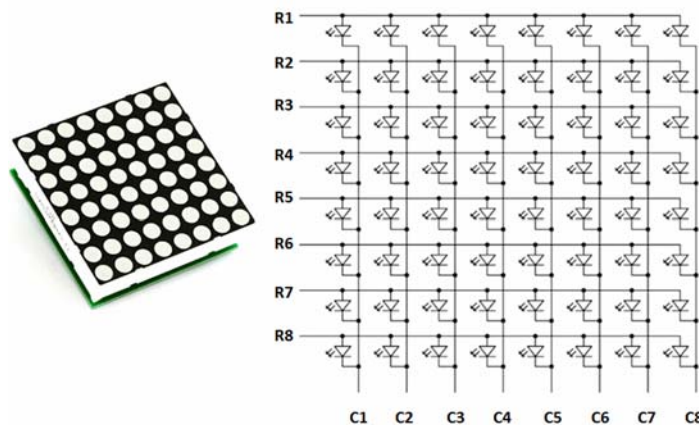


9 آشنایی با نمایشگرهای Dot-matrix و LCDهای گرافیکی

این بخش به بررسی نمایشگرهای Dot-matrix و LCDهای گرافیکی می‌پردازد.

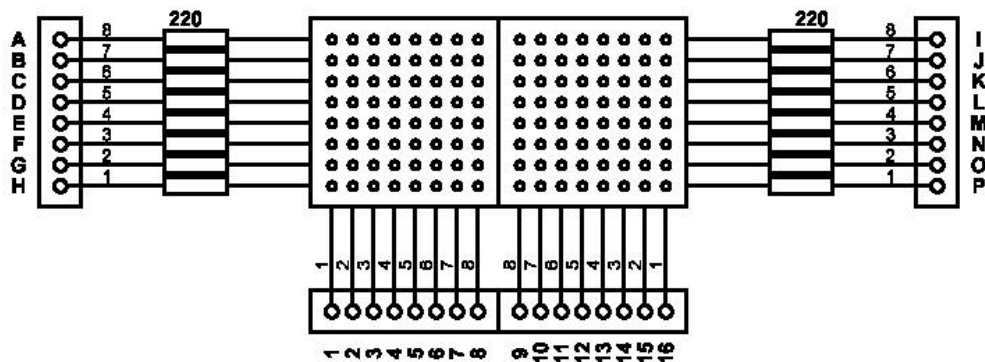
9.1 نمایشگر Dot-matrix

این نمایشگرها مانند شکل 9-1 از تعدادی LED تشکیل شده‌اند که در هر ردیف دارای آندهای مشترک و در هر ستون کاتدهای مشترک می‌باشد. برای روشن شدن هر یک از LEDها بایستی سطر و ستون مورد نظر به درستی تنظیم گردد.



شکل 9-1: 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 LEDها می‌توان به ستون مربوط به همان LED صفر منطقی و به ردیف متناظر یک منطقی اعمال نمود. شماتیک مربوط به این بلوک در شکل 9-2 مشاهده می‌شود.

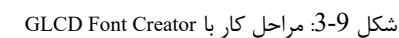


شکل 9-2 - ساختار نمایشگر Dotmatrix روی پکیج آزمایشگاه

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 (مابقی صفر) می‌گردد و سپس مقادیر ستون‌ها با توجه به وضعیت خاموش یا روشن بودن هر LED در این سطر تنظیم می‌شود. سپس بعد از تاخیر زمانی مشخص، سطر دوم برابر با یک (مابقی صفر) می‌گردد و و سپس مقادیر مربوط به ردیف دوم بر روی 8 پایه ستون 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 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، به راحتی وضعیت 64 عدد LED در لحظه مشاهده می‌گردد. این روش همانند ایجاد کاراکترهای خاص (مثلاً علائم و حروف فارسی) در LCDهای کاراکتری می‌باشد.

9.1.1 ایجاد فونت برای نمایش روی Dot-Matrix

برای ایجاد فونت می‌توان از نرم‌افزارهای جانبی مانند GLCD Font Creator استفاده نمود. مراحل مختلف کار با این نرم‌افزار در شکل 9-3 نشان داده شده است.



6: تمام کدها را انتخاب و با copy کردن می‌توان در یک فایل txt ذخیره نمود و یا از save استفاده کرد.

9.1.2 اصول کلی ارسال داده

داده‌های حاصل از نرم‌افزار را به عنوان داده‌های مربوط به **ستون‌های dot-Matrix** مانند برنامه 9-1، و داده‌های مانند برنامه 9-2، به عنوان **داده‌های ردیفی** در نظر گرفته شوند.

برنامه 9-1

```
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 9-2

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

می‌توان به ازای هر نشانه، یک case جدید در جدول جستجو مانند

برنامه 9-3 تشکیل داد. شماره‌ی حرف در متغیر locat ذخیره می‌شود بدین صورت که حرف اول (a) دارای مقدار صفر، حرف دوم (b) دارای مقدار 1 می‌باشد و همین طور تا انتها ادامه می‌یابد. بنابراین باید آدرس locat در تابع lookup نیز به طور صحیح وارد شود.

برنامه 9-3

```
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;
```

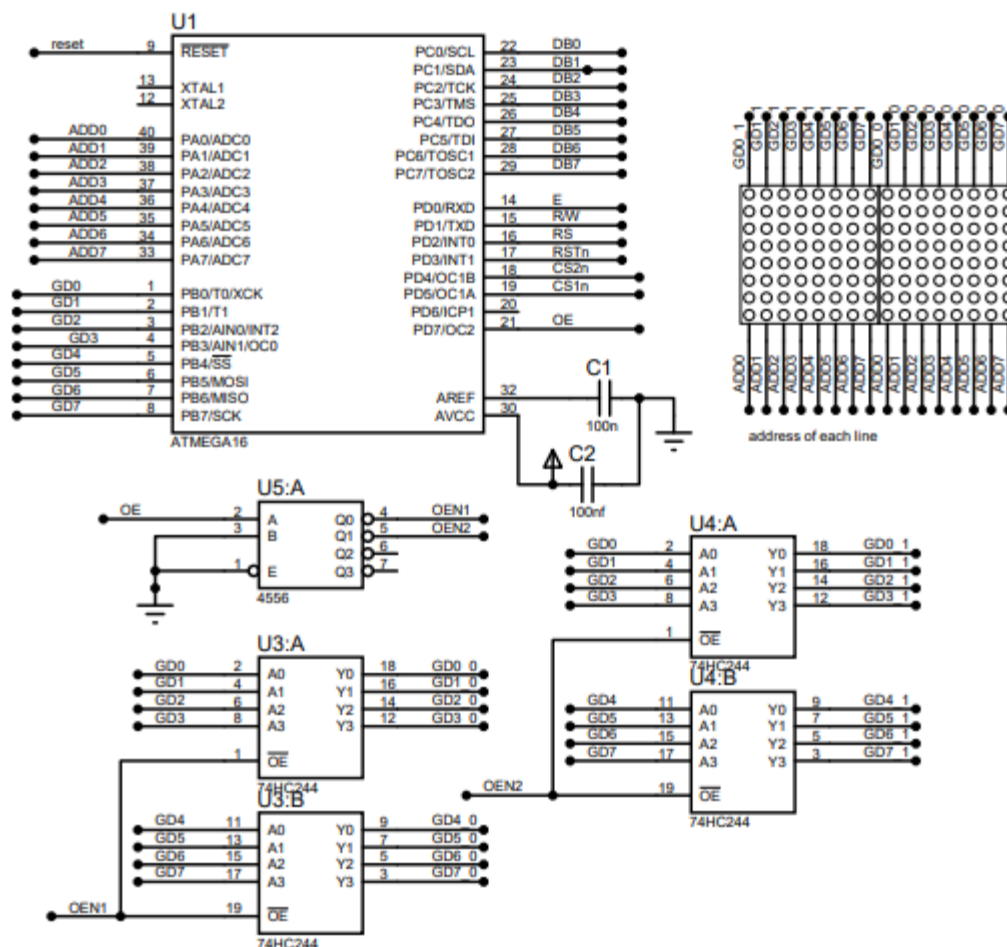
```

case 'C' :
locat=5;
break;
}
}

```

9.1.3 استفاده از چند ماژول Dot matrix

با توجه به این که هر ماژول به دو درگاه جهت سامان دهی ستون ها و ردیف ها نیاز دارد، در صورت استفاده بیش از یک ماژول، می توان سطرهای همه ماژول ها را به یکی از درگاه ها متصل کرد و برای ردیف ها از تعدادی D-Flip Flop استفاده نمود. نمایی از این سخت افزار در شکل 9-4 نشان داده شده است. در این سخت افزار از درگاه A برای سطرها و از درگاه B به صورت مشترک برای ستون ها استفاده شده است. در این سخت افزار با فعال نمودن OE داده های ورودی تراشه در خروجی ثبت می شود و با غیرفعال شدن OE، خروجی ها همچنان ماندگار هستند. بدین ترتیب در بازه زمانی های مناسب می توان داده های هر ستون ماژول Dot matrix را مقدار دهی کرد. برای مدیریت سیگنال های OE، می توان از یک انکدر استفاده نمود تا بتوان با تعداد پایه های مشخصی از میکرو سیگنال های OE بیشتری ایجاد نمود.



شکل 4-9: استفاده از چند ماژول Dot matrix

9.2 معرفی LCD گرافیکی

اساس کار GLCD¹ مشابه Dot-matrix با واحدهای بسیار کوچکتر می باشد که برای ارسال داده ها از روش جاروب کردن سطر استفاده می شود، بدین صورت که در هر مرحله تنها یکی از سطرها یک و مابقی صفر می شوند که در این لحظه، داده های مربوط به این سطر نوشته می شوند. لذا برای تعیین زمان بندی روشن و خاموش شدن هر سطر و ستون نیازمند یک راه انداز² می باشد.

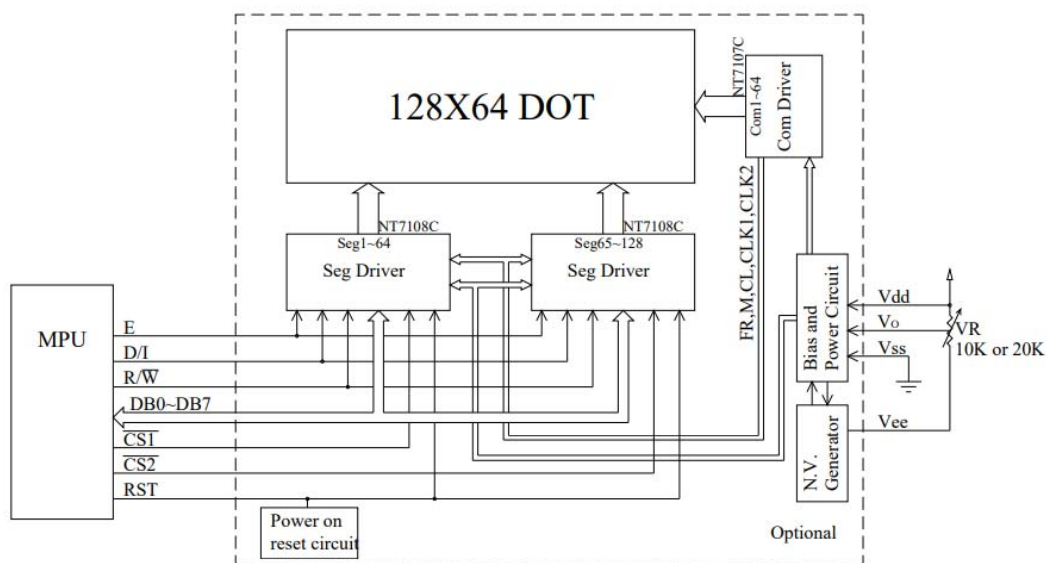
برای راه اندازی ستون های GLCD، تراشه ی NT7108C استفاده می شود. این تراشه، شامل یک RAM، 64-bit data latch و دیکدر می باشد. RAM برای ذخیره داده هایی که از طریق ریزپردازنده منتقل می شود، به کار می رود.

¹ Graphical LCD

² Drive

تراشه NT7107C، نیز یک راهانداز 64 کاناله است که برای در فاصله‌ی زمانی مشخصی، هر یک از سطرهاى GLCD را فعال می‌نماید تا داده‌های ارایه شده توسط NT7108C روی GLCD نمایش داده شود، به عبارتی این تراشه عملیات جاروب کردن سطر را انجام می‌دهد.

GLCD مورد استفاده در این آزمایشگاه دارای ابعاد 128*64 پیکسل می‌باشد و به همین دلیل GLCD دارای دو تراشه NT7108C و یک تراشه‌ی NT7107C برای راهاندازی نمودن 128 ستون و 64 سطر می‌باشد. این تراشه‌ها با KS0107B و KS0108B از شرکت سامسونگ سازگار هستند و می‌تواند در تنظیمات CodeVision جایگزین گردد.



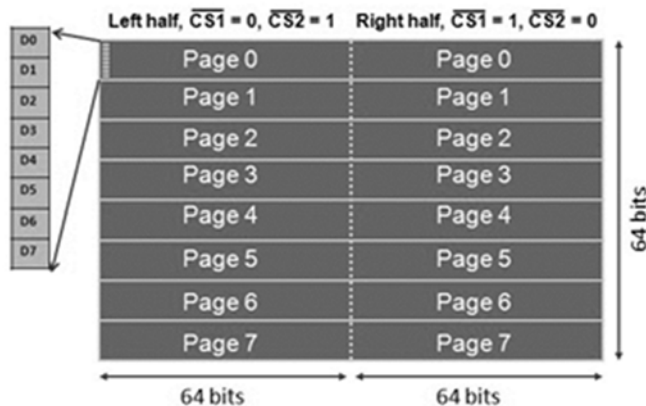
شکل 5-9: Application Diagram

نمایی از ساختار GLCD در شکل 5-9 نشان داده شده است. در این شکل، تراشه‌ی NT7107C، 64 سطرنمایشگر را (COM1 – COM64) را راهاندازی می‌کند. در هر لحظه از زمان تنها یکی از خطوط COM فعال می‌باشد. اولین سگمنت‌های نیمه سمت چپ را فعال می‌کند و دومین NT7108C (IC2) سگمنت‌های نیمه سمت راست را فعال می‌نماید و به هر یک از نیمه‌های نمایشگر می‌توان به طور جداگانه با استفاده از پین‌های CS11 و CS22 دست یافت. هر کدام از نیمه‌ها بر پایه 8 صفحه افقی 0-7 می‌باشد که هر کدام از یک بایت تشکیل شده است.

مقداردهی از صفحه صفر شروع می‌شود. اگر یک بایت داده را به GLCD انتقال داده شود، در اولین ستون صفحه صفر نمایش داده می‌شود. (شکل 6-9) اگر این کار برای 64 بایت داده تکرار شود و سپس در نیمه دوم نیز برای 64 داده دیگر ادامه یابد 8 خط اول GLCD کشیده خواهد شد. برای 8 خط دوم نیز همین روال تکرار می‌گردد.

¹ Chip Select 1

² Chip Select 2



شکل 9-6: pages

9.2.1 آشنایی با پایه‌ها

پایه‌های GLCD مانند جدول 9-1 به دو دسته تقسیم می‌شوند. 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 High داشته باشد، داده‌های موجود در DB[7:0] را نمایش می‌دهد و زمانی که CS1B=L و CS2B=L، E=H توسط CPU می‌تواند خوانده شود.

CS1 و CS2: زمانی که CS1=H و CS2=L باشد تراشه NT7108C (IC1) انتخاب می‌شود و زمانی که CS1=L و CS2=H باشد تراشه NT7108C (IC2) انتخاب می‌شود.

RS: با توجه به **Error! Reference source not found. 16** اگر RS=L باشد، GLCD دستورالعمل‌های دریافتی را اجرا می‌کند در غیر این صورت مقدار خوانده شده از پایه‌های داده را در Data Register قرار می‌دهد.

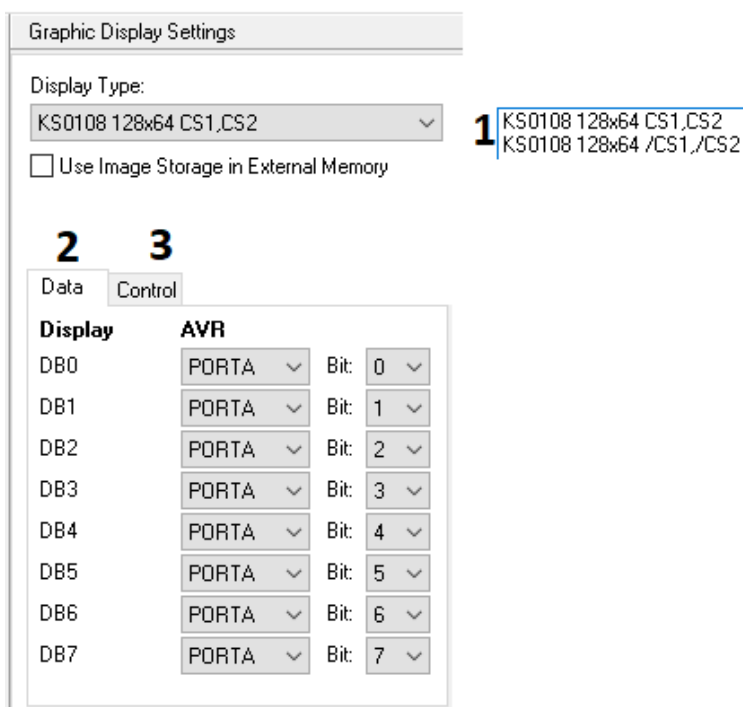
جدول 9-1: مشخصات پایه‌ها

	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

9.3 برنامه نویسی GLCD در محیط کد ویژن

تنظیمات LCD گرافیکی از طریق محیط CodeWizard مانند شکل 7-9 انجام می‌شود. در تنظیمات GLCD لازم است که تراشه کنترل کننده‌ی آن مشخص شود، لذا تراشه‌ی KS0108B انتخاب می‌شود.



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 7-9: تنظیمات کد ویژن برای GLCD

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

Project>configure>C compiler>Libraries>Graphic display

9.4 معرفی فایل سرآیند `glcd.h`

در کدهای ایجاد شده فایل سرآیند جدیدی به نام `glcd.h` فراخوانی می‌شود که به فایل‌های سرآیند حاوی دستورالعمل‌های لازم برای کار کردن با GLCD اشاره دارد. این دستورات 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 Code vision مراجعه گردد.

9.5 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 و چند خط هم روی GLCD رسم گردد. قطعه کد شکل 8-9 چگونگی نمایش یک متن و شکل را نشان می‌دهد و 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 8-9: نمونه کد برای GLCD

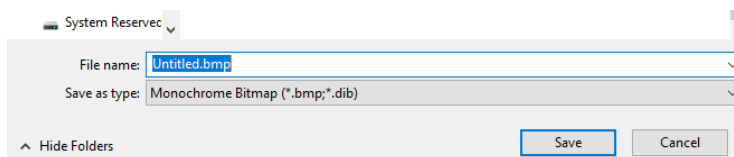
- نکات:
- 1: تعریف ساختار
 - 2: تعیین نوع فوت در ساختار
 - 3: تعیین خواندن از حافظه خارجی
 - 4: تعیین نوشتن در حافظه خارجی
 - 5: پیکربندی GLCD با تنظیمات مشخص شده در ساختار
 - 6: نمایش متن روی GLCD
 - 7: پاک کردن GLCD
 - 8, 10: تنظیم استایل خط قبل از رسم شدن
 - 9: رسم خط

9.6 فونت

معمولاً از دو فونت با ابعاد 5x8 و 8x8 استفاده می‌شود. فونت‌های فارسی معمولاً 8x8 و فونت‌های انگلیسی 5x8 می‌باشند. لذا از فایل سرآیند font5x7.h برای فونت انگلیسی استفاده می‌شود. همچنین می‌توان با استفاده از نرم‌افزارهای طراحی فونت مانند قسمت dot-matrix اقدام به ایجاد فونت نمود.

9.7 نمایش تصویر روی GLCD

با توجه به این که GLCD در پکیج آزمایشگاه، سیاه و سفید است می‌توان تصاویر محدودی را روی آن نشان داد. این تصاویر باید دارای فرمت monochrome باشند که می‌توان در محیط paint عکس مورد نظر را مانند شکل 9-9 در این فرمت ذخیره نمود. همچنین بهتر است اندازه عکس 128*64 پیکسل باشد.

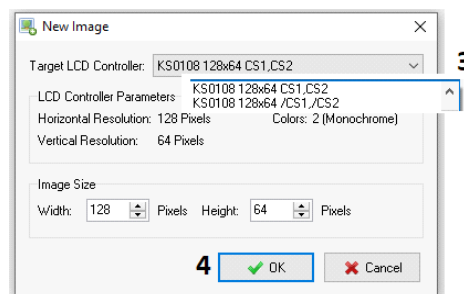


شکل 9-9: ذخیره تصویر در محیط Paint

9.7.1 تهیه کدهای عکس با استفاده از LCD Vision

پس از تهیه عکس مورد نظر می‌توان در محیط LCD Vision کدهای مبنای 16 تصاویر را تهیه نمود که مراحل آن در شکل 9-10 آمده است.

- 1: open LCD Vision software
- 2: file > new Image
- 3: select controller>
- 4: ok



5*: you can draw your picture
 5: file> import image
 6: select your picture then open
 7: On opened window click 'ok'



8: select area for GLCD
 9: file > Export
 10: On opened window click 'ok'
 11: code created on 'Export preview' window
 12: file > save export
 13: On opened window click 'ok'
 14: select path and name then 'save'
 15: 2 file created: 'picthre_name.c' and 'picture_name.h'



شکل 9-10: مراحل تهیه کدهای مبنای 16 تصویر در محیط LCD Vision

9.7.2 برنامه نمایش عکس در محیط Code vision

بهتر است فایل‌های تهیه شده توسط LCD Vision در مسیر پروژه قرار گیرند و در محیط Code vision از طریق مسیر ذیل به آن اضافه گردند.

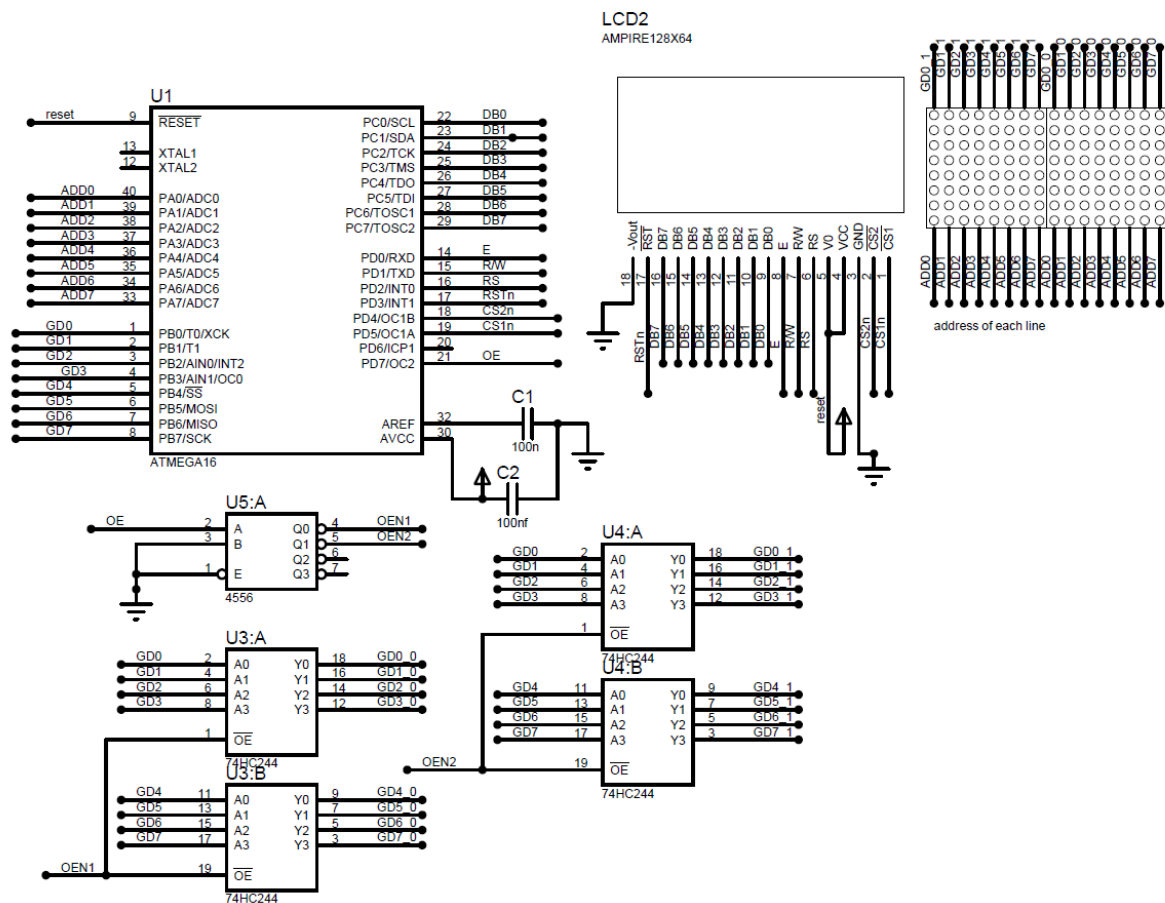
Project > Configure > files > input files > add

همچنین سرآیند فایل ایجاد شده نیز در ابتدای برنامه فراخوانی گردد (# include "picture_name.h"). لازم به ذکر است که نام آرایه‌ی شامل کدهای تصویر نبایستی با عدد شروع شود. در غیر این صورت با خطای کامپایلر همراه خواهد بود. برای رسم تصویر روی GLCD از دستورات موجود در graphic.h به نام gld_putimage استفاده می‌گردد. و چنانچه کدها در حافظه flash ذخیره شده باشد از دستور gld_putimagef استفاده می‌گردد.

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

9.8 برنامه‌های اجرایی مبحث GLCD

سیستم طراحی شده شکل 9-11 را در نظر بگیرید که حاوی دو مجموعه dot matrix می‌باشد و هر کدام برای فعال شدن به دو درگاه (جمعاً 32 پایه) نیاز دارد و از آن جا که در این سخت افزار از GLCD نیز استفاده گردید بایستی تعداد درگاه‌های مورد نیاز برای راه اندازی dot matrix 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 dot matrix را ذخیره نمود و در فاصله زمانی مشخصی آن را به روز نمود. برای تهیه سیگنال OEn بافر نیز می‌توان از یک مالتی پلکسر استفاده کرد تا بتوان تعداد زیادی dot-matrix را درایو نمود. بدین ترتیب فقط 17 پایه از ریزپردازنده برای آن اشغال می‌شود.



شکل 9-11: نمایی از سخت افزار مبحث GLCD و dot matrix

1. زیربرنامه‌ای بنویسید که حرف اول نام و نام خانوادگی شما را روی dot matrix به صورت روان 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 عقربه‌های ساعت، دقیقه و ثانیه شمار 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 و سپس متن روان روی dot-matrix و نهایتاً ساعت روی lcd گرافیکی نمایش داده شود.