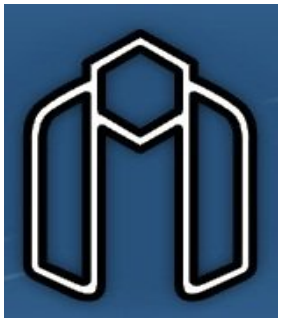
به نام خدا

TFTLCDمعروف به LCD N96 چینی

گردآورنده

علی ایمانی فر

دانشکده مهندسی برق



دانشگاه صنعتی شاهرود

کلمات کلیدی

TFT LCD,AVR,CodeVision AVR

چکیده

در این مقاله مختصری در باره ی LCD N96**چینی و نحوه ی راه اندازی آن توسط میکروکنترلر** AVR **با استفاده از کامپایلر** CodeVision AVR **آورده شده است.**

فهرست

مقدمه.....3
خصوصیات.....3
شناسایی LCD.....4
دسترسی به پایه های LCD.....4
شناختی کلی بر پایه های LCD.....4
کتابخانه tft_functions.h.....6
توابع.....9
دسته اول توابع.....9
دسته دوم توابع.....9
دسته سوم توابع.....11
ضمیمه 1.....13
منابع.....14

راه اندازی TFT LCD(معروف به LCD N96چینی)

مقدمه

یکی از راههای نمایش اطلاعات وبرقراری ارتباط دستگاهها با انسان استفاده از LCD است . LCDهای موجود در بازار و شناخته شده برای کاربران الکترونیک معمولا LCDهای کارکتری وگرافیکی (سیاه سفید)هستند. LCDهای کارکتری محدودیت بالایی برای نمایش اطلاعات دارند و LCDهای گرافیکی (سیاه سفید)نیز محیط یکنواخت و سیاه سفید و خسته کننده با وضوح پایین دارند .پس نیاز به یک LCDرنگی با وضوح بالا داریم تا بتوانیم از این مشکلات رهایی یابیم .در این جا به معرفی یکی از LCDهای به کار برده شده در گوشی های چینی می پردازیم با نام ELT240320ATP که معروف به LCD N96 چینی است .

این LCD ها معمولا دارای یکی از کنترلر های ILI9325 و ILI9320 هستند .کنترلهای دیگری نیز وجود دارند ولی این مقاله فقط مختص به دو کنترلر ذکر شده است.

خصوصیات :

1:ابعاد 2.8 اینچی

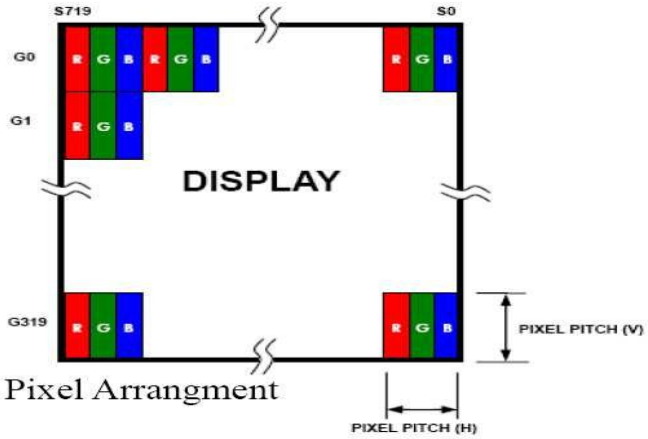
2:تعداد پیکسل 240*320

3:دارای دو مد دیتا باس 8 و 16بیتی

4:توان مصرفی پایین

5:ولتاژ3.3 ولتی

6:دارای یک تاچ اسکرین



Parameter	Value	Unit
LCD Mode	a-Si TFT/transmissive	-
Color	262K	-
Display Resolution	240*RGB*320	pixels
OUTLINE DIMEMSIONS	50.00(W) x69.20(H) x4.05(T)	mm
Active Area(A.A)	43.20 (W) x 57.60(H)	mm
Pixel Arrangement	RGB-stripe	-
Viewing Direction	12 O'clock	
Display Mode	Normally white	
LCD Controller/Driver	ILI9325	-
IC Package Type	COG	-
MPU interface	Standard 8080 system18-/16-bit paraller	-
Power Supply Voltage	2.5~3.3	V
Back-light	White LED*4	pcs

شناسایی LCD

LCD گوشی های چینی مدل های بسیار زیادی دارد (حدود 200–300 نوع) که فقط تعداد انگشت شماری توسط این مقاله قابل راه اندازی اند .برای تشخیص این LCD باید فلت LCD موارد زیر را دارا باشد.

1:37 پایه باشد .

2: پایه های 12 تا15 مختص تاچ اسکرین باشد .

3:پایه NC 21 (Not Connected) باشد .

4:پایه های16 تا 20به قسمت بک لایتLCD رفته باشد .

5:پایه های 5و 34 زخیم تر از پایه های دیگر باشند .

اگر فلت LCD تمام موارد بالا را تایید کرد شما می توانید LCD را تهیه کنید .

دسترسی به پایه های LCD

فاصله روی پایه های فلت این LCD یک میلی متر است که فاصله خیلی کمی است برای دسترسی راحت تر به این پایه ها راههای مختلفی وجود دارد از جمله :

1:لحیم کردن سیم های خیلی نازک به هر پایه .

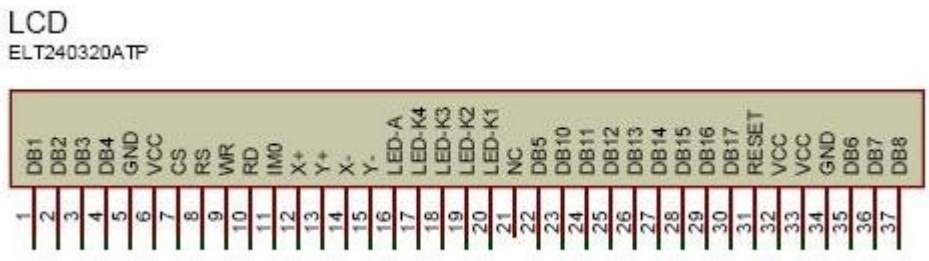
2:ساخت برد مدار چاپی که پایه هایLCD را تبدیل به پایه های قابل استفاده برای بِرد بُرد کند .

3:استفاده از کابل هارد کامپیوتر بدین صورت که پایه های LCD را به کابل لحیم می کنیم واز کانکتور IDE مربوط به کابل به عنوان خروجی استفاده می کنیم .

توجه:

در بازار سوکتی برای این LCD وجود ندارد .

شناختی کلی بر پایه های LCD



*پایه های تغذیه LCD(VCCوGND):این پایه ها مجموعا 5 پایه از LCD را به خود اختصاص داده اند که ولتاژ های حدود 3.3 ولت نیاز دارند.که پایه های 5 و 34 مختص GNDو پایه های 6و 32و 33مختص VCC اند .

***:پایه های انتقال Data :(DB1-DB8و DB10-DB17):این 16 پایه که دیتا باس یا گذر گاه داده نامیده می شود برای انتقال داده ها وکد دستورات بکار میروند .**

***:پایه Reset:این پایه در صورتی که صفر شود کنترلر LCD را باز نشانی میکند .**

***پایهRS:این پایه وظیفه انتخاب رجیستر دستورالعمل یا رجیستر داده را دارد .**

***پایه CS:پایه انتخاب تراشه است.**

***پایهRW:در صورتی که این پایه صفر شود قصد نوشتن داده یا دستور در رجیستر های دستورالعمل یا داده را داریم.**

***پایه RD:در صورتی که این پایه صفر شود قصد خواندن داده ها از روی LCD را داریم.**

***IM0: این پایه اگر صفر باشدمد 16 بیتی و اگر یک باشد مد 8بیتی (DB10-DB17) انتخاب می شود .**

توجه:

جای 3 عدد مقاومت روی فلت LCD وجود دارد R1,R2,R3 اگر R1 را لحیم کنیم مد 8 بیتی را انتخاب می کنید و اگر R2را لحیم کنیم مد 16 بیتی را انتخاب کرده ایم واگر R3 را لحیم کنیم IM0 مد ارسال داده را تعیین می کند .(در این مقاله ما LCD را به صورت 16 بیتی راه اندازی می کنیم . بنابراین باید IM0 را به صفر ولت متصل کنیم.)

***پایه NC :(شماره 21): این پایه بدون اتصال است .**

***پایه های X+,X-,Y+,Y-:این چهار پایه مختص تاچ اسکرین است.**

***پایه های LEDAوLED-K1وLED-K2وLED-K3وLED-K4:این پایه ها مربوط به بک لایت می باشد که LED-A آند چهار دیود نورانی است ومابقی کاتد اند.**

این پایه ها باید توسط ولتاژ 3.3 ولت تغذیه شوند .

کتابخانه tft_functions.h

قبل از فرا خوانی فایل سرآمد کتابخانه چند کار را باید انجام دهید :

1– ابتدا باید فایل *tft_functions.h* را در پوشه inc موجود در محل نصب نرم افـزار *CodeVision AVR* کپـی کنید .(به عنوان مثال *C:\cvavr\inc*)

2– ابتدا باید *PORT* مربوط به دیتاپورت وکنترل پورت را مشخص کنید .برای این منظور باید از الگوی زیـر اسـتفاده کنیم .

```
#define LCD_RST_DDR    DDRX
#define LCD_RST_PORT   PORTX
#define LCD_RST_PIN    N
```

X : پورت مربوط به کنترلر
N : پین مربوط به RST

```
#define LCD_RS_DDR     DDRX
#define LCD_RS_PORT    PORTX
#define LCD_RS_PIN     N
```

X : پورت مربوط به کنترلر
N : پین مربوط به RS

```
#define LCD_CS_DDR     DDRX
#define LCD_CS_PORT    PORTX
#define LCD_CS_PIN     N
```

X : پورت مربوط به کنترلر
N : پین مربوط به CS

```
#define LCD_RD_DDR     DDRX
#define LCD_RD_PORT    PORTX
#define LCD_RD_PIN     N
```

X : پورت مربوط به کنترلر
N : پین مربوط به RD

```
#define LCD_WR_DDR     DDRX
#define LCD_WR_PORT    PORTX
#define LCD_WR_PIN     N
```

X : پورت مربوط به کنترلر
N : پین مربوط به WR

```
#define LCD_DATAPORT_MSB_DDR     DDRX
#define LCD_DATAPORT_MSB_PORT    PORTX
#define LCD_DATAPORT_MSB_PIN     PINX
```

X : پورت مربوط به دیتا باس
DB10-DB17

```
#define LCD_DATAPORT_LSB_DDR     DDRX
#define LCD_DATAPORT_LSB_PORT    PORTX
#define LCD_DATAPORT_LSB_PIN     PINX
```

X : پورت مربوط به دیتا باس
DB1-DB8

```
#define LCD_CONTROLPORT_DDR      DDRX
#define LCD_CONTROLPORT_PORT     PORTX
#define LCD_CONTROLPORT_PIN      PINX
```

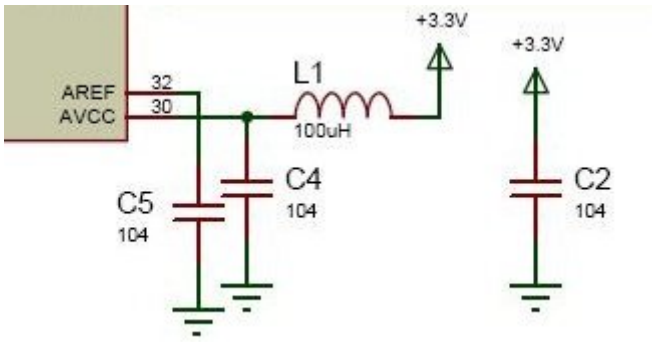
X : پورت مربوط به کنترلر(همان پورتی که در بالا برای RST,RS,CS,RD, WR معرفی کردیم)

در این الگو ما یکی از پورت های میکرو را به *DB1-DB8* و پورت دیگر رابه *DB10-DB17* و 5 پین از یک پورت دیگر را به کنترلر *LCD* اختصاص می دهیم .

به عنوان مثال *PORTC* را به *DB1-DB8* و*PORTD*را *DB10-DB17* و *PORTB.0*را به *RST* و *PORTB.1*را به *RS* و *PORTB.2*را به *CS* و *PORTB.3*را به *RD* و *PORTB.4*را به *WR* اختصاص دهیم باید کد زیر را در برنامه قرار دهیم .

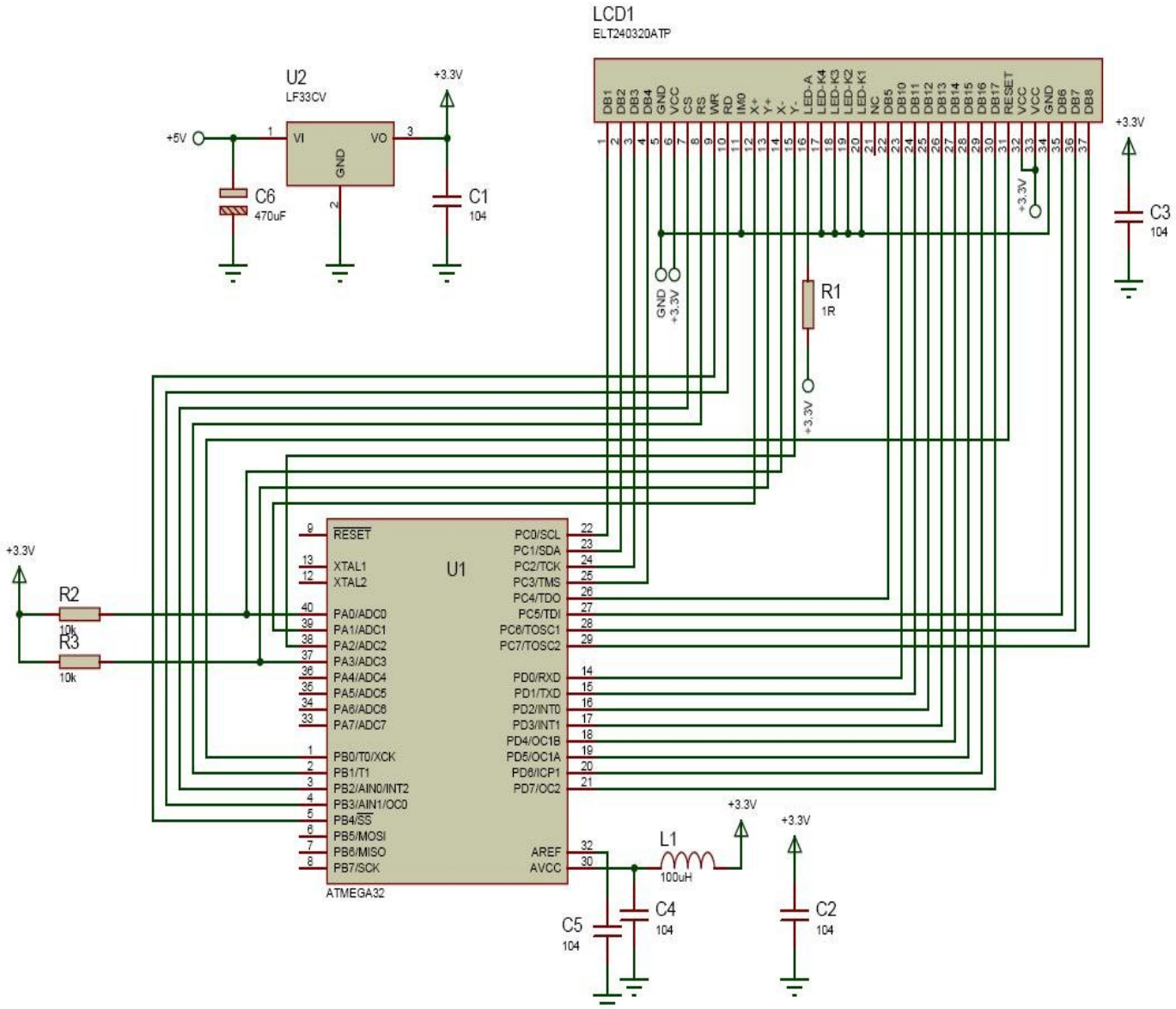
```
#define LCD_RST_DDR     DDRB
#define LCD_RST_PORT    PORTB
#define LCD_RST_PIN      0

#define LCD_RS_DDR      DDRB
#define LCD_RS_PORT     PORTB
#define LCD_RS_PIN      1

#define LCD_CS_DDR      DDRB
#define LCD_CS_PORT     PORTB
#define LCD_CS_PIN      2

#define LCD_RD_DDR      DDRB
#define LCD_RD_PORT     PORTB
#define LCD_RD_PIN      3

#define LCD_WR_DDR      DDRB
#define LCD_WR_PORT     PORTB
#define LCD_WR_PIN      4

#define LCD_DATAPORT_MSB_DDR        DDRD
#define LCD_DATAPORT_MSB_PORT       PORTD
#define LCD_DATAPORT_MSB_PIN        PIND

#define LCD_DATAPORT_LSB_DDR        DDRC
#define LCD_DATAPORT_LSB_PORT       PORTC
#define LCD_DATAPORT_LSB_PIN        PINC

#define LCD_CONTROLPORT_DDR         DDRB
#define LCD_CONTROLPORT_PORT        PORTB
#define LCD_CONTROLPORT_PIN         PINB
```

شماتیک هم در پایین آورده شده . دقت کنید که شماتیک پایین با توجه به تعاریف بالا رسم شده است . در شماتیک پـــایین مدار تاچ اسکرین نیز بسته شده است ولی ما قصد راه اندازی تاچ اسکرین را نداریم (البته در این مقاله) پس لازم نیســت شما این قسمت را ببندید.



توجه:

مدار مربوط به پایه های *AREF* و *AVCC* را حتماً ببندید .



3– بعد از این مرحله باید نحوه ی مختصات بندی و قرارگیری *LCD* را مشخص کنید. در صورتی که قصد دارید *LCD* را به صورت عمودی استفاده کنید تعریف زیر را در برنامه خود و قبل از اضافه کردن فایل سرآمد قرار دهید .

```
#define PORTRAIT
```

که قصد دارید *LCD* را به صورت افقی استفاده کنید دستور زیر را را به جای قبلی قرار دهید .

```
#define LANDSCAPE
```

4– حال باید فایل سر امد توابع راه انداز *LCD* را به برنامه اضافه کنید .

```
#include <tftlcd_functions.h>
```

حالا که تعاریف اولیه برنامه را انجام دادیم می بایست LCD را مقدار دهی اولیه کنیم .برای این منظور باید دستور زیر را در ابتدای تابع main و قبل از دیگر توابع LCD قرار دهیم . lcd_init ();

توابع

حال به تشریح توابع موجود در این کتابخانه می پردازیم . این توابع سه دسته اند :

1:توابع پایه که می توان به وسیله آن توابه قسمت دوم و سوم را بوجود اورد .

2:توابع مربوط به رسم اشکال هندسی .

3:توابع مربوط به چاپ کاراکتر و رشته .

دسته اول توابع :

تابع lcd_write_index_register( ):

این تابع عملیات نوشتن بر روی رجیستر دستور العمل را انجام می دهد .(دستور را به LCD ارسال می کند) و الگوی آن به صورت زیر است.آرگومان ارسالی آن همان کد دستوری است که قرار است به LCD ارسال شود .

```
void lcd_write_index_register(char ins)
```

تابع lcd_write_wdr( ):

این تابع عملیات نوشتن درون رجیستر wdr را برعهده دارد .داده هایی که درون این رجیستر نوشته می شوند به طور معمول به حافظه ی RAM صفحه نمایش ارسال می شود . این تابع یک پارامتر ارسالی دارد که قراراست درون رجیستر wdr نوشته شود . با الگوی زیر :

```
void lcd_write_wdr(int data)
```

تابع lcd_read_rdr( ):

این تابع وظیفه خواندن رجیستر حاوی اطلاعات موجود روی صفحه ی نمایش LCD را بر عهده دارد .پارامتر بازگشتی این تابع همان محتویات موجود در رجیستر rdr ماژول LCDاست و الگوی این تابع به صورت زیر است .

```
int lcd_read_rdr(void)
```

دسته دوم توابع:

تابع lcd_clear_screen( ):

این تابع کل صفحه نمایش را پاک کرده مکان نمای مجازی را به مختصات (0,0) انتقال می دهد والگوی این تابع به صورت زیر می باشد :

void lcd_clear_screen(void)

تابع lcd_background_color( ):

این تابع رنگ پس زمینه را عوض می کند ودارای یک آرگومان ورودی است که همان عدد رنگ پس زمینه است.و الگوی کلی آن به صورت زیر است :

void lcd_background_color(int color)

تابع lcd_write_pixel( ):

با استفاده ازاین تابع می توانیم یک پیکسل از LCDرا با رنگ دلخواه روشن نماییم .این تابع سه آرگومان ورودی دارد .اولی مختصات x و دومی مختصات y و سومی رنگ پیکسل است .الگوی آن به شکل زیر است :

void lcd_write_pixel(unsigned int x,unsigned int y,int color)

تابع lcd_read_pixel( ):

با استفاده از این تابع می توانیم با مشخص کردن مختصات یک نقطه رنگ نقطه مورد نظر را بدست آوریم . این تابع دو آرگومان ورودی دارد که به ترتیب مختصات xوy است و یک آرگومان بازگشتی که رنگ نقطه مورد نظر است و الگوی آن به صورت زیر است

int lcd_read_pixel(int x,int y)

تابع lcd_draw_line( ):

این تابع برای رسم خط به کار می رود و دارای 5 آرگومان ورودی است .*x0 ,y0* که مختصات ابتدای خط اند *y1,x1* که مختصات انتهای خط اند و پارامتر *color* که رنگ خط را مشخص می کند.

void lcd_draw_line(int x0,int y0,int x1,int y1,int color)

تابع lcd_draw_rectangle( ):

این تابع وظیفه رسم یک مستطیل تو خالی *(fill=0)*ویا تو پر *(fill=1)* را دارد . پارامتر های *y0,x0* مختصات گوشه بالا و سمت چپ هستند و*y1,x1* مختصات گوشه پایین وسمت راست هستند . پارامتر *color* نیز رنگ مستطیل را مشخص می کند .

void lcd_draw_rectangle(int x0, int y0, int x1, int y1,char fill,int color)

تابع lcd_draw_circle( ):

این تابع برای رسم یک دایره تو خالی*(fill=0)* ویا تو پر *(fill=1)* به مرکز *y0,x0* و شعاع *radius* به کار می رود وپارامتر color رنگ این دایره را مشخص می کند .

void lcd_draw_circle(int x0, int y0, int radius,char fill,int color)

تابع lcd_write_pic( ) :

این تابع یک عکس که در حافظه *flash* میکرو ذخیره شده است را در مختصات شروع (x,y) نمایش می دهد و الگوی آن به صورت زیر است:

```
void lcd_write_pic(int x, int y,flash unsigned int *pointer)
```

آرگومان سوم این تابع همان عکس است که ما به صورت آرایه در حافظه فلش ذخیره کرده ایم . آرایه ای که ما برای عکس استفاده می کنیم به صورت زیر است و برای اطلاع از چگونگی تبدیل عکس های خود به آرایه به ضمیمه 1 مراجعه کنید .توجه کنید که پارامتر های اول ودوم این آرایه باید طول وعرض عکس مورد نظر باشد .

دسته سوم توابع :

تابع lcd_gotoxy( ):

این تابع مکان نمای مجازی متن را به مختصات *(x,y)* میبرد .باید توجه کنید که در حالت افقی حداکثر مقدار x برابر *40*و y برابر*15* است ودر حالت عمودی حداکثر مقدار *x* برابر *15* و *y*برابر *40* است این تابع با الگوی زیر به کار می رود .

```
void lcd_gotoxy(unsigned char x,unsigned char y)
```

تابع lcd_putchar( ):

این تابع دارای 4 آرگومان ورودی است .اولی همان کارکتری است که در مختصات فعلی مکان نما ی مجازی چاپ می شود . آرگومان دوم رنگ کارکتر و آرگومان سوم برای خاصیت شفاف سازی به کار میرود بدین صورت که اگر این آرگومان برابر صفر(غیر شفاف) باشد رنگ پس زمینه کارکتر همان رنگی خواهد بود که توسط آرگومان چهارم دریافت می شود و اگر این آرگومان برابر یک(شفاف) قرار داده شود رنگ پس زمینه ای کارکتر همان رنگ فعلی پیکسل ها در نظر گرفته می شود .شکل زیر این ویژگی را بهتر معرفی می کند .

```
void lcd_putchar(char character,int forground_color,char background_color_activation,int background_color)
```

تابع lcd_puts( ):

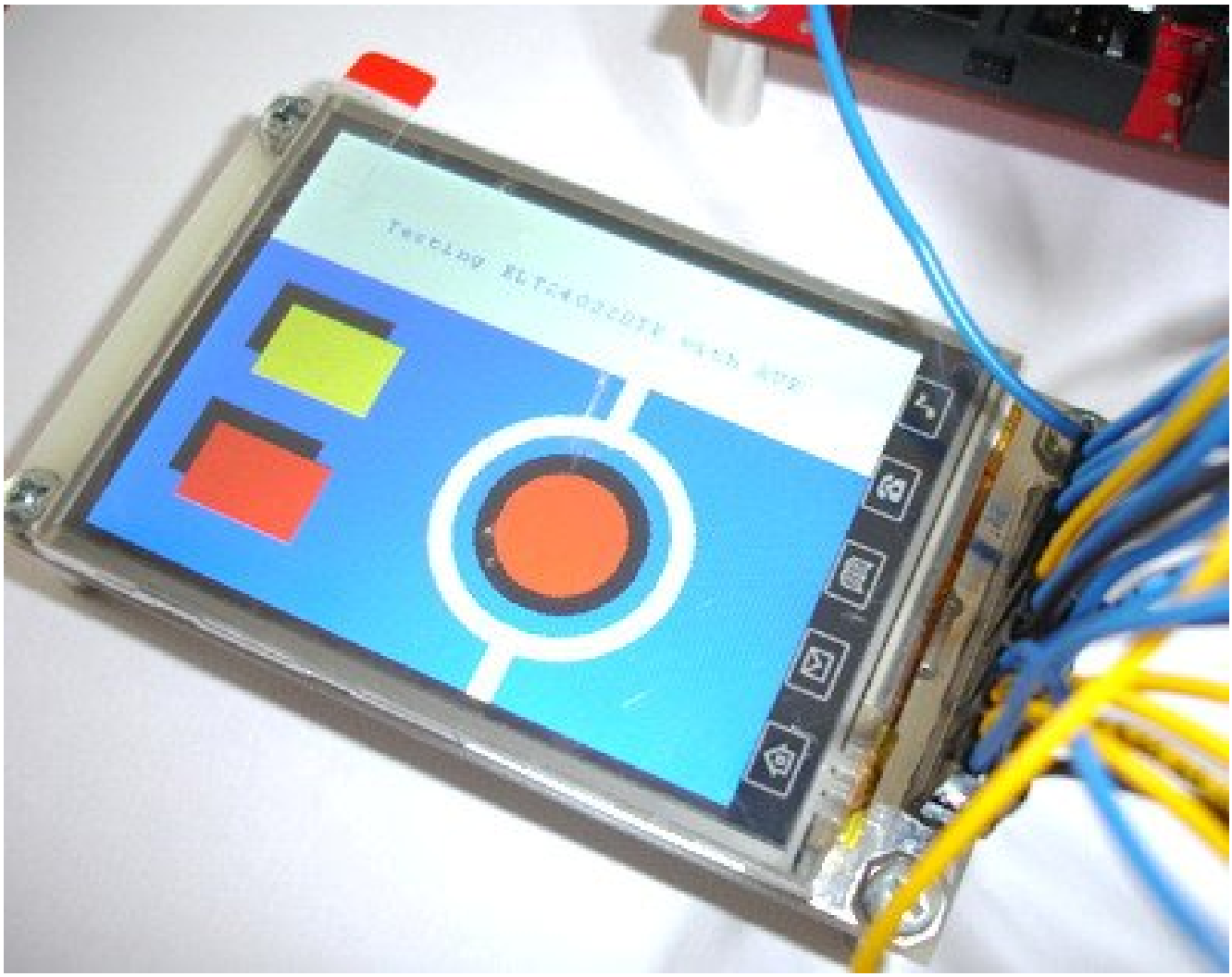
این تابع همانند تابع بالاست با این تفاوت که می تواند یک رشته را که در SRAM میکروکنترلر ذخیره شده را چاپ کند .

```
void lcd_puts(char *string,int forground_color,char background_color_activation,int background_color)
```

تابع lcd_putsf( ):

این تابع نیز همانند تابع بالاست با این تفاوت که رشته ای را که روی حافظه flashمیکروکنترلر ذخیره شده است را می تواند نمایش دهد .

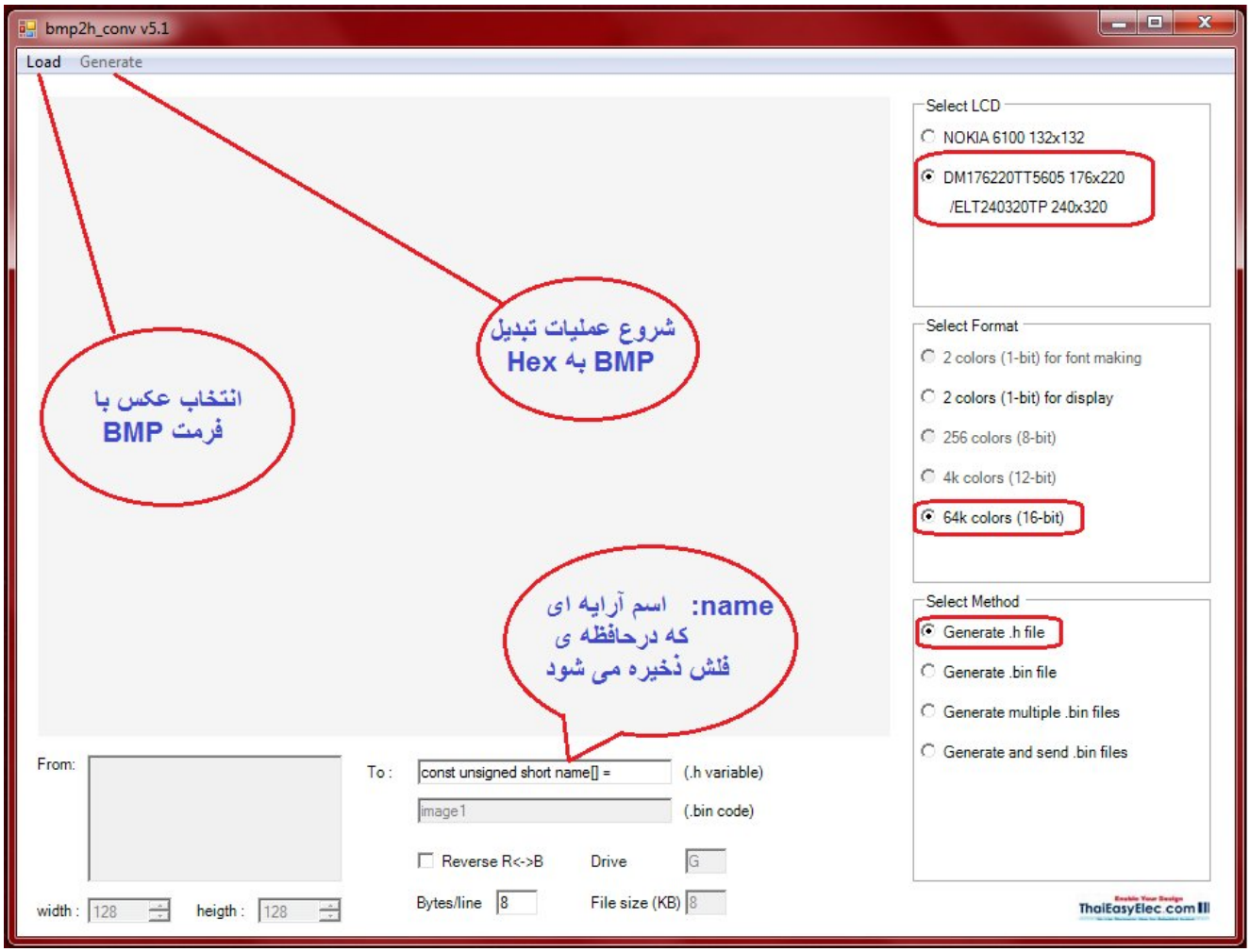
```
void lcd_putsf(flash char *string,int forground_color,char background_color_activation,int background_color)
```



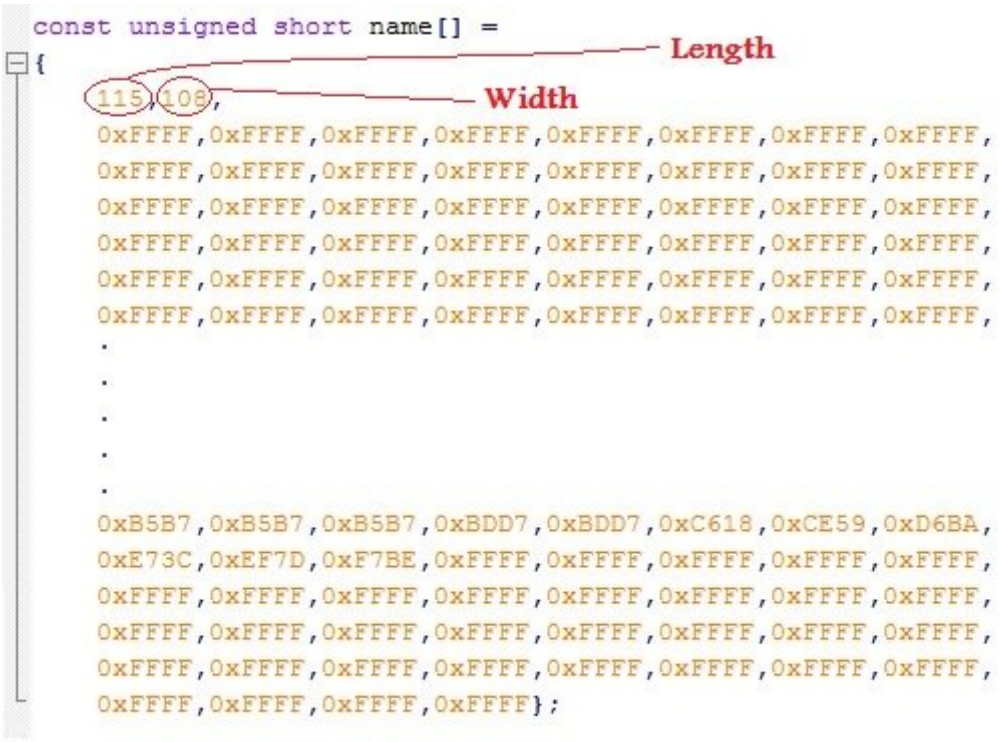
ضمیمه 1

نحوه کار با نرم افزار تبدیل تصاویر با فرمت BMP به آرایه (BMP2h)

برای این منظور ابتدا یک عکس با فرمت BMP که ابعاد آن کمتر از320*240 باشد را توسط کزینه Load بار گذاری می کنیم .سپس تمام گزینه هایی را که در تصویر زیر مشخص شده را انتخاب می کنیم .حال گزینه Generate را زده و از شما آدرسی برای ذخیره ی فایل آرایه می خواهد .مسیر دلخواه را داده و گزینه Save را بزنید .



از انجایی که پارامتر های اول ودوم آرایه می بایست طول و عرض تصویر باشند شما باید این کار را به صورت دستی انجام دهید . برای بدست اوردن طول وعرض تصویر باید هنگامی که نرم افزار را باز میکنید و عکس را بار گذاری کردید در گوشه سمت چپ وپایین Widthو length را به عنوان طول و عرض تصویر به یاد بسپارید .سپس فایل آرایه را باز کرده و با الگوی زیر .Width و length را به آن اضافه کنید .سپس فایل را save کنید و خارج شوید .




```
const unsigned short name[] =
{
    115,108,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    .
    .
    .
    .
    .
    0xB5B7,0xB5B7,0xB5B7,0xBDD7,0xBDD7,0xC618,0xCE59,0xD6BA,
    0xE73C,0xEF7D,0xF7BE,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF};
```

منابع:

1- *www.ECA.ir/forum2*

2- *5th Persian Microcontroller Magazine*