

به نام خدا

آموزش میکروکنترلر AVR به زبان C (WinAVR)

(جلسه دوم)



مقدمه:

در جلسه قبل به معرفی کامپایلر رایگان WinAVR و نوشتن یک برنامه ساده برای روشن و خاموش کردن LED پرداختیم. WinAVR برای پروژه های سنگین تر به تنهایی فاقد کتابخانه های کاربردی می باشد، ولی با این حال به دلیل محبوبیت زیاد زبان C و رایگان بودن کامپایلر مذکور کتابخانه های رایگان بسیار زیادی توسط توسعه دهندگان سراسر دنیا نوشته شده است که مجتمع ترین آنها کتابخانه avr-lib اثر پروفیسور Pascal Stang می باشد. این کتابخانه مشتمل بر صدها هدر فایل و ده ها مثال کاربردی در زمینه ارتباط با LCD های کاراکتری و گرافیکی، مموری کارت های SD/MMC، توابع FAT برای کار با فایلها، ارتباط با پورت سریال و... می باشد. در این جلسه تصمیم داریم با توابع مربوط به LCD کاراکتری معمولی ۲ در ۱۶ کار کنیم.

اولین قدم برای استفاده از توابع کتابخانه avr-lib دانلود آن می باشد. متأسفانه این کتابخانه در حال حاضر نه توسط طراح و نه توسط هیچ وب سایتی پشتیبانی و به روز نمی شود؛ به همین روی آخرین ورژن این کتابخانه که از یک وب سایت شخص ثالث دانلود شده در پوشه این مقاله قرار داده شده است. داخل این پوشه تعدادی فایل c و h قرار گرفته که هدر فایل ها و سورس های اصلی و کاربردی می باشند. همچنین پوشه دیگری به نام examples حاوی چندین مثال کاربردی و نحوه استفاده از هدر فایلها است که در داخل پوشه avr-lib قرار دارد. در این مقاله به نحوه استفاده از هدر فایل lcd.h و سورس lcd.c می پردازیم.

کد نویسی پروژه راه اندازی LCD کاراکتری 16*2 با کتابخانه lcd.h

برای شروع کار یک پوشه به نام lcd ایجاد کنید. سپس وارد پوشه avr-lib شده و فایل های lcd.h و lcd.c و timer.h و timer.c و avr-libtypes.h و avr-libdefs.h و rprintf.c و rprintf.h و port128.h و (اگر از میکرووی mega128 استفاده می کنید) از پوشه avr-lib و فایل های lcdconf.h و global.h را از پوشه avr-lib/conf کپی کنید و در پوشه جدید lcd که برای پروژه ایجاد کرده اید paste نمایید. این فایلها برای پروژه راه اندازی LCD مورد استفاده قرار می گیرند، البته برخی از فایلها عمومی بوده و در اکثر پروژه ها مورد نیاز است.

بعد از کپی فایلها نوبت به ساخت فایل make می رسد. 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 نام فایل اصلی را main.c و نوع میکرو را ATmega16 انتخاب کنید و Makefile ایجاد شده را بدون تغییر نام در پوشه lcd ذخیره نمایید. سپس همانند شکل زیر Makefile را به وسیله Notepad++ باز کرده و در جلوی خط SRC نام فایل های lcd.c و timer.c و rprintf.c را بنویسید

```

80
81
82 # List C source files here. (C dependencies are automatically generated.)
83 SRC = $(TARGET).c lcd.c timer.c rprintf.c
84

```

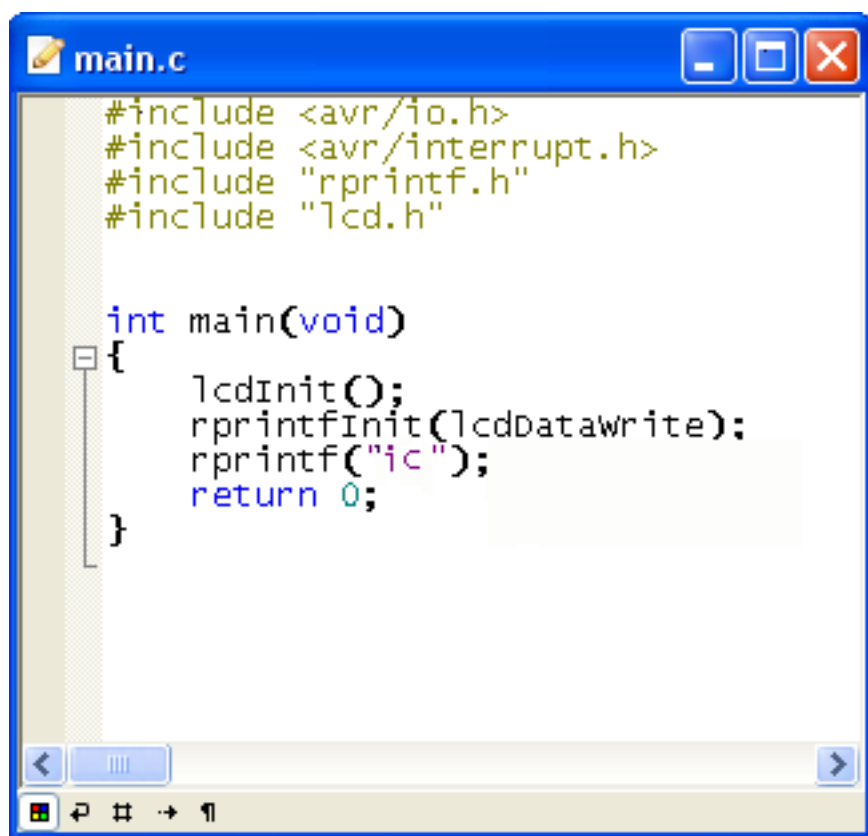
پس از ساخت Makefile وارد پوشه lcd شده و فایل lcdconf.h را با Notepad++ باز کنید و به خطوط 51 تا 61 نگاه کنید. به صورت پیش فرض بایستی LCD به پورت C و A میکرو متصل شود که با ویرایش این خطوط می توانید پورت مورد نظر را تغییر دهید. (۸ خط دیتا به پورت A و ۳ خط کنترل به پورت C). برای این پروژه از پورت D به عنوان کنترل و از پورت B به عنوان دیتا استفاده می کنیم و فایل را به صورت زیر ویرایش می نمایم:

```

50         // port and pins you will use for control lines
51         #define LCD_CTRL_PORT   PORTD
52         #define LCD_CTRL_DDR    DDRD
53         #define LCD_CTRL_RS     2
54         #define LCD_CTRL_RW     3
55         #define LCD_CTRL_E      4
56     #endif
57     #ifndef LCD_DATA_POUT
58         // port you will use for data lines
59         #define LCD_DATA_POUT    PORTB
60         #define LCD_DATA_PIN     PINB
61         #define LCD_DATA_DDR    DDRB

```

پس از ویرایش فایلها در پوشه lcd به منوی start 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 lcd ذخیره و کامپایل نمایید. در صورتی که همه چیز به خوبی پیش رفته باشد فایل hex در پوشه lcd ایجاد می شود که می توانید از طریق کامپایلر Bascom و یا هر نرم افزار دیگری آن را داخل میکروکنترلر پروگرام کنید.



```

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

int main(void)
{
    lcdInit();
    rprintfInit(lcdDatawrite);
    rprintf("iC");
    return 0;
}

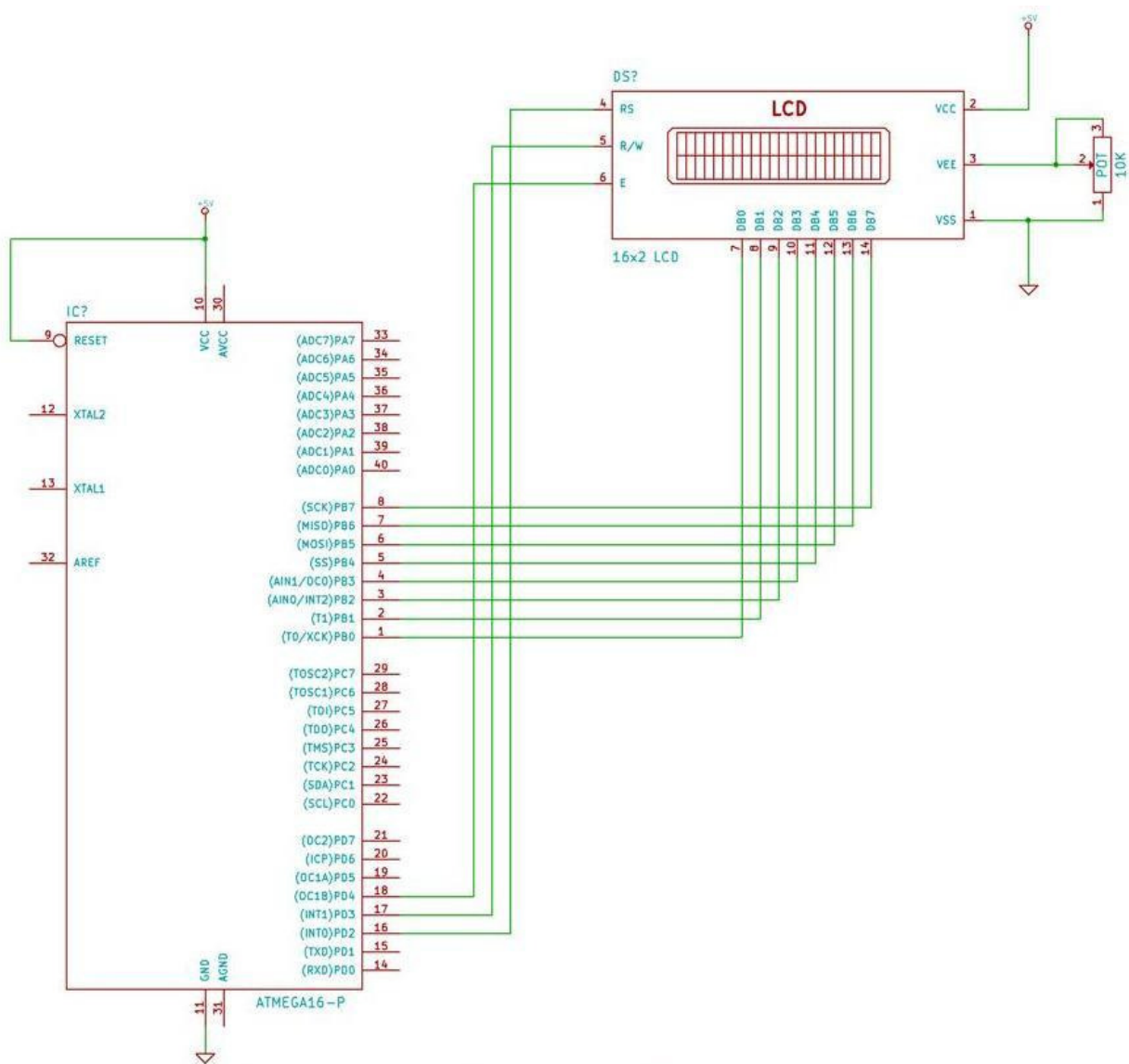
```

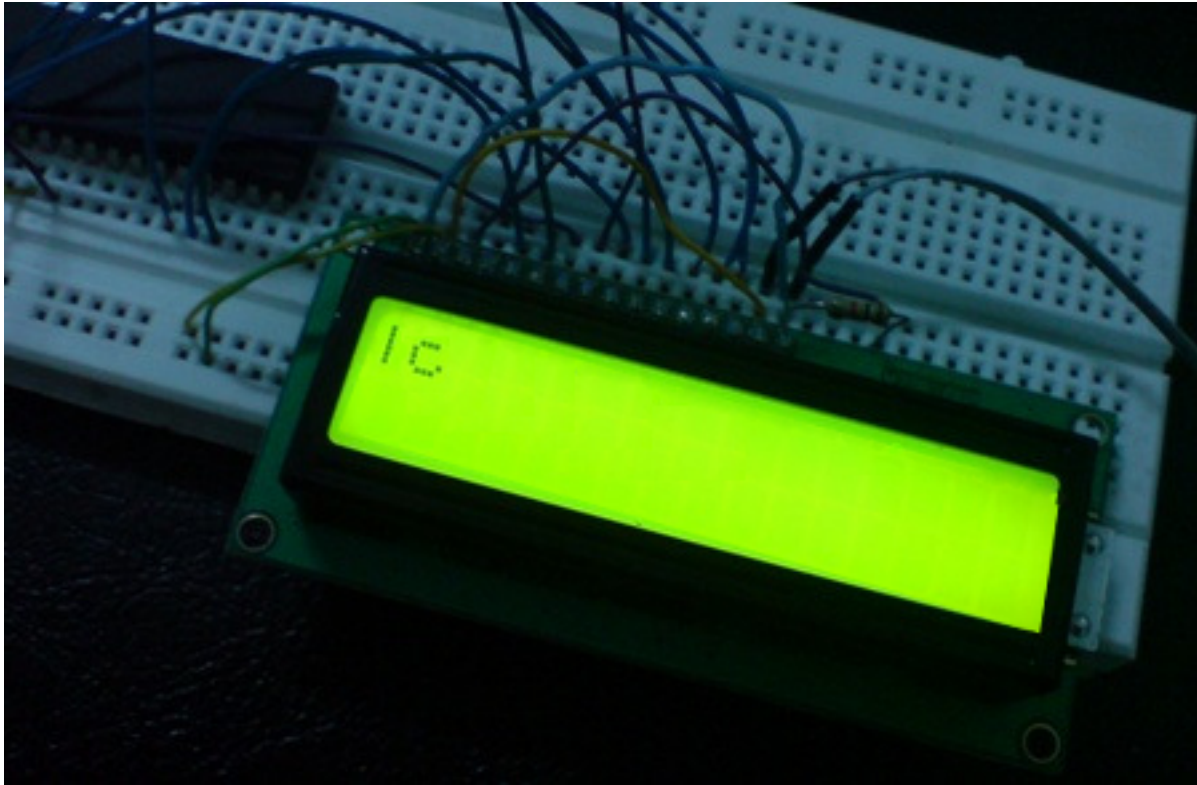
نکته: چرا باید تمام فایل‌های با پسوند c را در خط SRC فایل Makefile بنویسیم ولی فقط بعضی از h فایل ها را در داخل پروژه include می کنیم؟

پاسخ: تمام فایل‌های با پسوند c توسط کامپایلر مورد استفاده قرار می‌گیرند و باید افزوده شوند ولی تنها هدر فایل‌هایی را به پروژه اینکلود می‌کنیم که حداقل یک تابع از آن هدر فایل را در پروژه استفاده کرده باشیم. توجه داشته باشید که وقتی از توابع فایل lcd استفاده می‌کنیم فایل lcd.c نیز کامپایل می‌شود و در این فایل می‌بینیم که timer.h فراخوانی شده پس باید timer.c را نیز در داخل Makefile بنویسیم. برای جلوگیری از پیش آمدن مشکلاتی از این قبیل و مشکل بودن بررسی نیازهای توابع استفاده شده، تمام سورس فایل‌ها را در بخش SRC فایل Make می‌افزاییم تا موقع کامپایل با هیچ مشکلی مواجه نشویم.

شماتیک پروژه و راه اندازی فیزیکی

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





فایل lcd.h شامل توابع زیادی برای کار با LCD می باشد. توابعی از قبیل پاک کردن نمایشگر، بردن مکان نما به موقعیت خاص، Progress Bar برای نمایش پیشرفت روند برنامه یا Loading دستگاه و... این فایل را با Notepad باز کنید و توابع آن را بررسی نمایید. ما در این برنامه ورودی LCD را برای توابع rpintf تعریف کرده ایم. می توان به طور مستقیم نیز در LCD نوشت. در هر صورت با بررسی مثال های کتابخانه avrlib می توانید تجربیات بیشتری در زمینه چگونگی استفاده از این توابع کسب کنید. در جلسه بعدی با LCD گرافیکی 128 در 64 کار می کنیم و با کتابخانه های مورد نیاز برای راه اندازی آن آشنا می شویم.
(TS12864A2)

ادامه دارد...

