

به نام خدا

زهرا عربی 9523083

گزارش تمرین 2

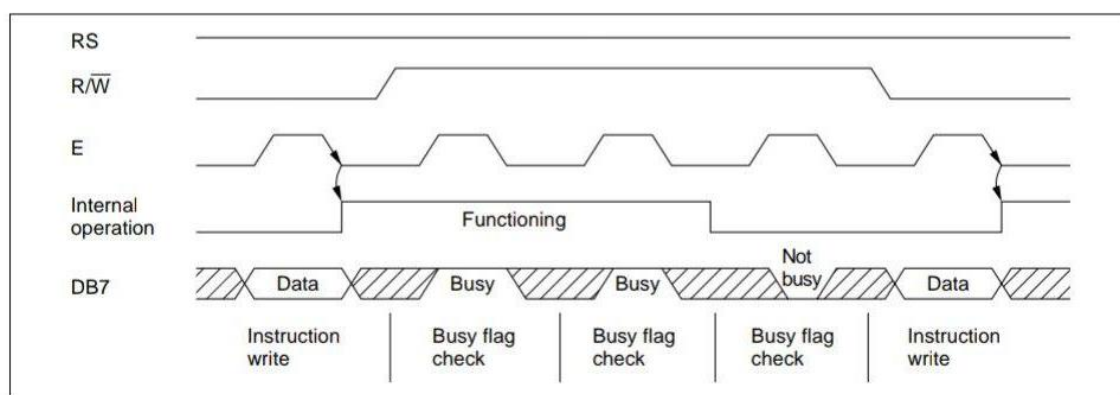
در این تمرین سعی در نوشتن کتابخانه ی کاراکتری داشتیم. اولین موردی که باید درباره ی lcd بدانیم این است که می‌خواهیم 4 بیتی آن را راه اندازی کنیم یا 8 بیتی. اگر 4 بیتی راه اندازی شود از بین بیت های داده، بیت های d4 تا d7 lcd به میکرو وصل می‌شود ولی اگر بخواهیم 8 بیتی آن را راه اندازی کنیم بیت های d0-d7 ال سی دی به میکرو وصل می‌شود. در این کتابخانه با تعریف enum و مد آن را مشخص کردیم. سپس با تعریف یک struct ویژگی های lcd را مشخص می‌کنیم (که این ویژگی ها شامل پورت و پین دیتا و بیت های rs, en, rw می‌باشد). برای اینکه بتوانیم پورت و پین های struct را از کاربر بگیریم تابعی به نام lcd_create تعریف می‌کنیم که ورودی آن پورت و پین های مهم lcd می‌باشد و برای راحتی برای راحتی هرکدام از ویژگی های struct را برابر ورودی قرار می‌دهیم. و انتهای این تابع lcd را init می‌کنیم.

Init کردن lcd به صورت زیر است (طبق دیتاشییت):

ابتدا باید زمان مشخصی را صبر کرد تا مقدار تغذیه ی به مقدار مطلوب (برای مثال 4.7 برسد) سپس باید function set را برای آن تعریف کنیم. برای این کار از یک تابع write_command استفاده می‌کنیم که در ورودی آن برای ست کردن d0-d7 می‌باشد و مقادیر rs, en, rw به صورت جداگانه ست یا ریست می‌شوند.

برای init کردن توجه به این نکته ضروری است که برای چاپ باید مقدار rs صفر شود و مقدار rw نیز باید صفر شود.

برای بیت en طبق جدول زیر باید یکبار یک شود و بعد صفر بشود.



تعریف فانکشن ست مانند تصویر زیر می باشد.

Function set	Code	RS	R \overline{W}	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
		0	0	0	0	1	DL	N	F	*	*

DI : طول داده را تعیین میکند. برای ال سی دی 8 بیتی طول داده باید 8 باشد بنابراین این بیت باید 1 باشد و برای ال سی دی 4 بیتی طول داده باید 4 بیت باشد بنابراین این بیت باید صفر باشد (البته زمانی که 4 بیت انتخاب میشه دیتا باید دوبار فرستاده و دریافت بشه).

N : این بیت تعداد خط ها را مشخص میکند که برای هر دو ال سی دی مقدار آن را 2 در نظر میگیریم.

F : فونت کاراکتر را مشخص میکند که طبق جدول زیر N و F تعیین میگردد. که ما pixel 5*8 را انتخاب میکنیم.

N	F	No. of Display Lines	Character Font	Duty Factor	Remarks
0	0	1	5 × 8 dots	1/8	
0	1	1	5 × 10 dots	1/11	
1	*	2	5 × 8 dots	1/16	Cannot display two lines for 5 × 10 dot character font

Note: * Indicates don't care.

بنابراین فانکشن ست 8 بیتی برابر می شود با $(00111000)_2 = 0x38$

و فانکشن ست 4 بیتی برابر می شود با $(00101000)_2 = 0x28$

در ابتدای شروع به کار ال سی دی باید مشخص کنیم که cursor آن کجا باشد برای این کار از یک تابع استفاده میکنیم که سطر و ستون را به عنوان ورودی به آن می دهیم. طبق جدول زیر مختصات خانه های ال سی دی 16*2 بر اساس هگز مشخص شده اند.

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

سطر اول: $(0000000)_2 = 0x00$

سطر دوم: $(1000000)_2 = 0x40$

برای اینکه جای کرسر را مشخص کنیم باید رجیستر آدرس DDRAM را ست کنیم طبق جدول زیر در دیتاشیت:

Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 μ s
-------------------------	---	---	---	-----	-----	-----	-----	-----	-----	-----	---	------------

پس ستون ورودی باید به اضافه ی مختصات نقطه ی اول بشود ولی سطر ورودی تعریف شده است که مقدار آن یا صفر است یا یک.

برای سطر اول: $(10000000)_2 = 0x80$

برای سطر دوم: $(11000000)_2 = 0xc0$

برای پاک کردن صفحه نیز طبق جدول زیر در دیتاشیت عمل میکنیم:

Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms

برای نوشتن روی lcd تابعی تعریف میکنیم که ورودی آن یک string است و تا زمانی که ورودی آن null نشده کاراکتر به کاراکتر آن را چاپ میکنیم. توجه به این نکته ضروری است که برای چاپ باید مقدار rs یک شود و مقدار rw باید صفر شود.

سوالات:

1. اگر اشاره گر تعریف نمیشد باید پین های مشخصی را در نظر می گرفتیم و در فایل h. ان های را define میکردیم و کاربر اگر میخواست شماره ی پینی را عوض کند باید در کتابخانه تغییرات ایجاد میکرد و مشکل دیگر سائز پین دیتا بود که اگر struct مشمول آن را اشاره گر تعریف نمی کردیم باید سائز ان را مشخص میکردیم و چون در دو مد 4 بیت و 8 بیت میخواستیم راه اندازی کنیم نیاز به دو تابع داشت. بنابراین ان را اشاره گر تعریف میکنیم که طبق مثال زده شده کاربر پین دیتا را به تعداد دلخواه (4 یا 8) و با پین های دلخواه در main.c تعریف کند. در واقع با این کار برنامه کاربر پسند تر شده است.

2. دیتا پورت: برای فرستادن اطلاعات روی ال سی دی استفاده میشود. برای راه اندازی 8 بیتی نیاز به 8 پورت و برای راه اندازی 4 بیتی نیاز به 4 پورت داریم

پایه ی فعال ساز یا en : برای فعال سازی ال سی دی است هرگاه بخواهیم دیتایی را به ال سی دی بفرستیم این پایه باید یکبار یک و سپس صفر شود در این صورت دیتایی دریافت نمیشود.

پایه ی rs : این پایه برای انتخابی متن یا دستور استاگر بخواهیم دستور بفرستیم اگر بخواهیم دستور بفرستیم باید صفر و اگر بخواهیم متن بفرستیم باید یک شود.

پایه rw : در این پایه انتخاب میکنیم که میخواهیم روی ال سی دی بنویسیم یا از روی آن بخوانیم. البته امروزه خواندن از روی ال سی دی رایج نیستو بیشتر روی ال سی دی مینویسیم که در این حالت مقدار این پایه باید 0 باشد.

پایه ی gnd و vcc پایینترین و بالاترین ولتاژ مدار را خواهند داشت.

پایه ی vee برای تنظیم کنتراست به کار میرود.

پایه ی led+ و led- برای تنظیم نور پس زمینه به کار میرود.

3. محدوده ی تغذیه ی ولتاژ برای توان پایین 2.2 تا 5.5 ولت است (طبق دیتاشیت)

4. فرض میکنیم که ولتاژ lcd 5 ولت باشد ولی میکرو کنترلر با ولتاژ 2 ولت باید کار کند برای این کار باید از کاهش دهنده ی سطح ولتاژ dc برای تبدیل 5 ولت به 2 ولت استفاده کنیم که به صرفه تر است که از مدار level shifter استفاده کنیم.

