

فارسی نویسی در بسکام

نویسنده : اوژن کی نژاد

در این مجموعه از آموزش ها در مورد شیوه های ارتباط با LCD گرافیکی در نرم افزار بسکام و روش های نمایش شکل و فونت های انگلیسی و فارسی توضیح داده می شود. در ابتدا مطالب بر اساس LCD های 128x64 مبتنی بر تراشه کنترل کننده KS108 انجام می شود و در مرحله بعدی در مورد LCD با کنترل کننده T6963c توضیح داده خواهد شد.

کنترل کننده KS108

LCD مبتنی بر تراشه KS108 معمولا دارای ۲۰ پایه هستند که ارتباط سخت افزاری با آن توسط مجموعا ۱۴ بیت انجام می شود. توضیح پایه ها به شرح زیر است:

پایه ۱: GND

پایه ۲: VDD که به ۵ ولت متصل می شود.

پایه ۳: VO این پایه به سر وسط یک پتانسیومتر ۲۰ کیلو اهم متصل می شود. سرهای کناری این پتانسیومتر به ۵+ ولت و پایه ۱۸ (VEE) متصل می شوند.

پایه ۴: D/I که برای تعیین Data یا Instruction بودن اطلاعات بکار می رود. سطح Low به معنای Instruction است.

پایه ۵: R/W سطح Low روی این پایه به معنای نوشتن اطلاعات (Write) و سطح High به معنای خواندن اطلاعات (Read) است.

پایه ۶: Enable برای انتقال اطلاعات به چیپ LCD باید این ورودی یک شود.

پایه های ۷ تا ۱۴: DB0-DB7 خطوط دیتا

پایه ۱۵: CS1 ورودی Chip Select متناظر کنترل کننده نیمه سمت چپ LCD

پایه ۱۶: CS2 ورودی Chip Select متناظر کنترل کننده نیمه سمت راست LCD

پایه ۱۷: Reset

پایه ۱۸: VEE

پایه ۱۹: BL+ تغذیه مثبت BackLight

پایه ۲۰: BL- تغذیه منفی BackLight

خطوط دیتا (DB0-DB7) باید به یکی از پورت های میکروکنترلر متصل شوند که پایه شماره ۷ به بیت با کمترین ارزش و پایه شماره ۱۴ به بیت با بیشترین ارزش در آن پورت متصل شوند. ۶ خط کنترلی دیگر هم باید به یکی دیگر از پورت های AVR متصل شوند. ترتیب اتصال پایه ها در پورت اختیاری است و بعدا در برنامه تعریف می شود. به عنوان مثال فرض کنید خطوط دیتا به PORTA و خطوط کنترل به PORTC مطابق روال زیر متصل شده باشند:

CS1 --- PC7

CS2 --- PC6

RESET --- PC5

ENABLE --- PC4

R/W --- PC3

D/I --- PC2

برای تعریف این آرایش سخت افزاری، در نرم افزار BASCOM باید دستور زیر نوشته شود:

Config Graphlcd = 128 * 64sed , Dataport = Porta , Controlport = Portc , Ce = 6 , Ce2 = 7 ,
Cd = 2 , Rd = 3 , Reset = 5 , Enable = 4

البته دستورات بالا در یک خط نوشته می شوند و اگر جایی لازم باشد به دلیل طولانی بودن دستور، محتوای آن در بیش از یک خط نوشته شود، باید یک علامت _ در انتهای خط آورده شود و ادامه آن در خط بعدی نوشته شود.

توجه کنید که در دستور نوشته شده، Ce2 معادل CS1 و Ce معادل CS2 می باشد. نکته قابل ذکر دیگر در مورد دو پایه مربوط به Backlight می باشد که اگر به جایی متصل نشوند، در بعضی از انواع LCD و به خصوص انواع آبی رنگ امکان رویت اطلاعات نمایش داده شده به درستی ممکن نخواهد بود. برای روشن کردن دائمی نور پس زمینه، می توان تغذیه منفی آن را به زمین و تغذیه مثبت را با سری کردن با مقاومتی در حد چند اهم به ۵ ولت متصل کرد. اگر کنترل نور پس زمینه مورد نظر باشد، این عمل می تواند از طریق یکی از پین های خروجی میکروکنترلر و با واسطه یک بافر ترانزیستوری انجام پذیرد. در صورت انجام روش اخیر، امکان کنترل شدت نور پس زمینه بوسیله روش PWM وجود خواهد داشت.

برای استفاده از کتابخانه مربوط به LCD های گرافیکی با کنترل کننده KS108، باید این کتابخانه را در ابتدای برنامه معرفی کرد. به عنوان مثال فرض کنید که با آرایش سخت افزاری ذکر شده در بخش قبل، LCD به یک mega64 با کریستال ۱۶ مگاهرتز متصل شده باشد. برای فعال سازی LCD باید از دستورات زیر استفاده شود:

```
$regfile = "m64def.dat"
```

```
$crystal = 16000000
```

```
$lib "glcdks108.lbx"
```

```
Config Graphlcd = 128 * 64sed , Dataport = Porta , Controlport = Portc , Ce = 6 , Ce2 = 7 ,  
Cd = 2 , Rd = 3 , Reset = 5 , Enable = 4
```

این توضیح لازم است که کریستال ۱۶ مگاهرتز باید بوسیله تنظیم صحیح فیوزیتهای فعال سازی شده باشد. بعد از این تعاریف می توان از توابع مربوط به LCD برای رسم شکل، نمایش فایل گرافیکی و نوشتن متن با فونت های مختلف استفاده کرد.

اولین دستوری که می تواند برای رسم نقاط در صفحه بکار رود، دستور **PSET X,Y,value** است. در این دستور برای LCD با دقت 128x64، مقدار X در محدوده (0-127) و مقدار Y در محدوده (0-63) می باشد. مقدار value اگر برابر 0 باشد، موجب پاک کردن نقطه و اگر 1 باشد موجب روشن کردن نقطه مورد نظر می شود.

توجه: مبدا مختصات در بالا و گوشه سمت چپ صفحه می باشد و با افزایش Y محل نقطه از بالا به پایین حرکت می کند.

مثال: برنامه زیر یک سیکل از یک تابع سینوسی را روی LCD رسم می کند.

```
$regfile = "m64def.dat"
```

```
$crystal = 16000000
```

```
$lib "glcdks108.lbx"
```

```
Config Graphlcd = 128 * 64sed , Dataport = Porta , Controlport = Portc , Ce = 6 , Ce2 = 7 ,  
Cd = 2 , Rd = 3 , Reset = 5 , Enable = 4
```

```
Dim X As Byte
```

```
Dim Y As Byte
```

```
Dim Theta As Single
```

```
Dim Y1 As Single
```

```
Cls
```

```
For X = 0 To 127
```

```
Theta = 6.28 * X
```

```
Theta = Theta / 128
```

$Y1 = 31 * \text{Sin}(\text{theta})$

$Y = \text{Int}(y1)$

$Y = 31 - Y$

Pset X, Y, 1

Next X

Do

Loop

End

به غیر از دستور PSET، دستورات زیر برای نمایش خط، دایره، مستطیل و مستطیل توپر بکار می روند:

دستور **LINE(x0,y0) - (x1,y1), color** برای رسم خط بین دو نقطه بکار می رود. مقدار $\text{color}=0$ موجب پاک شدن نقاط و مقادیر بزرگتر از صفر موجب رسم نقاط می شود. مثلا دستور $\text{LINE}(0,0) - (127,63), 255$ بالاترین نقطه سمت چپ به پائین ترین نقطه سمت راست، رسم می کند.

دستور **CIRCLE(x0,y0) , radius, color** دایره ای به مرکز $(x0,y0)$ و به شعاع radius رسم می کند. مقدار $\text{color}=0$ موجب پاک شدن نقاط و مقادیر بزرگتر از صفر موجب رسم نقاط می شود.

دستور **BOX (x1,y1) - (x2,y2) , color** برای رسم مستطیل بکار می رود. $(x1,y1)$ و $(x2,y2)$ مختصات دو راس متقابل در مستطیل هستند. مانند دستورات قبل، مقدار غیر صفر برای color سبب رسم نقاط مستطیل و مقدار صفر سبب پاک شدن نقاط آن می شود.

دستور **BOXFILL (x1,y1) - (x2,y2) , color** نیز برای رسم مستطیل بکار می رود و تفاوت آن این است که کلیه نقاط داخل مستطیل را بسته به مقدار color ، رسم یا پاک می کند.

نمایش شکل از روی فایل **BMP**:

یکی از امکانات BASCOM، امکان نمایش فایل های Bitmap طراحی شده در کامپیوتر است. برای این کار ابتدا در محیطی مانند paint یا سایر نرم افزارها فایل مورد نظر در ابعاد متناظر با LCD طراحی می شود (طول ۱۲۸ و عرض ۶۴) و بصورت تک رنگ ذخیره می شود. سپس در محیط BASCOM از منوی Tools گزینه Graphic Converter اجرا می شود. در این محیط باید گزینه SED علامت زده شود و نوع 128×64 به عنوان LCD انتخاب شود. سپس از طریق گزینه Load فایل طراحی شده باز می شود و با گزینه Save با پسوند BGF در محلی که برنامه قرار دارد، ذخیره می شود. در متن برنامه از طریق دستور **SHOWPIC x,y,label**، اطلاعات مورد نظر که در حافظه flash

برنامه ریزی شده در مکان x,y نمایش داده می شود. x و y باید مضربی از ۸ باشند و طول و عرض شکل هم باید از همین قاعده پیروی کند.

توجه : اگر فایل گرافیکی طراحی شده دارای ابعاد 128×64 باشد، برای x,y باید مقادیر $0,0$ در نظر گرفته شود. اما اگر فایل کوچکتر از این مقدار باشد، متناسب با اندازه شکل می توان با تغییر مقادیر x و y شکل را در صفحه جابجا کرد. مثلا اگر شکل اولیه در ابعاد 96×48 طراحی شود، به ازای $x=0$ و $y=0$ شکل در بالا و سمت چپ صفحه نمایش داده می شود و به ازای $x=32$ و $y=16$ شکل در گوشه پائین و سمت راست نمایش داده می شود.

مثال: برنامه زیر شکل موجود در فایل `sample.bgf` را که با ابعاد اولیه 96×48 طراحی شده مانند یک تابلو روان روی صفحه LCD از چپ به راست شیفت می دهد.

```
$regfile = "m64def.dat"
```

```
$crystal = 16000000
```

```
$lib "glcdks108.lbx"
```

```
Config Graphlcd = 128 * 64sed , Dataport = Porta , Controlport = Portc , Ce = 6 , Ce2 = 7 ,  
Cd = 2 , Rd = 3 , Reset = 5 , Enable = 4
```

```
Dim X As Byte
```

```
Cls
```

```
Do
```

```
For X = 0 To 32 Step 8
```

```
Showpic X , 16 , Label
```

```
Waitms 200
```

```
Next X
```

```
Cls
```

```
Loop
```

```
End
```

```
Label:
```

```
$bgf "sample.bgf"
```

دستور **SHOWPICE**: این دستور شبیه دستور SHOWPIC عمل می کند و تفاوت آن در این است که اطلاعات ذخیره شده شکل بجای حافظه flash از eeprom خوانده می شود. برای استفاده از این دستور باید فایل BGF را در eeprom تعریف کرد تا در موقع برنامه ریزی میکروکنترلر در این فضا قرار بگیرد.

مثال:

```
$regfile = "m64def.dat"
```

```
$crystal = 16000000
```

```
$lib "glcdks108.lbx"
```

```
Config Graphlcd = 128 * 64sed , Dataport = Porta , Controlport = Portc , Ce = 6 , Ce2 = 7 ,  
Cd = 2 , Rd = 3 , Reset = 5 , Enable = 4
```

```
Dim X As Byte
```

```
$EEPROMHEX
```

```
$eeprom
```

```
Label:
```

```
$bgf "sample.bgf"
```

```
$data
```

```
Cls
```

```
Do
```

```
For X = 0 To 32 Step 8
```

```
Showpice X , 16 , Label
```

```
Waitms 200
```

```
Next X
```

```
Cls
```

```
Loop
```

```
End
```

در نرم افزار BASCOM این امکان وجود دارد که کاراکترهای ASCII شامل حروف و اعداد و علامت ها روی صفحه نمایش داده شوند و حداقل دو نوع فونت با سایزهای 8x8 و 16x16 قابل تعریف است.

دستور لازم برای نمایش متن بصورت LCDAT Y,X,var[,inv] است که توضیح پارامترهای آن به شرح زیر است:

Y : شماره سطر در محدوده (۱-۸) است و عرض این سطر به اندازه ۸ نقطه (پیکسل) است. در هنگام نمایش فونت های با سایز ۱۶x۱۶، مقادیر Y باید به اندازه ۲ واحد اختلاف داشته باشند تا فونت های موجود در دو سطر متوالی با یکدیگر تداخل نکنند.

X : شماره ستون محل نمایش کاراکتر است که در محدوده (۰-۱۲۷) است.

var : کاراکترهای مورد نظر برای نمایش در این بخش دستور آورده می شوند. این کاراکترها باید بین دو علامت "" قرار بگیرند.

inv : این پارامتر بصورت اختیاری است و اگر مقدار آن غیرصفر باشد سبب نمایش کاراکترها بصورت معکوس می شود. در حالت معکوس نقاط اصلی کاراکتر خاموش می شوند و سایر نقاط اطراف در محدوده همان فونت روشن می شوند.

دو فایل با نام های font8x8.font و font16x16.font در محلی که نرم افزار BASCOM ذخیره شده در فولدر SAMPLES\LCDGRAPH قرار دارند که برای استفاده از فونت های موجود باید هر یک از این دو فایل در فولدری که فایل برنامه در آن قرار دارد، کپی شوند و هر یک از این دو فایل که مورد استفاده قرار می گیرند باید در برنامه include شوند. همچنین دستورات SETFONT font8x8 یا SETFONT font16x16 سبب می شوند که هر یک از این دو فونت در برنامه انتخاب شوند.

مثال : برنامه زیر عبارت GLCD Test را روی سطرهای فرد با فونت 8x8 نمایش می دهد:

```
$regfile = "m64def.dat"
```

```
$crystal = 16000000
```

```
$lib "glcdks108.lbx"
```

```
Config Graphlcd = 128 * 64sed , Dataport = Porta , Controlport = Portc , Ce = 6 , Ce2 = 7 ,  
Cd = 2 , Rd = 3 , Reset = 5 , Enable = 4
```

```
Dim Y as Byte
```

```
Cls
```

```
SetFont Font8x8
```

```
For X = 1 To 7 Step 2
```

```
Lcdat Y , 1 , "GLCD test"
```

```
Next Y
```

Do

Loop

End

```
$include "font8x8.font"
```

نکته مهم: محل قرارگیری دستور `$include "font8x8.font"` اهمیت زیادی دارد و باید در جایی نوشته شود که اجرای خطوط برنامه با آن برخورد نکند و عدم رعایت این مسئله موجب نتایج غیر قابل پیش بینی در اجرای برنامه می شود. این دستور می تواند بعد از حلقه اصلی برنامه مطابق مثال بالا نوشته شود.

مثال: برنامه زیر در دو سطر اول LCD عبارت `Font1` را با فونت `16x16` و در سطر سوم عبارت `Font2` را با فونت `8x8` نمایش می دهد.

```
$regfile = "m64def.dat"
```

```
$crystal = 16000000
```

```
$lib "glcdks108.lbx"
```

```
Config Graphlcd = 128 * 64sed , Dataport = Porta , Controlport = Portc , Ce = 6 , Ce2 = 7 ,  
Cd = 2 , Rd = 3 , Reset = 5 , Enable = 4
```

Cls

```
SetFont Font16x16
```

```
Lcdat 1 , 1 , "Font1"
```

```
SetFont Font8x8
```

```
Lcdat 3 , 1 , "Font2"
```

Do

Loop

End

```
$include "font8x8.font"
```

```
$include "font16x16.font"
```

مثال اخیر نشان می دهد که نوع فونت را می توان در بین اجرای برنامه تغییر داد و فقط باید تمام فونت های مورد نظر در برنامه `include` شده باشند.

فارسی نویسی:

فارسی نویسی در بسکام از جمله مواردی است که مورد نیاز بسیاری از افرادی است که از این نرم افزار استفاده می کنند. اما با مقداری جستجو می توان به این نتیجه رسید که در مورد این بخش اطلاعات کامل و شفافی در دسترس نیست و به همین دلیل در ادامه این سری مقالات به این بحث پرداخته می شود.

اولین شرط لازم برای نوشتن کاراکترهای فارسی، تعریف فونت آن بصورت مناسب برای تمام شکل های مختلف حروف و علامت های مورد استفاده است. برای تعریف این فونت ها باید این مسئله مد نظر قرار داده شود که به دلیل چسبیدن حروف به یکدیگر در فارسی، ارتفاع فونت ها باید بگونه ای طراحی شوند که در هنگام چسبیدن به حروف کناری، یک پیوستگی منطقی بین آنها ایجاد شود. نکته بعدی در حروفی مانند "خ" یا "ق" است که باید علاوه بر حفظ ارتفاع درست برای فونت، در فضای موجود برای تعریف آن وجود قوس و نقطه در حرف هم رعایت شود که در سایزی مانند 8x8، رعایت همه این موارد با یکدیگر کار چندان ساده ای نیست. مسئله بعدی این است که با فرض صحت تعریف فونت ها، چگونه باید آنها را توسط دستور LCDAT احضار کرد و کلمات و جملات مورد نظر را روی LCD ایجاد کرد.

برای حل هر دوی این موارد یک نوع فونت فارسی طراحی شده توسط مولف این مقاله، در اختیار علاقه مندان قرار خواهد گرفت و روش استفاده از آن در اینجا توضیح داده می شود.

روش های مختلفی برای احضار فونت های فارسی قابل تعریف است که یکی از آنها، متناسب کردن یک عبارت به هر فونت است که بجای فونت جایگزین می شود. در این روش برای کنار هم قرار دادن فونت ها از علامت جمع (+) استفاده می شود. به عنوان مثال بر مبنای کدی که در ادامه خواهد آمد، دستور زیر عبارت "سلام" را روی سطر اول LCD نمایش می دهد:

Dim S As String * 3

S = Mim_bozorg + Laam_alef + Sin_koochak

Lcdat 1, 1, S

با این روش، ترکیب حروف فارسی و لاتین هم مقدور است و کاراکترهای استاندارد باید در بین دو علامت "" قرار بگیرند. متن کامل برنامه ای که بتواند عبارت های فارسی و لاتین را بصورت دلخواه بنویسد، در زیر آورده شده است. در این برنامه در سطر اول عبارت "سلام" و در سطر سوم عبارت "فارسی+ENGLISH" نمایش داده می شود.

توجه: برای نمایش کاراکترهای فارسی باید فایل اصلاح شده font8x8.font بجای فایل استاندارد بسکام جایگزین شود. در غیر اینصورت کاراکترهای دیگری بجای کاراکترهای فارسی نمایش داده خواهند شد. این فایل اصلاح شده برای دانلود در سایت قرار داده شده است.

\$regfile = "m64def.dat"

\$crystal = 16000000

\$lib "glcdks108.lbx"

_0 Alias Chr(128)

_1 alias Chr(129)

_2 alias Chr(130)

_3 alias Chr(131)

_4 alias Chr(132)

_5 alias Chr(133)

_6 alias Chr(134)

_7 alias Chr(135)

_8 alias Chr(136)

_9 alias Chr(137)

Virgool alias Chr(138)

Khateh_chasban alias Chr(139)

Alamat_soal alias Chr(140)

A_ba_kolah alias Chr(141)

Hamzeh_chasban alias Chr(142)

Hamzeh_joda alias Chr(143)

Alef alias Chr(144)

Alef_chasbaan alias Chr(145)

Be_bozorg alias Chr(146)

Be_koochak alias Chr(147)

Pe_bozorg alias Chr(148)

Pe_koochak alias Chr(149)

Te_bozorg alias Chr(150)

Te_koochak Alias Chr(151)

Se_bozorg alias Chr(152)
Se_koochak alias Chr(153)
Jim_bozorg alias Chr(154)
Jim_koochak alias Chr(155)
Che_bozorg alias Chr(156)
Che Koochak alias Chr(157)
He_jimi_bozorg alias Chr(158)
He_jimi_koochak alias Chr(159)
Khe_bozorg alias Chr(160)
Khe_koochak alias Chr(161)
Daal alias Chr(162)
Zaal alias Chr(163)
Re Alias Chr(164)
Ze Alias Chr(165)
Zhe alias Chr(166)
Sin_bozorg alias Chr(167)
Sin_koochak alias Chr(168)
Shin_bozorg alias Chr(169)
Shin_koochak alias Chr(170)
Saad_bozorg alias Chr(171)
Saad_koochak alias Chr(172)
Zaad_bozorg alias Chr(173)
Zaad_koochak alias Chr(174)
Taa alias Chr(175)
Zaa alias Chr(176)
Ein_bozorg alias Chr(177)
Ein_akhar_chasban alias Chr(178)

Ein_vasat alias Chr(179)
Ein_koochak alias Chr(180)
Ghein_bozorg alias Chr(181)
Ghein_akhar_chasban alias Chr(182)
Ghein_vasat alias Chr(183)
Ghein_koochak alias Chr(184)
Fe_bozorg alias Chr(185)
Fe_koochak alias Chr(186)
Ghaaf_bozorg alias Chr(187)
Ghaaf_koochak alias Chr(188)
Kaaf_bozorg alias Chr(189)
Kaaf_koochak alias Chr(190)
Gaaf_bozorg alias Chr(191)
Gaaf_koochak alias Chr(192)
Laam_bozorg alias Chr(193)
Laam_alef alias Chr(194)
Laam_koochak alias Chr(195)
Mim_bozorg alias Chr(196)
Mim_koochak alias Chr(197)
Noon_bozorg alias Chr(198)
Noon_koochak alias Chr(199)
Vav Alias Chr(200)
He_koochak alias Chr(201)
He_vasat alias Chr(202)
He_chasbaan_avval alias Chr(203)
Ye_chasbaan alias Chr(204)
Ye_bozorg alias Chr(205)

Ye_koochak alias Chr(206)

Config Graphlcd = 128 * 64sed , Dataport = Porta , Controlport = Portc , Ce = 6 , Ce2 = 7 ,
Cd = 2 , Rd = 3 , Reset = 5 , Enable = 4

SetFont Font8x8

Cls

Dim S As String * 3

Dim S1 As String * 16

S = Mim_bozorg + Laam_alef + Sin_koochak

Lcdat 1 , 1 , S

S1 = "ENGLISH+" + Ye_chasbaan + Sin_koochak + Re + Alef_chasbaan + Fe_koochak

Lcdat 3 , 1 , S1

Do

Loop

End

\$include "font8x8.font"

کنترل کننده T6963c:

در ادامه در مورد LCD های مبتنی بر تراشه کنترل کننده T6963C و با دقت 240x128 به دلیل پر کاربرد بودن توضیح داده می شود. این نوع LCD معمولا دارای ۲۲ پایه هستند و توضیح پایه ها به شرح زیر است:

پایه ۱: FG یا Ground Frame که به زمین متصل می شود.

پایه ۲: GND

پایه ۳: VDD که به ۵ ولت متصل می شود.

پایه ۴: VO این پایه به سر وسط یک پتانسیومتر ۲۰ کیلو اهم متصل می شود. سرهای کناری این پتانسیومتر به +۵ ولت و پایه ۱۹ (VEE) متصل می شوند.

پایه ۵: WR سطح Low روی این پایه به معنای نوشتن اطلاعات (Write) است.

پایه ۶: RD سطح Low روی این پایه به معنای خواندن اطلاعات (Read) است.

پایه ۷: Enable برای انتقال اطلاعات به چیپ LCD باید این ورودی Low شود.

پایه ۸: C/D که اگر Low باشد به معنای خواندن یا نوشتن Data و اگر High باشد به معنی نوشتن Command یا خواندن Status است.

پایه ۹: RESET با اعمال Low سبب RESET شدن چیپ T6963C می شود.

پایه های ۱۰ تا ۱۷: DB0-DB7 خطوط دیتا

پایه ۱۸: FS اگر Low باشد، سایز 8x8 و اگر High باشد، سایز 7x8 برای فونت لاتین انتخاب می شود.

پایه ۱۹: VEE

پایه ۲۰: +BL تغذیه مثبت BackLight

پایه ۲۱: -BL تغذیه منفی BackLight

پایه ۲۲: به جایی متصل نیست.

از طریق دستور Config Graphlcd نوع ارتباط این پایه ها با پورت های میکروکنترلر تعیین می شود. مثال:

$\text{Config Graphlcd} = 240 * 128$, Dataport = Portc , Controlport = Portd , Ce = 7 , Cd = 4 , Wr = 5 , Rd = 6 , Reset = 3 , Fs = 2 , Mode = 8

از جمله تفاوت های موجود این است که در ابتدای برنامه نیازی به include کردن کتابخانه ای برای این نوع LCD نیست. برای رسم نقاط و خط و دایره و مربع و فایل BMP، می توان از دستوراتی که در بخش سوم و چهارم این مقاله توضیح داده شد، استفاده کرد و تفاوت در استفاده از این دستورات در محدوده مختصات است که مختصات محور افقی در محدوده (۰-۲۳۹) و محور عمودی در محدوده (۰-۱۲۷) است.

دستور LCDAT که در نوع مبتنی بر ks108 برای نمایش متن بکار می رفت، در اینجا کاربردی ندارد و بجای آن باید از ترکیب دستورات LOCATE و LCD استفاده شود. دستور LOCATE Y,X محل مکان نما را به مختصات (X,Y) منتقل می کند که X در محدوده (۰-۳۰) و Y در محدوده (۰-۱۶) است. بعد از قرار گرفتن مکان نما در محل مناسب از طریق دستور LCD k می توان متن مورد نظر را روی LCD نمایش داد.

برای کار با این نوع LCD، مثال های مختلفی در Help نرم افزار بسکام و سایر منابع وجود دارد. اما یکی از مواردی که مورد سوال اکثر کسانی است که قصد راه اندازی این نوع LCD را دارند، شیوه فارسی نویسی در بسکام است که در مورد آن توضیح داده خواهد می شود.

برای نوشتن فونت های فارسی می توان به دو روش ارتباط مستقیم با T6963C و بدون استفاده از توابع بسکام و یا ساخت فونت بصورت نقطه به نقطه و با استفاده از دستور Pset عمل کرد. در این مقاله از روش دوم برای ایجاد فونت های فارسی 8x8 روی LCD استفاده شده است.

به این منظور یک روال به شکل زیر تعریف شده که با استفاده از آن می توان در مختصات دلخواه، فونت مورد نظر را نمایش داد:

Lcdfarsi (X , Y , FONT)

مقدار X در محدوده (1-30) و مقدار Y در محدوده (1-16) است و بجای فونت فهرستی از نمادها که قبلا هم در فارسی نویسی روی LCD های 128x64 معرفی شده بود، قرار می گیرد (مانند vav یا alef_chasbaan و مانند آن). برای احضار این روال باید از دستور Call استفاده شود. به عنوان مثال دستور زیر حرف "ق" را در بالا و سمت چپ صفحه نمایش می دهد:

Call Lcdfarsi(1 , 1 , Ghaaf_bozorg)

برای نوشتن یک جمله و صرفه جویی در نوشتن دستورات عمل ها، می توان در یک حلقه از دستور Lookup استفاده کرد که نمونه ای از آن در مثال انتهای مقاله آورده شده است.

اگر لازم باشد ترکیبی از حروف فارسی و لاتین نمایش داده شوند، برای حروف لاتین از ترکیب دستورات Locate و LCD استفاده می شود که این عمل هم در مثال انجام شده است. فونت فارسی از نظر شکل مشابه همان فونتی است که قبلا برای LCD های 128x64 با کنترل کننده ks108 طراحی شد.

مثال: برنامه زیر در سطر اول LCD از راست، عبارت فارسی " دانایی توانایی است " و در سطر سوم از سمت چپ عبارت "Knowledge is POWER" را نمایش می دهد:

' Written by Ozhan Keynezhad

```
$regfile = "m64def.dat"
```

```
$crystal = 8000000
```

```
$hwstack = 32
```

```
$swstack = 10
```

```
$framesize = 40
```

```
Const_0 = 0
```

```
Const_1 = 1
```

```
Const_2 = 2
```

Const_3 = 3
Const_4 = 4
Const_5 = 5
Const_6 = 6
Const_7 = 7
Const_8 = 8
Const_9 = 9
Const Virgool = 10
Const Khateh_chasban = 11
Const Alamat_soal = 12
Const A_ba_kolah = 13
Const Hamzeh_chasban = 14
Const Hamzeh_joda = 15
Const Alef = 16
Const Alef_chasbaan = 17
Const Be_bozorg = 18
Const Be_koochak = 19
Const Pe_bozorg = 20
Const Pe_koochak = 21
Const Te_bozorg = 22
Const Te_koochak = 23
Const Se_bozorg = 24
Const Se_koochak = 25
Const Jim_bozorg = 26
Const Jim_koochak = 27
Const Che_bozorg = 28
Const Che Koochak = 29

Const He_jimi_bozorg = 30
Const He_jimi_koochak = 31
Const Khe_bozorg = 32
Const Khe_koochak = 33
Const Daal = 34
Const Zaal = 35
Const Re = 36
Const Ze = 37
Const Zhe = 38
Const Sin_bozorg = 39
Const Sin_koochak = 40
Const Shin_bozorg = 41
Const Shin_koochak = 42
Const Saad_bozorg = 43
Const Saad_koochak = 44
Const Zaad_bozorg = 45
Const Zaad_koochak = 46
Const Taa = 47
Const Zaa = 48
Const Ein_bozorg = 49
Const Ein_akhar_chasban = 50
Const Ein_vasat = 51
Const Ein_koochak = 52
Const Ghein_bozorg = 53
Const Ghein_akhar_chasban = 54
Const Ghein_vasat = 55
Const Ghein_koochak = 56

Const Fe_bozorg = 57
Const Fe_koochak = 58
Const Ghaaf_bozorg = 59
Const Ghaaf_koochak = 60
Const Kaaf_bozorg = 61
Const Kaaf_koochak = 62
Const Gaaf_bozorg = 63
Const Gaaf_koochak = 64
Const Laam_bozorg = 65
Const Laam_alef = 66
Const Laam_koochak = 67
Const Mim_bozorg = 68
Const Mim_koochak = 69
Const Noon_bozorg = 70
Const Noon_koochak = 71
Const Vav = 72
Const He_koochak = 73
Const He_vasat = 74
Const He_chasbaan_avval = 75
Const Ye_chasbaan = 76
Const Ye_bozorg = 77
Const Ye_koochak = 78

Config Graphlcd = 240 * 128 , Dataport = Portc , Controlport = Portd , Ce = 7 , Cd = 4 , Wr = 5 , Rd = 6 , Reset = 3 , Fs = 2 , Mode = 8

Declare Sub Lcdfarsi(byval X As Byte , Byval Y As Byte , Byval Font As Byte)

Dim X As Byte

Dim Y As Byte

Dim Font As Byte

Dim X1 As Byte

Dim Y1 As Byte

Dim X2 As Byte

Dim Y2 As Byte

Dim Y3 As Byte

Dim Mask As Byte

Dim F As Byte

Dim F1 As Byte

Dim Table As Word

Dim N As Byte

Cls

Cursor Off

For N = 0 To 15

X = N + 15

Font = Lookup(n , Word_table)

Call Lcdfarsi(x , 1 , Font)

Next N

Locate 3 , 1

Lcd "Knowledge is POWER"

Do

Loop

End

'*****'

Sub Lcdfarsi(byval X As Byte , Byval Y As Byte , Byval Font As Byte)

If X > 30 Then X = 3

If X = 0 Then X = 1

If Y > 16 Then Y = 16

If Y = 0 Then Y = 1

If Font > Ye_koochak Then Font = Ye_koochak

X = X - 1 : Y = Y - 1

Table = Font * 8

X1 = X * 8

Y1 = Y * 8

For X = 0 To 7

F = Lookup(table , Font_table)

For Y = 0 To 7

Y3 = Y

Mask = 2 ^ Y3

F1 = F And Mask

X2 = X + X1

Y2 = Y + Y1

If F1 <> 0 Then Pset X2 , Y2 , 1

Next Y

Table = Table + 1

Next X

End Sub

'*****'

Word_table:

Data Te_bozorg , Sin_koochak , Alef

Data Ye_chasbaan , Ye_koochak , Alef_chasbaan , Noon_koochak , Alef , Vav , Te_koochak

Data Ye_chasbaan , Ye_koochak , Alef_chasbaan , Noon_koochak , Alef , Daal

'*****'

Font_table:

Data &H00 , &H00 , &H30 , &H48 , &H30 , &H00 , &H00 , &H00

DATA &H00 , &H00 , &H00 , &HFC , &H00 , &H00 , &H00 , &H00

DATA &H00 , &H00 , &H00 , &HFC , &H10 , &H1C , &H00 , &H00

DATA &H00 , &HFC , &H10 , &H1C , &H10 , &H1C , &H00 , &H00

DATA &H00 , &HFC , &H10 , &H18 , &H14 , &H14 , &H00 , &H00

DATA &H00 , &HE0 , &H98 , &HC4 , &H98 , &HE0 , &H00 , &H00

DATA &H00 , &H80 , &HD8 , &H64 , &H24 , &H08 , &H00 , &H00

DATA &H00 , &H0C , &H70 , &HC0 , &H70 , &H0C , &H00 , &H00

DATA &H00 , &HC0 , &H38 , &H0C , &H38 , &HC0 , &H00 , &H00

DATA &H00 , &H00 , &H1C , &H14 , &HFC , &H00 , &H00 , &H00

DATA &H00 , &H00 , &H60 , &HD0 , &H40 , &H00 , &H00 , &H00
DATA &H20 , &H20 , &H20 , &H20 , &H20 , &H20 , &H20 , &H20
DATA &H00 , &H00 , &H0C , &HB2 , &H02 , &H04 , &H00 , &H00
DATA &H00 , &H08 , &H04 , &H7C , &H04 , &H02 , &H00 , &H00
DATA &H20 , &H20 , &H34 , &H2A , &H2A , &H20 , &H20 , &H18
DATA &H00 , &H00 , &H20 , &H54 , &H2A , &H0A , &H00 , &H00
DATA &H00 , &H00 , &H00 , &H00 , &H7E , &H00 , &H00 , &H00
DATA &H00 , &H00 , &H00 , &H00 , &H1E , &H20 , &H20 , &H20
DATA &H00 , &H18 , &H20 , &H20 , &HA0 , &H20 , &H20 , &H18
DATA &H20 , &H20 , &H20 , &H20 , &HA0 , &H20 , &H20 , &H18
DATA &H00 , &H18 , &H20 , &H60 , &HA0 , &H60 , &H20 , &H18
DATA &H20 , &H20 , &H20 , &H60 , &HA0 , &H60 , &H20 , &H18
DATA &H00 , &H18 , &H20 , &H24 , &H20 , &H24 , &H20 , &H18
DATA &H20 , &H20 , &H20 , &H24 , &H20 , &H24 , &H20 , &H18
DATA &H00 , &H18 , &H20 , &H24 , &H22 , &H24 , &H20 , &H18
DATA &H20 , &H20 , &H20 , &H24 , &H22 , &H24 , &H20 , &H18
DATA &H00 , &H00 , &H70 , &H98 , &H8A , &HAA , &H8C , &H18
DATA &H20 , &H20 , &H20 , &H28 , &HA4 , &H14 , &H18 , &H30
DATA &H00 , &H00 , &H70 , &H98 , &HAA , &HCA , &HAC , &H18
DATA &H20 , &H20 , &H20 , &H68 , &HA4 , &H54 , &H18 , &H30
DATA &H00 , &H00 , &H70 , &H98 , &H8A , &H8A , &H8C , &H18
DATA &H20 , &H20 , &H20 , &H28 , &H24 , &H14 , &H18 , &H30
DATA &H00 , &H00 , &H00 , &H60 , &H94 , &H95 , &H98 , &H10
DATA &H20 , &H20 , &H20 , &H28 , &H25 , &H14 , &H18 , &H30
DATA &H00 , &H00 , &H00 , &H00 , &H20 , &H44 , &H48 , &H30
DATA &H00 , &H00 , &H00 , &H00 , &H20 , &H40 , &H4A , &H30
DATA &H00 , &H00 , &H00 , &H00 , &H80 , &H80 , &H40 , &H38

DATA &H00 , &H00 , &H00 , &H00 , &H80 , &H80 , &H40 , &H3A
DATA &H00 , &H00 , &H00 , &H00 , &H80 , &H84 , &H42 , &H34
DATA &H00 , &H60 , &H80 , &H70 , &H20 , &H30 , &H20 , &H18
DATA &H20 , &H20 , &H20 , &H30 , &H20 , &H30 , &H20 , &H18
DATA &H00 , &H60 , &H80 , &H70 , &H24 , &H32 , &H24 , &H18
DATA &H20 , &H20 , &H20 , &H30 , &H24 , &H32 , &H24 , &H18
DATA &H00 , &H60 , &H80 , &H60 , &H30 , &H28 , &H28 , &H18
DATA &H20 , &H20 , &H30 , &H20 , &H30 , &H28 , &H28 , &H18
DATA &H00 , &H60 , &H80 , &H60 , &H30 , &H28 , &H2A , &H18
DATA &H20 , &H20 , &H30 , &H20 , &H30 , &H28 , &H2A , &H18
DATA &H00 , &H20 , &H20 , &H3E , &H30 , &H28 , &H28 , &H18
DATA &H20 , &H20 , &H20 , &H3E , &H30 , &H28 , &H2A , &H18
DATA &H00 , &H00 , &H00 , &H00 , &H40 , &HB0 , &HA8 , &HA8
DATA &H00 , &H00 , &H00 , &H00 , &H60 , &HAC , &H94 , &HAC
DATA &H20 , &H20 , &H20 , &H2C , &H14 , &H2C , &H20 , &H20
DATA &H20 , &H20 , &H20 , &H20 , &H20 , &H30 , &H28 , &H28
DATA &H00 , &H00 , &H00 , &H00 , &H40 , &HB0 , &HAA , &HA8
DATA &H00 , &H00 , &H00 , &H00 , &H60 , &HAC , &H96 , &HAC
DATA &H20 , &H20 , &H20 , &H2C , &H15 , &H2C , &H20 , &H20
DATA &H20 , &H20 , &H20 , &H20 , &H20 , &H30 , &H2A , &H28
DATA &H00 , &H10 , &H20 , &H20 , &H20 , &H38 , &H2A , &H38
DATA &H20 , &H20 , &H20 , &H20 , &H20 , &H38 , &H2A , &H38
DATA &H00 , &H00 , &H70 , &H80 , &HBA , &HA8 , &HBA , &H70
DATA &H20 , &H20 , &H20 , &H20 , &H20 , &H3A , &H28 , &H3A
DATA &H00 , &H30 , &H20 , &H20 , &H20 , &H28 , &H34 , &H32
DATA &H20 , &H20 , &H20 , &H20 , &H20 , &H28 , &H34 , &H32
DATA &H00 , &H30 , &H20 , &H20 , &H24 , &H2A , &H34 , &H32

DATA &H20 , &H20 , &H20 , &H20 , &H24 , &H2A , &H34 , &H32
DATA &H00 , &H00 , &H60 , &H80 , &H80 , &H80 , &H80 , &H7E
DATA &H00 , &H00 , &H00 , &H00 , &H3E , &H20 , &H20 , &H1E
DATA &H20 , &H20 , &H20 , &H20 , &H20 , &H20 , &H20 , &H1E
DATA &H00 , &H00 , &HE0 , &H10 , &H30 , &H48 , &H48 , &H30
DATA &H20 , &H20 , &H20 , &H20 , &H20 , &H50 , &H48 , &H30
DATA &H00 , &H00 , &H00 , &H70 , &H80 , &H84 , &H80 , &H70
DATA &H20 , &H20 , &H20 , &H20 , &H24 , &H20 , &H20 , &H18
DATA &H00 , &H00 , &H00 , &H00 , &H00 , &HB8 , &HA4 , &H7C
DATA &H00 , &H00 , &H00 , &H00 , &H00 , &H18 , &H24 , &H38
DATA &H20 , &H20 , &H30 , &H4C , &H78 , &H48 , &H30 , &H20
DATA &H20 , &H20 , &H20 , &H30 , &H4C , &H78 , &H48 , &H30
DATA &H00 , &H60 , &H80 , &H80 , &HA0 , &H50 , &H10 , &H20
DATA &H00 , &H60 , &H80 , &H80 , &H98 , &HA4 , &H64 , &H04
DATA &H20 , &H20 , &H20 , &HA0 , &H20 , &HA0 , &H20 , &H18