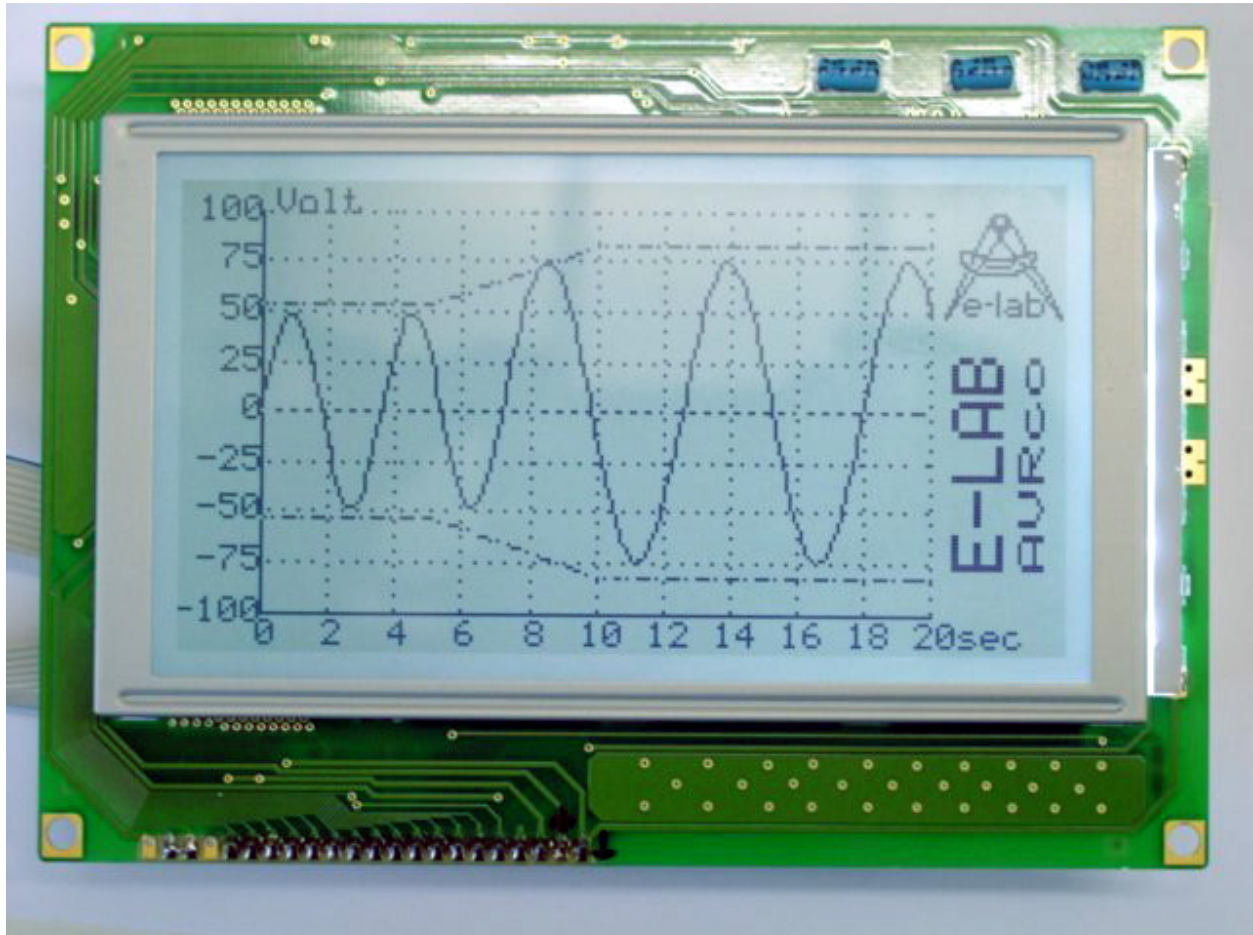


به نام خدا

آموزش میکروکنترلر AVR به زبان C (WinAVR)

(جلسه سوم)



مقدمه:

در جلسه قبل به راه اندازی LCD کاراکتری ۲ در ۱۶ معمولی پرداختیم. در این جلسه تصمیم داریم به معرفی توابع، کتابخانه ها و درایورهای مورد نیاز برای راه اندازی LCD های گرافیکی تک رنگ ۱۲۸ در ۶۴ مبتنی بر چیپ KS0108 و HD61202 که معمولاً با نام تجاری TS12864A-2 شناخته می شود بپردازیم. این کتابخانه با LCD نازک ۱۲۸ در ۶۴ با چیپ SAMSUNG نیز همخوانی داشته و تست شده است.

کد نویسی پروژه راه اندازی LCD گرافیکی 128x64 با کتابخانه `glcd.h` و درایور `KS0108.h`

برای شروع کار یک پوشه به نام `glcd` ایجاد کنید. سپس وارد پوشه `avrlib` شده و فایل‌های `glcd.h` و `glcd.c` و `ks0108.h` و `ks0108.c` و `font5x7.h` و `fontgr.h` و `avrlibdefs.h` و `avrlibtypes.h` و `rprintf.c` و `rprintf.h` و `port128.h` (اگر از میکرووی `mega128` استفاده می‌کنید) از پوشه `avrlib` و فایل‌های `ks0108conf.h` و `global.h` را از پوشه `avr/lib/conf` کپی کنید و در پوشه جدید `glcd` که برای پروژه ایجاد کرده اید `paste` نمایید. این فایلها برای پروژه راه اندازی `GLCD` مورد استفاده قرار می‌گیرند، البته برخی از فایلها عمومی بوده و در اکثر پروژه ها مورد نیاز است.

بعد از کپی فایلها نوبت به ساخت فایل `make` می‌رسد. برای این منظور از منوی `start` پوشه `winavr` و سپس برنامه `mfile` را اجرا نمایید. تمام تنظیمات این برنامه را مطابق جلسه قبل انجام داده و فرکانس `CPU` را روی `8000000` تنظیم کنید و سپس نام فایل اصلی را `main.c` و نوع میکرو را `ATmega16` انتخاب کنید و `Makefile` ایجاد شده را بدون تغییر نام در پوشه `glcd` ذخیره نمایید. سپس همانند شکل زیر `Makefile` را به وسیله `Notepad++` باز کرده و در جلوی خط `SRC` نام فایل‌های `glcd.c` و `ks0108.c` و `rprintf.c` را بنویسید.

```

81
82 # List C source files here. (C dependencies are automatically generated.)
83 SRC = $(TARGET).c glcd.c rprintf.c ks0108.c
84

```

پس از ساخت `Makefile` وارد پوشه `glcd` شده و فایل `ks0108conf.h` را با `Notepad++` باز کنید. همانند جلسه قبل با ویرایش این فایل می‌توانید پایه‌های متصل شده میکرو به `GLCD` را تغییر دهید. در این پروژه دیتا را به پورت `B` و کنترل را به پورت `C` متصل نموده ایم، پس دقت کنید که پورت `JTAG` غیر فعال باشد تا پورت `C` قابل استفاده شود. فایل `ks0108conf.h` را مطابق تصویر زیر ویرایش نمایید:

```

37 // GLCD_PORT_INTERFACE specifics
38 #ifndef GLCD_PORT_INTERFACE
39 // make sure these parameters are not already defined elsewhere
40 #ifndef GLCD_CTRL_PORT
41 #define GLCD_CTRL_PORT PORTC // PORT for LCD control signals
42 #define GLCD_CTRL_DDR DDRC // DDR register of LCD_CTRL_PORT
43 #define GLCD_CTRL_RS PC0 // pin for LCD Register Select
44 #define GLCD_CTRL_RW PC1 // pin for LCD Read/Write
45 #define GLCD_CTRL_E PC2 // pin for LCD Enable
46 #define GLCD_CTRL_CS0 PC3 // pin for LCD Controller 0 Chip Select
47 #define GLCD_CTRL_CS1 PC4 // pin for LCD Controller 1 Chip Select(*)
48 #define GLCD_CTRL_CS2 PC6 // pin for LCD Controller 2 Chip Select(*)
49 #define GLCD_CTRL_CS3 PC7 // pin for LCD Controller 3 Chip Select(*)
50 #define GLCD_CTRL_RESET PC5 // pin for LCD Reset
51 // (*) NOTE: additional controller chip selects are optional and
52 // will be automatically used per each step in 64 pixels of display size
53 // Example: Display with 128 horizontal pixels uses 2 controllers
54 #endif
55 #ifndef GLCD_DATA_PORT
56 #define GLCD_DATA_PORT PORTB // PORT for LCD data signals
57 #define GLCD_DATA_DDR DDRB // DDR register of LCD_DATA_PORT
58 #define GLCD_DATA_PIN PINB // PIN register of LCD_DATA_PORT
59 #endif
60 #endif

```

پس از ویرایش فایلها در پوشه glcd به منوی start و سپس WinAVR رفته و Programmer's Notepad را باز کنید. سپس کدهای زیر را در پنجره پروژه جدید تایپ کرده و برنامه را با نام main.c در پوشه glcd ذخیره و کامپایل نمایید. در صورتی که همه چیز به خوبی پیش رفته باشد فایل hex در پوشه glcd ایجاد می شود که می توانید از طریق کامپایلر Bascom و یا هر نرم افزار دیگری آن را داخل میکروکنترلر پروگرام کنید. توجه داشته باشد که فرکانس میکرو را روی ۸ مگاهرتز RC داخلی تنظیم کنید. همچنین پورت JTAG را غیر فعال نمایید تا پورت C قابل استفاده شود.

```
main.c
#include <avr/io.h>
#include <avr/interrupt.h>
#include "rprintf.h"
#include "glcd.h"

int main (void)
{
    glcdInit();
    rprintfInit(glcdwriteChar);
    rprintfProgStrM("GLCD, AVR64.com");

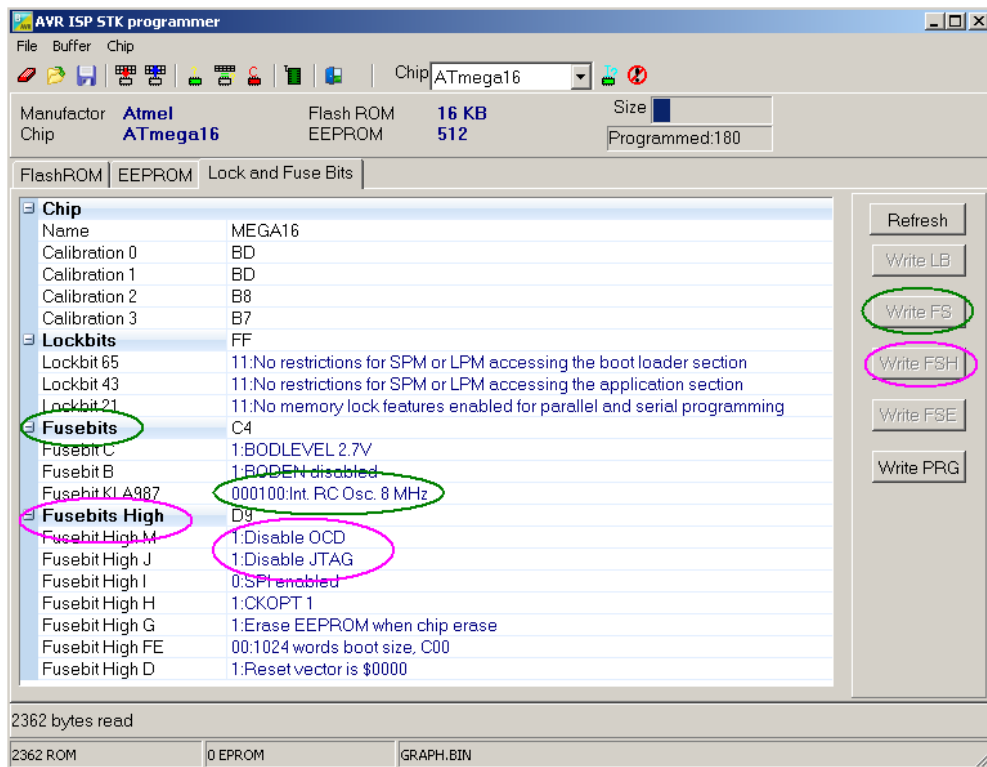
    glcdSetAddress(4,LINE2);
    glcdwriteChar('H');

    glcdSetAddress(4,LINE3);
    rprintfProgStrM("line 3");

    //glcdClearScreen();
    //glcdClearDot(0,0);
    //glcdSetDot(1,1);
    //glcdSetAddress(4,LINE3);
    //glcdPutStr("HD61202/3 controller");

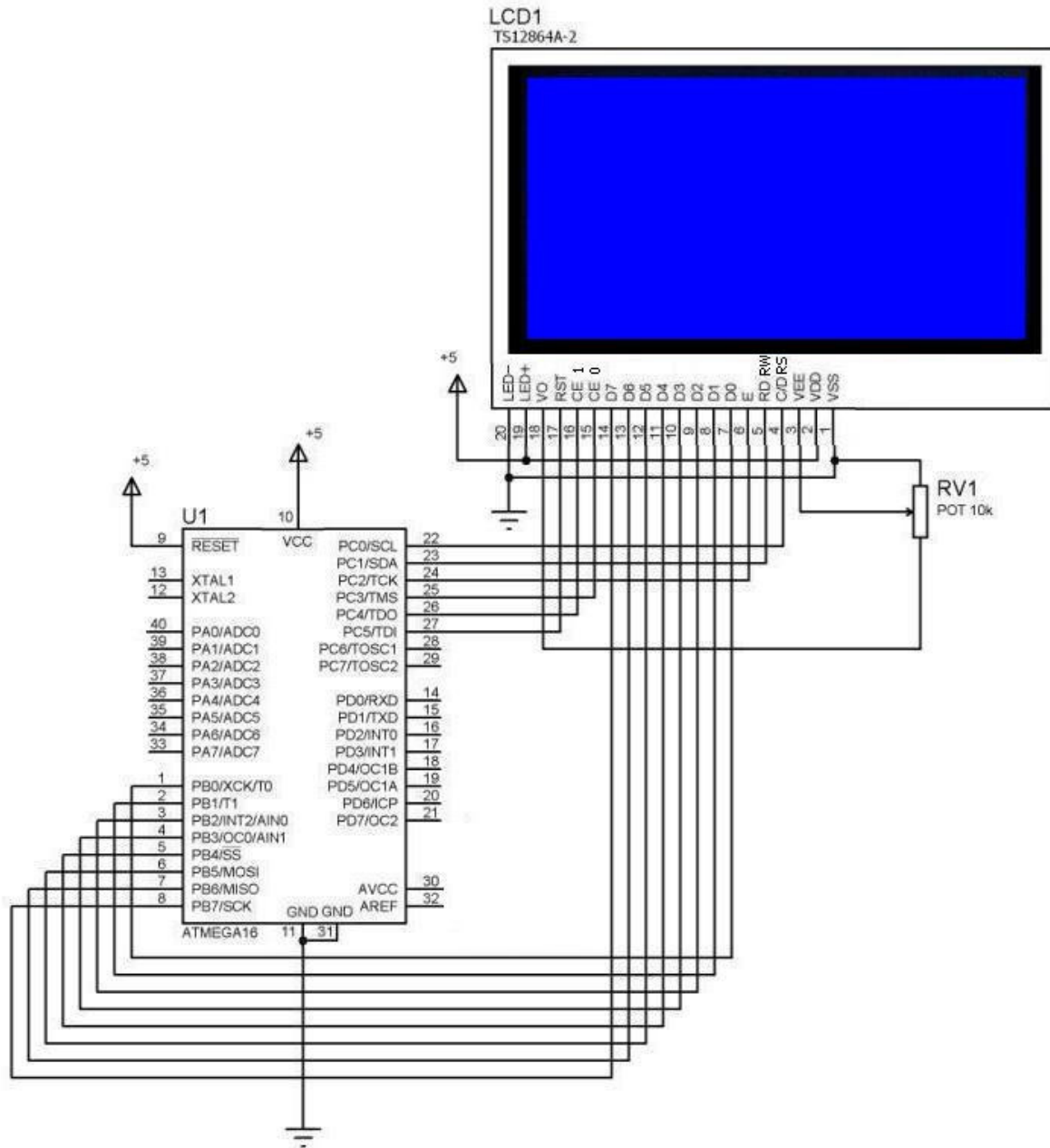
    glcdRectangle(100, 30, 20, 25);
    glcdCircle(30,45,10);
    //glcdStartLine(5);
}
```

برای غیر فعال کردن پورت JTAG و تنظیم فرکانس CPU میکرو روی ۸ مگاهرتز داخلی RC اگر از محیط بیسکام برای پروگرام کردن میکرو استفاده می کنید میتوانید مطابق شکل زیر پس از پروگرام کردن، وارد سربرگ فیوزبیت ها شده و تنظیمات را انجام دهید. پس از انجام تنظیمات هر گروه روی نام آن گروه کلیک کنید و دکمه مربوطه به آن را که فعال خواهد شد کلیک کنید تا فیوز بیت مربوطه در میکرو تنظیم شود. موقع تنظیم فیوزبیت کلاک دقت کنید که سایر منابع رزرو شده را انتخاب نکنید چرا که میکرو بلا استفاده شده و براحتی قابل بازگشت نخواهد بود.



شماتیک پروژه و راه اندازی فیزیکی

برای تست این پروژه مداری را مطابق شکل زیر بر روی بُرد پیاده کنید و فایل هگز را داخل میکرو بارگزاری نمایید.



تصویر پروژه:



تابع `glcd.h` شامل توابع کاربردی بسیار زیادی برای کار با انواع LCD های گرافیکی می باشد. در این پروژه این تابع به همراه درایور `KS0108` استفاده شده است. در واقع `KS0108.h` در سطح سخت افزار عمل کرده و `glch.h` در لایه بالاتر و `rprintf.h` در لایه کاربر فعالیت می کند. کاربری که با دستورات `rprintf` کار می کند انتظار دارد که متن هایش در نمایشگر نشان داده شود. در این پروژه همچنین می توان به طور مستقیم از توابع `glcd.h` نیز استفاده کرد. توابعی مثل تابع رسم دایره و چهار ضلعی از این دسته اند.



برای مشاهده سایر توابع، هدرفایلهای پروژه را بررسی نمایید.

از مزیت های زبان C باز بودن معماری آن است؛ بدین صورت که بر خلاف Bascom و یا سایر کامپایلرهای "بسته" به راحتی می توانید توابع را ویرایش کرده و در صورت نیاز از توابع دیگران در پروژه های خود استفاده نمایید، البته قبل از استفاده از توابع سایرین حتماً به قوانین کپی رایت توجه کنید و لایسنس آن را مطالعه نمایید.

جلسه چهاردهم در این قسمت به پایان می رسد. در ۷ جلسه بعدی که جلسات پایانی WinAVR می باشد به ترتیب به موضوعات زیر خواهیم پرداخت:

- ۴- پورت سریال و توابع مربوطه
- ۵- مبدل آنالوگ به دیجیتال و توابع
- ۶- تایمر / کانتر و توابع کاربردی
- ۷- مموری کارت SD/MMC
- ۸- سیستم فایل و توابع کاربردی
- ۹- مباحث تکمیلی وین ای وی آر
- ۱۰- مثال ها و پروژه های کاربردی

ادامه دارد...

پایان جلسه سوم آموزش AVR

مولف: بهنام زکی زاده

www.avr64.com

۱۵ مرداد ۱۳۹۱

✓ آخرین ویرایش: ۱۴ اسفند ۱۳۹۲