

8 جلسه هشتم

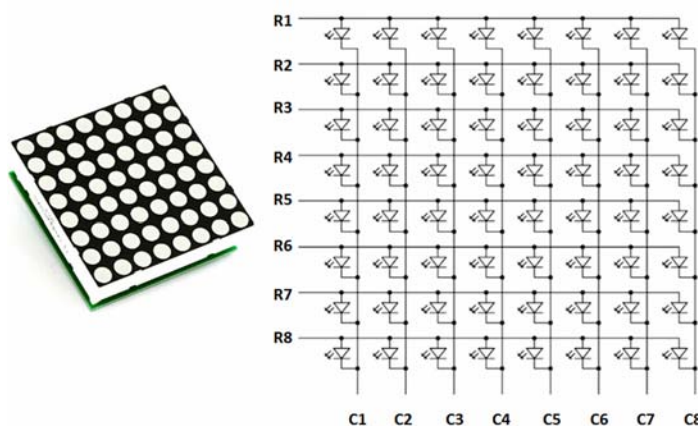
آشنایی با نمایشگرهای Dot-matrix و LCD های گرافیکی

8.1 هدف

در این جلسه به بررسی نمایشگرهای Dot-Matrix و LCD های گرافیکی می‌پردازیم و ویژگی‌ها و نحوه کار هریک از آن‌ها را تشریح خواهیم نمود.

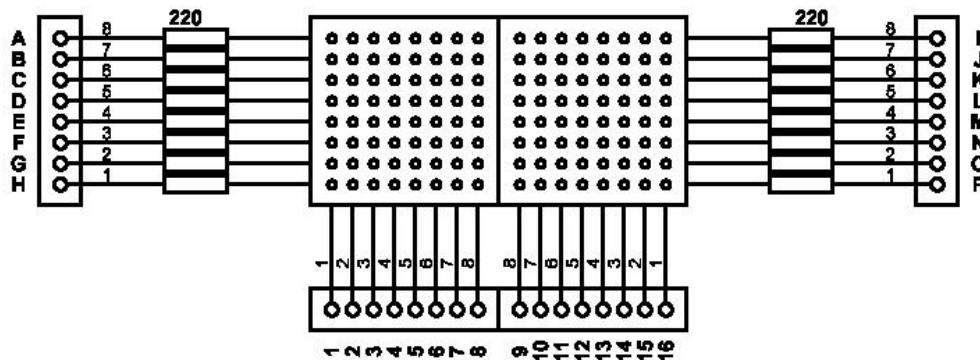
8.2 نمایشگر Dot-Matrix

این نمایشگرها مانند شکل 1-8 از تعدادی LED تشکیل شده‌اند که در هر ردیف دارای آندهای مشترک و در هر ستون با کاتدهای مشترک می‌باشد. برای روشن شدن هر یک از LED ها بایستی سطر و ستون مورد نظر به نحو مناسب فعال گردد.



شکل 1-8: نمایی از ساختار Dot-Matrix

در مجموعه آموزشی آزمایشگاه، دو عدد نمایشگر Dot-Matrix از نوع 8x8 (8 ستون و 8 ردیف) به رنگ سبز در بلوکی تحت عنوان Dot-Matrix Display قرار داده شده است. ستون‌ها توسط اعداد 1 تا 16 و ردیف‌ها توسط حروف A تا P نام‌گذاری شده‌اند. لذا برای روشن نمودن هر کدام از LED ها می‌توان به ستون مربوط به همان LED صفر منطقی و به ردیف متناظر یک منطقی اعمال نمود. شماتیک مربوط به این بلوک در شکل 2-8 مشاهده می‌شود.

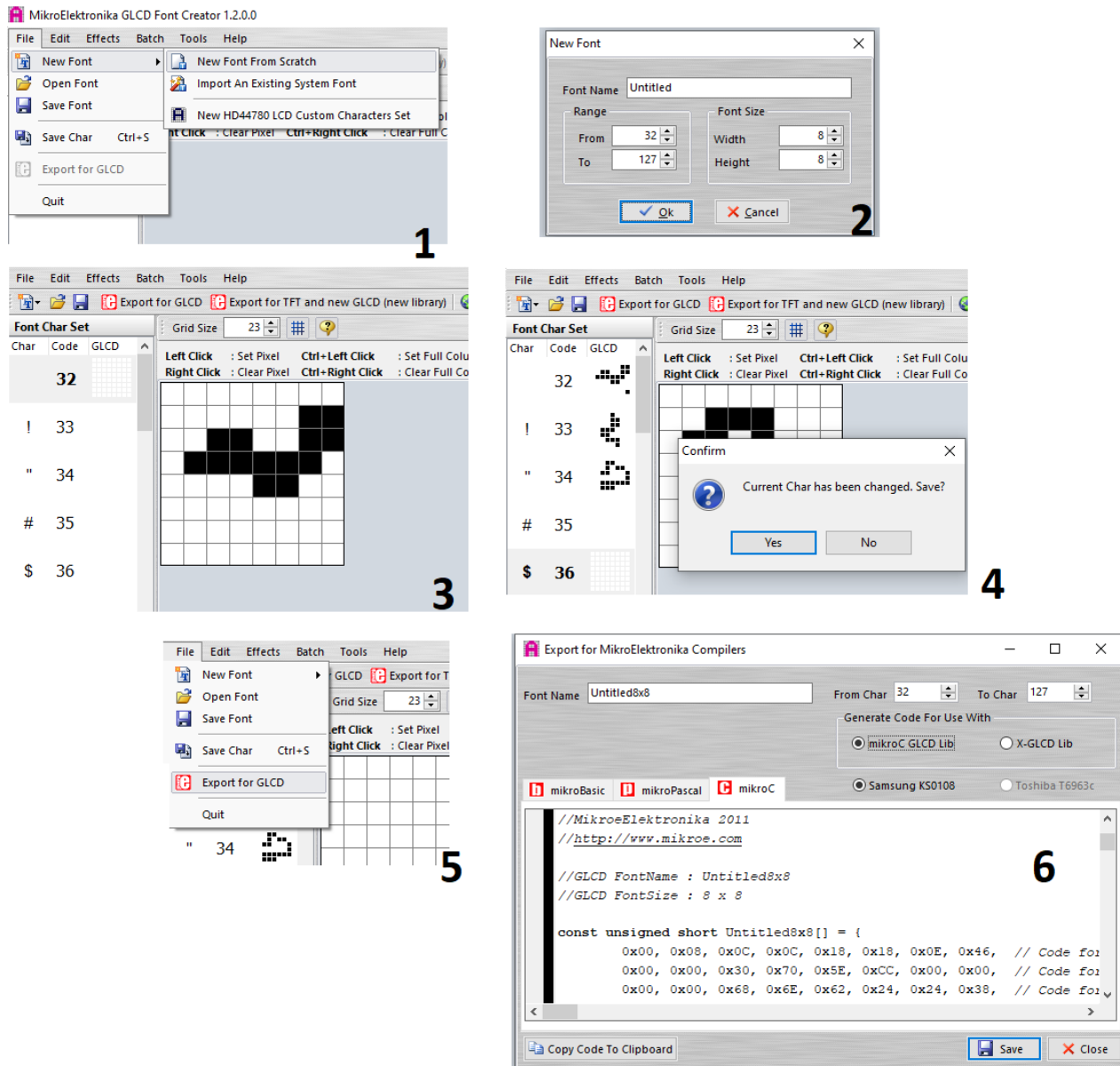


شکل 2-8 - ساختار نمایشگر DotMatrix تعبیه شده بر روی پکیج آزمایشگاه

برای نمایش حروف و علائم روی 64 عدد LED، از روش جاروب کردن سطر و ستون استفاده می‌گردد، به این صورت که ابتدا سطر اول برابر با یک و مابقی سطرها صفر می‌گردد. سپس مقادیر ستون‌ها با توجه به وضعیت خاموش یا روشن بودن هر LED در این سطر تنظیم می‌شود. بعد از تاخیر زمانی مشخص، سطر دوم برابر با یک و مابقی سطرها صفر شده و مقادیر مربوط به ردیف دوم بر روی 8 پایه متناظر با ستون‌ها قرار می‌گیرد. این روند تا سطر هشتم ادامه می‌یابد. (عمل جاروب کردن را می‌توان با شیفت دادن صفر منطقی روی ستون‌ها و قرار دادن مقادیر روی ردیف‌ها نیز انجام داد). برای این که طرح مورد نظر بر روی Dot-Matrix روشن بماند، پروسه ذکر شده با تاخیرهای در حد چند میلی ثانیه به دفعات اجرا می‌گردد و با توجه به خطای دید در چشم انسان، طرح مورد نظر بر روی کل 64 عدد LED در هر لحظه مشاهده می‌گردد. مشابه این روش در LCDهای کاراکتری و برای ایجاد کاراکترهای خاص (مانند علائم و حروف فارسی) نیز استفاده می‌گردد.

8.2.1 ایجاد قلم برای نمایش متون روی Dot-Matrix

برای ایجاد قلم جهت نمایش نشانه‌های نوشتاری مختلف می‌توان از نرم‌افزارهای جانبی مانند GLCD Font Creator استفاده نمود. مراحل مختلف کار با این نرم‌افزار در شکل 3-8 نشان داده شده است.



1: قلم جدید ایجاد می گردد.

2: می توان تعداد کاراکترها را محدود کرد.

3: می توان قلم مورد نظر را طراحی نمود.

4: با کلیک روی **رایه بعدی (مانند کد 33)** پنجره ذخیره نشان داده می شود.

5: کدهای متناظر با قلمها ایجاد می گردد.

6: تمام کدها را انتخاب و با copy کردن می توان در یک فایل txt ذخیره نمود و یا از save استفاده کرد. (در برخی از موارد به سبب اشکالات برنامه، گزینه Save فایلی را ذخیره نمی کند).

شکل 3-8. مراحل کار با GLCD Font Creator

8.2.2 اصول کلی نمایش متن بر روی DotMatrix

داده‌های خروجی حاصل از نرم‌افزار ایجاد قلم برای هر نشانه را به عنوان مقادیر مربوط به ستون‌های Dot-Matrix مطابق برنامه 1-8 و مقادیر مربوط به روند جاروب کردن سطرها را مطابق برنامه 2-8، در ساختارهای آرایه‌ای تعریف و ذخیره می‌نماییم.

```
برنامه 1-8 char alphabet[][8]={
{ 0x06, 0x29, 0x29, 0x29, 0x1F, 0x00, 0x00, 0x00},// a Locate in array = 0
{ 0xFF, 0x09, 0x11, 0x11, 0x0E, 0x00, 0x00, 0x00},// b Locate in array = 1
{ 0x1E, 0x21, 0x21, 0x21, 0x12, 0x00, 0x00, 0x00},// c Locate in array = 2
{ 0x7F, 0x88, 0x88, 0x88, 0x7F, 0x00, 0x00, 0x00},// A Locate in array = 3
{ 0xFF, 0x91, 0x91, 0x91, 0x6E, 0x00, 0x00, 0x00},// B Locate in array = 4
{ 0x7E, 0x81, 0x81, 0x81, 0x42, 0x00, 0x00, 0x00},// C Locate in array = 5
};
```

```
برنامه 2-8 unsigned char R_data[8] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
```

در ادامه یک ساختار switch/case مطابق

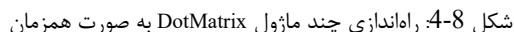
برنامه 3-8 پیاده‌سازی می‌گردد که در آن به ازای هر نشانه، یک case مجزا در نظر گرفته می‌شود. با اجرای این ساختار، موقعیت داده متناظر با حرف مورد نظر در متغیر locat ذخیره می‌شود. به عنوان مثال با در نظر گرفتن برنامه 1-8، برای حرف اول (a) مقدار صفر را برمی‌گرداند، برای حرف دوم (b) مقدار 1 و همین طور برای مقادیر دیگر موقعیت را برمی‌گرداند.

```
برنامه 3-8 void lookup(char input)
{
switch (input)
{
case 'a' :
locat=0;
break;
case 'b' :
locat=1;
break;
case 'c' :
locat=2;
break;
case 'A' :
locat=3;
```

```
break;
case 'B' :
locat=4;
break;
case 'C' :
locat=5;
break;
    }
}
```

8.2.3 استفاده از چند ماژول DotMatrix

با توجه به این که هر ماژول به دو درگاه جهت سامان‌دهی ستون‌ها و ردیف‌ها نیاز دارد، در صورت استفاده بیش از یک ماژول، می‌توان سطرهای همه ماژول‌ها را به یکی از درگاه‌ها متصل کرد و برای ردیف‌ها از تعدادی D-Flip Flop استفاده نمود. نمایی از این سخت‌افزار در شکل 8-4 نشان داده شده است. در این سخت‌افزار از درگاه A برای سطرها و از درگاه B به صورت مشترک برای ستون‌ها استفاده شده است. در این سخت‌افزار با فعال نمودن پایه OE، داده‌های ارسالی از سوی ریزپردازنده در حافظه خارجی ثبت می‌شود و با غیرفعال شدن این پایه، این مقادیر ثابت باقی خواهند ماند. بدین ترتیب در بازه‌های زمانی مناسب می‌توان داده‌های هر ستون ماژول DotMatrix را به صورت مجزا مقدار دهی کرد. برای مدیریت سیگنال‌های OE، می‌توان از یک انکدر استفاده نمود تا بتوان با استفاده از تعداد پایه‌های کمتری از ریزپردازنده، سیگنال‌های OE بیشتری را کنترل نمود.



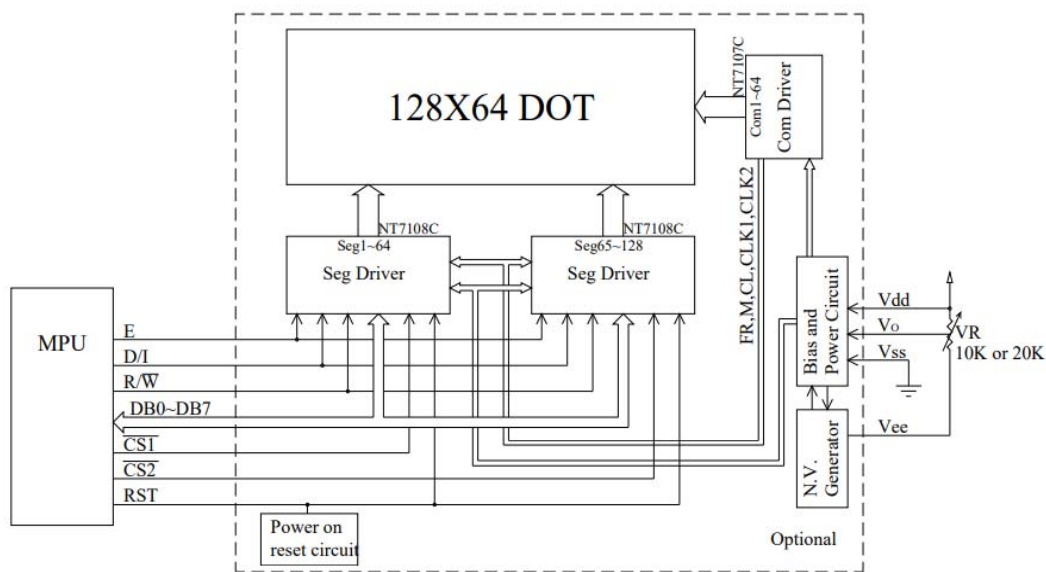
اساس کار GLCD^۱ مشابه DotMatrix است ولی واحدهای نمایشگر نورانی در آن بسیار کوچکتر می باشد. بنابراین برای نمایش داده ها روی آن نیز از روش جاروب کردن سطر استفاده می شود، بدین صورت که در هر مرحله تنها یکی از سطرها یک و مابقی صفر می شوند و در این لحظه، داده های مربوط به آن سطر نوشته می شوند. با توجه به تراکم بالای نقاط نورانی در GLCD، برای تعیین زمان بندی روشن و خاموش شدن هر سطر و ستون نیاز به یک راه انداز^۲ می باشد.

2 Drive

برای راهاندازی ستون‌های GLCD، از تراشه‌ی NT7108C استفاده می‌شود. این تراشه، شامل یک RAM، یک لچ 64 بیتی و یک دیکدر می‌باشد. حافظه RAM برای ذخیره داده‌هایی که از طریق ریزپردازنده منتقل می‌شود، به کار می‌رود.

تراشه NT7107C، نیز یک راهانداز 64 کاناله است که در فاصله‌های زمانی مشخص، هر یک از سطرهاى GLCD را فعال می‌نماید تا داده‌های ذخیره شده درون NT7108C بر روی GLCD نمایش داده شود. در حقیقت، این تراشه عملیات جاروب کردن سطرها را انجام می‌دهد.

GLCD مورد استفاده در برد آزمایشگاه دارای ابعاد 128×64 پیکسل می‌باشد و به همین دلیل مطابق شکل 5-8 از دو تراشه NT7108C و یک تراشه NT7107C برای راهاندازی نمودن 128 ستون و 64 سطر آن استفاده می‌شود. این تراشه‌ها با تراشه‌های KS0107B و KS0108B از شرکت سامسونگ سازگار هستند و می‌توان در تنظیمات CodeVision آن‌ها را انتخاب نمود.



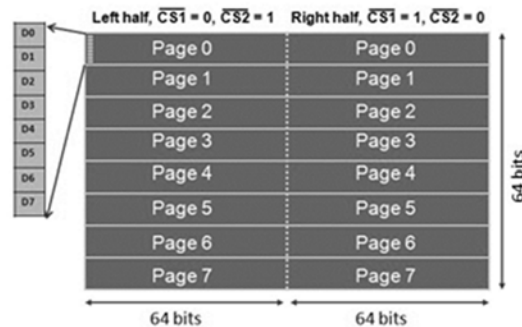
شکل 5-8: مدار راهانداز GLCD در پکیج آزمایشگاهی

تراشه‌ی NT7107C، 64 سطرنمایشگر را (COM1 – COM64) را راهاندازی می‌کند. در هر لحظه از زمان، تنها یکی از خطوط COM فعال می‌باشد. اولین تراشه NT7108C (IC1) سگمنت‌های نیمه سمت چپ را فعال می‌کند و دومین تراشه NT7108C (IC2)، سگمنت‌های نیمه سمت راست را فعال می‌نماید. با استفاده از پین‌های CS11 و CS22 می‌توان به طور جداگانه به هر یک از نیمه‌های نمایشگر دست یافت. هر کدام از نیمه‌ها مانند شکل 6-8 شامل 8 صفحه افقی با شماره‌های 0-7 می‌باشد که هر کدام از 64 بایت تشکیل شده است.

¹ Chip Select 1

² Chip Select 2

مقداردهی از صفحه صفر شروع می‌شود. اگر یک بایت داده به GLCD انتقال داده شود، در اولین ستون صفحه صفر نمایش داده می‌شود (شکل 6-8). اگر این کار برای 64 بایت داده تکرار شود و سپس در نیمه دوم نیز برای 64 داده دیگر ادامه یابد، 8x128 پیکسل از GLCD شامل صفحه صفر کشیده خواهد شد. برای صفحه یک نیز همین روال تکرار می‌گردد.



شکل 6-8: pages

8.2.5 آشنایی با پایه‌های GLCD

پایه‌های GLCD مانند جدول 1-8 به دو دسته تقسیم می‌شوند. دسته نخست پایه‌های داده (DB0-DB7) که وظیفه انتقال داده را برعهده دارند و دسته‌ی دیگر پایه‌های کنترلی هستند. پایه‌های کنترلی به شرح زیر می‌باشند:

R/W: زمانی که این پایه مقدار یک داشته باشد، داده‌های موجود در DB[7:0] را نمایش می‌دهد و زمانی که $E=1$ ، $CS1B=0$ و $CS2B=0$ باشد، مقدار آن توسط پردازنده می‌تواند خوانده شود.

$CS1$ و $CS2$: زمانی که $CS1=1$ و $CS2=0$ باشد، تراشه اول NT7108C (IC1) انتخاب می‌شود و زمانی که $CS1=0$ و $CS2=1$ باشد تراشه دوم NT7108C (IC2) انتخاب می‌شود.

RS: اگر $RS=0$ باشد، GLCD دستورالعمل‌های دریافتی را اجرا می‌کند، در غیر این صورت مقدار خوانده شده از پایه‌های داده را در Data Register قرار می‌دهد.

جدول 1-8: مشخصات پایه‌های GLCD

	Name	Level	Function
1	CS1	L	Chip Select 1 (segment 1-64)
2	CS2	L	Chip Select 2 (segment 65-128)
3	Vss	0V	Ground
4	Vcc	5.0V	5v Supply in+
5	V0	variable	Contrast Adjust
6	RS or D/I or CD	H/L	H: Data , L: Instruction

7	R/W	H/L	H: Read(MPU← Module) , L :Write(MPU→ Module)
8	E	H	ENABLE SIGNAL
9	DB0	H/L	Data bus line
10	DB1	H/L	Data bus line
11	DB2	H/L	Data bus line
12	DB3	H/L	Data bus line
13	DB4	H/L	Data bus line
14	DB5	H/L	Data bus line
15	DB6	H/L	Data bus line
16	DB7	H/L	Data bus line
17	RST	L	Reset the LCM
18	VEE		Negative Voltage Output
19	A		Power supply for B/L(+)LED BACKLIGHT
20	K		Power supply for B/L(-)LED BACKLIGHT

8.3 برنامه نویسی GLCD در محیط کد ویژن

تنظیمات LCD گرافیکی از طریق محیط CodeWizard مانند شکل 7-8 انجام می‌شود. در تنظیمات GLCD

لازم است که تراشه کنترل کننده‌ی آن مشخص شود، لذا تراشه‌ی KS0108B که سازگار با NT7108C است، انتخاب می‌شود.

Graphic Display Settings

Display Type:
 KS0108 128x64 CS1,CS2

☐ Use Image Storage in External Memory

1 KS0108 128x64 CS1,CS2
 KS0108 128x64 /CS1,/CS2

2 **3**

Data Control

Display	AVR
DB0	PORTA Bit: 0
DB1	PORTA Bit: 1
DB2	PORTA Bit: 2
DB3	PORTA Bit: 3
DB4	PORTA Bit: 4
DB5	PORTA Bit: 5
DB6	PORTA Bit: 6
DB7	PORTA Bit: 7

1: انتخاب کنترل کننده مورد نظر (یکی از دو مورد نشان داده شده انتخاب می‌شود).

2: انتخاب سیگنال‌های داده متصل شده به GLCD

3: انتخاب سیگنال‌های کنترلی متصل شده به GLCD

شکل 7-8: تنظیمات کد ویژن برای GLCD

بعد از تنظیمات انجام شده، کدها ذخیره می‌گردند. البته بعد از ذخیره کدها، در صورت نیاز می‌توان از طریق مسیر زیر پایه‌های متصل به GLCD را تغییر داد.

Project>configure>C compiler>Libraries>Graphic display

8.4 معرفی فایل سرآیند glcd.h

در کدهای ایجاد شده، فایل سرآیند جدیدی به نام glcd.h فراخوانی می‌شود که حاوی توابع لازم برای کار کردن با GLCD می‌باشد. این توابع شامل تنظیمات اولیه، پاک کردن GLCD، نوشتن متن، انتقال مکان نما و یا رسم شکل‌های هندسی و غیره می‌باشد. برای اطلاع از آخرین تغییرات دستورالعمل‌ها بایستی به راهنمای Codevision مراجعه گردد.

8.5 نمونه برنامه GLCD

برای آشنایی بیشتر با GLCD تنظیمات اولیه در محیط CodeWizard انجام شده و برخی دستورات به برنامه اضافه می‌گردد تا متن مورد نظر روی GLCD نشان داده شود. همچنین چند خط به عنوان یک طرح گرافیکی ساده نیز روی GLCD رسم می‌گردد. قطعه کد شکل 8-8 چگونگی نمایش یک متن و شکل را بر روی GLCD نشان می‌دهد که البته برای کاهش حجم برنامه، توضیحات هر دستور حذف شده است.

```

1  #include <mega16.h>
2  #include <glcd.h>
3  #include <font5x7.h>
4  #include <delay.h>
5
6  void main(void)
7  {
8      GLCDINIT_t glcd_init_data;           //note 1
9      glcd_init_data.font=font5x7;         //note 2
10     glcd_init_data.readxmem=NULL;         //note 3
11     glcd_init_data.writexmem=NULL;        //note 4
12     glcd_init(&glcd_init_data);           //note 5
13
14     glcd_outtextf("MICROPROCESSOR LAB");   //note 6
15     delay_ms(800);
16     glcd_clear();                          //note 7
17     glcd_outtextf("Some line styles:");
18
19     glcd_setlinestyle(1, GLCD_LINE_DOT_SMALL); //note 8
20     glcd_line(0,10,127,10);                //note 9
21     glcd_setlinestyle(1, GLCD_LINE_DOT_LARGE); //note 10
22     glcd_line(0,20,127,20);
23     while(1);
24 }
```

شکل 8-8: نمونه کد برای GLCD

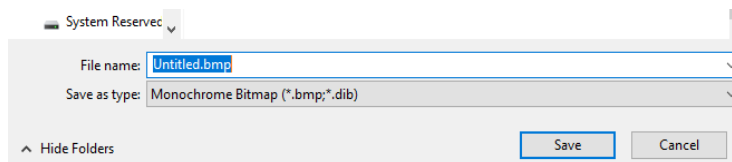
- نکات:
- 1: تعریف ساختار
 - 2: تعیین نوع قلم در ساختار
 - 3: تعیین قابلیت خواندن از حافظه خارجی
 - 4: تعیین قابلیت نوشتن در حافظه خارجی
 - 5: پیکربندی GLCD با تنظیمات مشخص شده در ساختار
 - 6: نمایش متن روی GLCD
 - 7: پاک کردن GLCD
 - 8: تنظیم استایل خط قبل از رسم شدن
 - 9: رسم خط

8.6 قلم‌های نمایش کاراکتر

برای نمایش حروف بر روی GLCD معمولاً از دو قلم متفاوت با ابعاد 5x8 و 8x8 استفاده می‌شود. قلم‌های فارسی معمولاً از نوع 8x8 و قلم‌های انگلیسی از نوع 5x8 می‌باشند. فایل سرآیند font5x7.h برای قلم انگلیسی قابل استفاده است. برای ایجاد قلم‌های جدید هم می‌توان از نرم‌افزارهای طراحی قلم مشابه قسمت DotMatrix استفاده نمود.

8.7 نمایش تصویر روی GLCD

با توجه به این که GLCD تعبیه شده در پکیج آزمایشگاه، سیاه و سفید است، صرفاً تصاویر محدودی را می‌توان بر روی آن نشان داد. این تصاویر باید دارای قالب monochrome باشند که می‌توان عکس مورد نظر را در محیط paint و مطابق شکل 8-9 در این قالب ذخیره نمود. همچنین برای جلوگیری از افت کیفیت بهتر است اندازه عکس 128*64 پیکسل باشد.

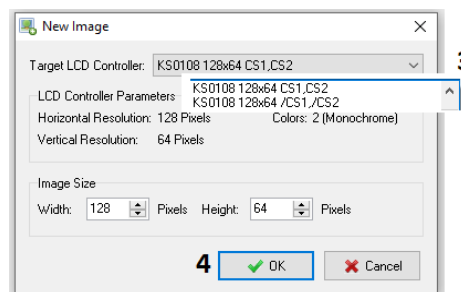


شکل 8-9: ذخیره تصویر در محیط Paint

8.7.1 تهیه کدهای عکس با استفاده از نرم‌افزار LCD Vision

برای ذخیره تصاویر در حافظه پردازنده و به منظور نمایش بر روی GLCD باید آن‌ها را به صورت آرایه دو بعدی از داده‌های باینری در آورد. البته جهت فشردگی از کد مبنای 16 به جای باینری استفاده می‌شود. به این منظور پس از آماده‌سازی اولیه عکس مورد نظر، می‌توان در محیط LCD Vision کدهای مبنای 16 تصاویر را تهیه نمود که مراحل آن در شکل 8-10 قابل مشاهده است.

- 1- نرم افزار LCD Vision را باز نمایید.
- 2- از منوی file گزینه new Image را بزنید.
- 3- در پنجره باز شده کنترلر مربوطه را انتخاب نمایید.
- 4- گزینه ok را بزنید.
- 5- در پنجره باز شده تصویر را با استفاده از موس رسم نمایید یا از منوی file گزینه import file تصویر مورد نظر را انتخاب نمایید.
- 6- در پنجره باز شده تصویر مورد نظر را انتخاب و open نمایید.
- 7- در پنجره باز شده گزینه ok را انتخاب نمایید.
- 8- بخشی از تصویر را که قصد دارید روی GLCD نمایش دهید را با موس انتخاب نمایید.
- 9- از منوی file گزینه Export را انتخاب نمایید.
- 10- در پنجره باز شده گزینه ok را انتخاب نمایید.
- 11- کدهای تولید شده در پنجره ی Export preview نشان داده می‌شود.
- 12- از منوی file گزینه Export save را انتخاب نمایید.
- 13- در پنجره باز شده گزینه ok را انتخاب نمایید.
- 14- مسیر مورد نظر را انتخاب و ذخیره نمایید.
- 15- در پنجره باز شده اعلام می‌نماید که د نمونه فایل با فرمت h و c ایجاد شده است.



شکل 8-10: مراحل تهیه کدهای مبنای 16 تصویر در محیط LCD Vision

8.7.2 برنامه نمایش عکس در محیط Codevision

بهتر است فایل‌های تهیه شده توسط LCD Vision در مسیر پروژه قرار گیرند و سپس در محیط Code vision با استفاده از منوی زیر به فایل‌های پروژه اضافه گردند.

Project > Configure > files > input files > add

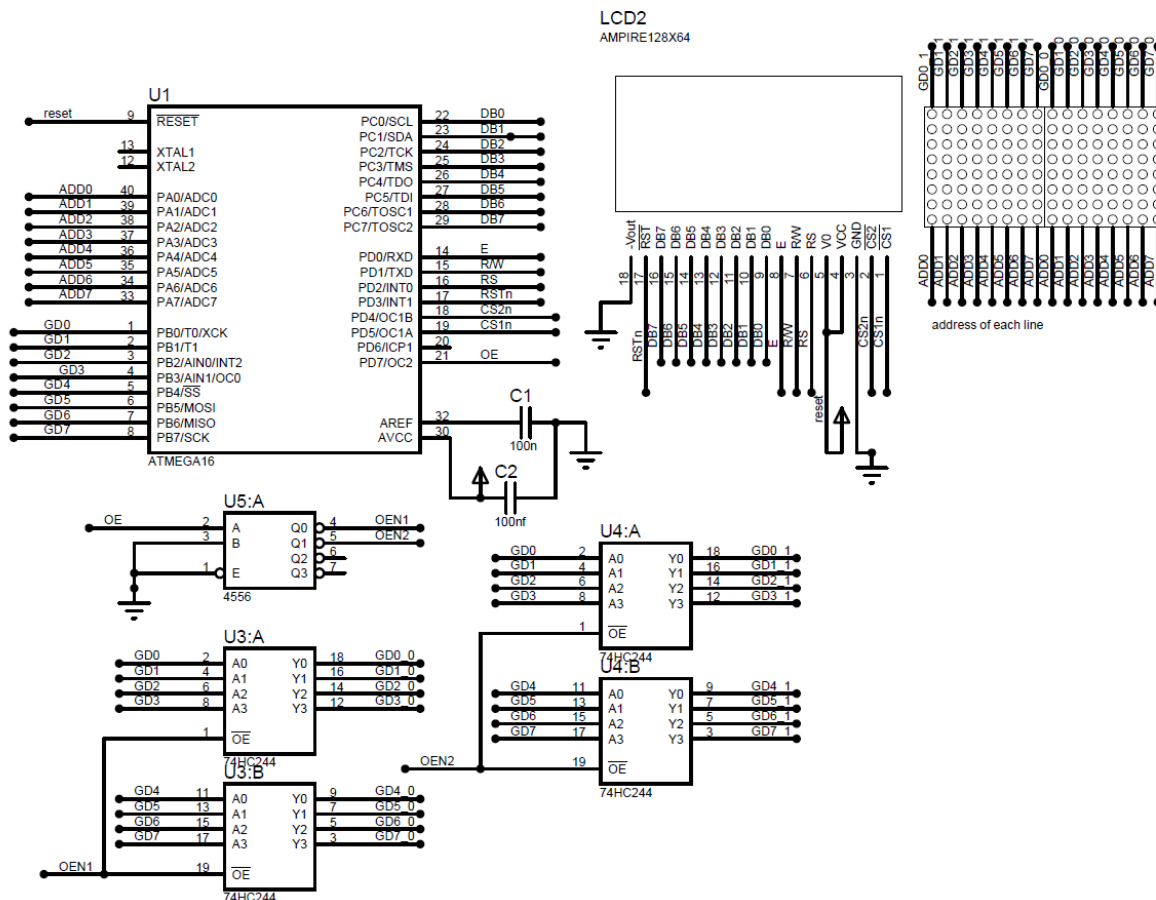
همچنین فایل سرآیند ایجاد شده نیز باید در ابتدای برنامه فراخوانی گردد ("picture_name.h" include #). لازم به ذکر است که نام آرایه‌ی شامل کدهای تصویر نبایستی با عدد شروع شود. در غیر این صورت خطای کامپایلر را به دنبال خواهد داشت. برای رسم تصویر بر روی GLCD از یکی از توابع موجود در graphic.h به نام glcd_putimage به صورت زیر استفاده می‌گردد. چنانچه کدها در حافظه flash ذخیره شده باشد، از تابع جایگزین glcd_putimagef استفاده می‌گردد.

```
gled_putimage(0, 0, picture_name, GLCD_PUTCOPY);
```

```
gled_putimagef(0, 0, picture_name, GLCD_PUTCOPY);
```

8.8 برنامه‌های اجرایی مبحث GLCD

سیستم طراحی شده شکل 8-11 را در نظر بگیرید که حاوی دو مجموعه DotMatrix می‌باشد و هر کدام برای فعال شدن به دو درگاه (جمعا 32 پایه) نیاز دارد. اما از آن جا که در این سخت‌افزار از GLCD نیز استفاده گردید بایستی تعداد درگاه‌های مورد نیاز برای راه اندازی DotMatrix بهینه گردد. لذا از یک درگاه به طور مشترک برای آدرس‌دهی هر ردیف استفاده شده است و با استفاده از بافرهای 74HC244 می‌توان داده‌های هر DotMatrix را ذخیره نمود و در فاصله زمانی مشخصی آن را به روز نمود. برای تهیه سیگنال OEn بافر نیز می‌توان از یک دیگر استفاده کرد تا بتوان چندین DotMatrix را درایو نمود. بدین ترتیب فقط 17 پایه از ریزپردازنده برای DotMatrix اشغال می‌شود.



شکل 8-11: نمایی از سخت‌افزار مبحث GLCD و DotMatrix

1. زیربرنامه‌ای بنویسید که حرف اول نام و نام خانوادگی شما را به صورت متن روان روی DotMatrix نشان دهد.
2. زیربرنامه‌ای بنویسید که یک تصویر دلخواه را روی LCD گرافیکی نمایش دهد.
3. یک ساعت آنالوگ دارای عقربه‌های ساعت شمار، دقیقه شمار و ثانیه شمار طراحی و روی LCD گرافیکی نمایش دهید. (راهنمایی: برای رسم ساعت آنالوگ می‌توان با استفاده از روابط مثلثاتی، مختصات هریک از عقربه‌ها را محاسبه کرد و با تغییر هر کدام، خط جدیدی به مبدا ساعت و نقطه مورد نظر رسم نمود).
4. بندهای فوق را در قالب یک پروژه در بیاورید به گونه ای که ابتدا تصویر به مدت سه ثانیه نمایش داده شود، سپس متن روان روی DotMatrix و نهایتاً ساعت روی LCD گرافیکی نمایش داده شود.