
1	CC1		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
2	CC2		"	2	B	R	b	r			「	イ	ツ	メ	β	θ
3	CC3		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
4	CC4		\$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
5	CC5		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
6	CC6		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
7	CC7		'	7	G	W	g	w			ア	キ	ヌ	ラ	g	π
8	CC0		(	8	H	X	h	x			イ	ク	ネ	リ	√	x̄
9	CC1		)	9	I	Y	i	y			ウ	ケ	ノ	ル	⁻¹	y
A	CC2		*	:	J	Z	j	z			エ	コ	ハ	レ	j	千
B	CC3		+	;	K	[	k	{			オ	サ	ヒ	ロ	x	万
C	CC4		,	<	L	¥	l				ヤ	シ	フ	ワ	¢	円
D	CC5		-	=	M	]	m	}			ユ	ス	ヘ	ソ	£	÷
E	CC6		.	>	N	^	n	→			ヨ	セ	ホ	°	ñ	
F	CC7		/	?	O	_	o	←			ツ	ソ	マ	°	ö	■

- افزودن افکت‌های گرافیکی LCD برای شروع و پایان بازی، حرکت و پریدن ماریو و ... (+5%)
- رابط کاربری گرافیکی برای نمایش آنلاین بازی بر روی کامپیوتر یا تلفن همراه. (+15%)
- رابط کاربری گرافیکی روی کامپیوتر برای کنترل ماریو، سرعت بازی و ... (+10%)
- اضافه کردن قابلیت کنترل خودکار ماریو، عبور از موانع و جمع‌آوری سکه‌ها. (+10%)
- قابلیت ذخیره‌سازی و بارگذاری بازی در حال انجام به کمک UART توسط کامپیوتر. (+10%)
- نمایش جدول امتیازات در پایان بازی و دریافت اسم بازیکن با پیاده‌سازی کیپد به صورت کاراکتری (مانند کیبورد موبایل‌های قدیمی). با هر بار فشردن یک کلید، کاراکتر بعدی مربوط به آن کلید نمایش داده شود و در نهایت با استفاده از یک کلید تأیید (کلید "B"، ردیف دوم ستون سمت راست کیپد) آن کاراکتر ثبت شود. (+15%)



- موارد دیگر بسته به خلاقیت، کاربرد و پیچیدگی موضوع می‌تواند دارای نمره اضافه باشد.

### با آرزوی موفقیت

تیم بخش پروژه درس ریزپردازنده و زبان اسمبلی