

بسم الله الرحمن الرحيم

راه اندازی کتابخانه ff.h

+ CodeVision AVR 3.12 Advanced



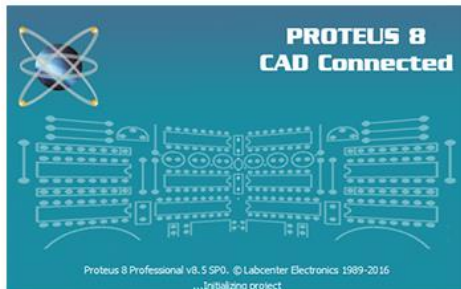
CodeVisionAVR

Integrated Development Environment, C Compiler,
Automatic Program Generator and In-System
Programmer for the Atmel AVR Microcontrollers

Version: 3.12 Advanced

© Copyright 1998-2014
Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfootech.com>

[Credits](#)



شناسنامه

نویسنده: محمّد امین حردانی

نشانی ایمیل: _____

شناسه تلگرام: _____

محل انتشار: گروه تلگرام چیپکده - کانال تلگرام <https://t.me/najafielectronic>

سایت چیپکده: <http://chipkadeh.ir>

تاریخ انتشار: ۱۳۹۶/۰۶/۰۶

شماره نگارش: ۰/۱/۰

فهرست

۱ مقدمه

۱ درباره‌ی این فایل راهنما

۱ درباره‌ی کتابخانه‌ها

۱ درباره‌ی نرم‌افزار و سخت‌افزار

۳ مثال

۳ توضیحات

۱۰ کد

۱۲ کلام پایانی

۱۲ ضمیمه‌ها

۱۲ منابع

مقدمه

درباره‌ی این فایل راهنما

این فایل راهنما فقط برای آشنایی اولیه و راه اندازی کتابخانه‌ی ff است و اصلاً به عنوان یک راهنمای کامل قابل استفاده نیست. برای اطلاعات بیشتر باید به راهنمای کتابخانه که در سایت elm-chang.org مراجعه کنید.

درباره‌ی کتابخانه‌ها

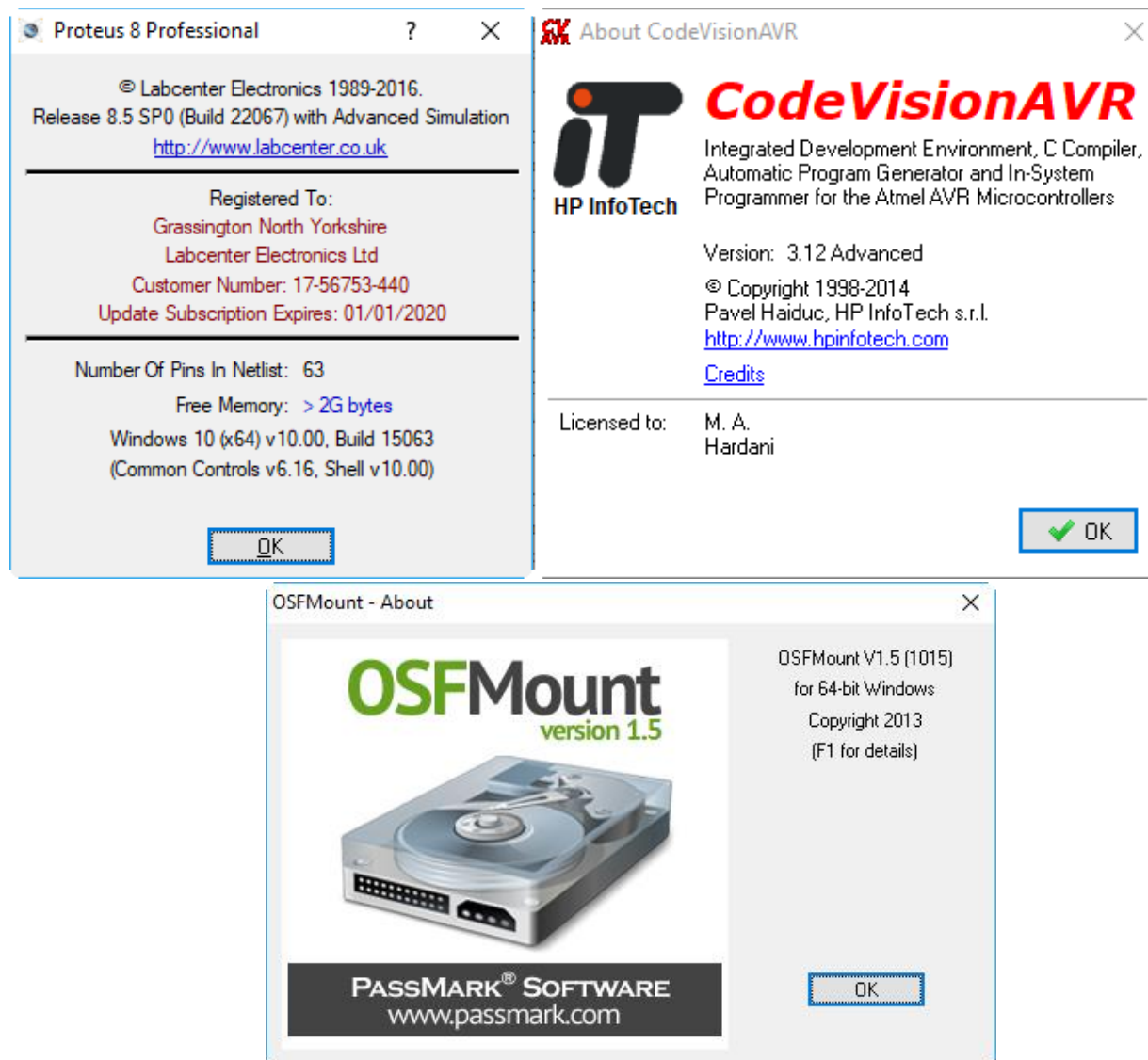
این سایت منتشر کننده‌ی دو کتابخانه‌ی ff و pff برای کار با سیستم فایل FAT است. کتابخانه‌ی ff برای میکروکنترلرهایی مانند ATmega که حافظه‌ی کافی دارند یا میکروکنترلرهای قوی‌تر ساخته شده در حالیکه کتابخانه pff با کمی امکانات کمتر برای استفاده روی میکروکنترلرهای با حافظه‌ی کمتر مانند ATTiny ساخته شده است. با توجه به مقدار حافظه‌ای که نیاز داریم یکی از این دو کتابخانه را استفاده می‌کنیم.

درباره‌ی نرم افزار و سخت افزار



سیستم عامل مورد استفاده Windows 10 x64 Version 1703 Build 15063.540 است. مثال مطرح شده توسط برنامه‌ی CodeVisionAVR 3.12 Advanced نوشته و کامپایل شده است. برای شبیه سازی نیز از نرم افزارهای Proteus 8.5 SP0 Build 22067 برای شبیه سازی منطق برنامه و دیدن خروجی و برنامه‌ی OSFMount 1.5 Build 1015 for 64-Bit Windows فضای شبیه سازی شده‌ی فایل سیستم FAT و اختصاص فضای شبیه سازی خام برای برنامه‌ی Proteus استفاده شده است.

مثال مطرح شده فقط نرم افزاری شبیه سازی شده و تاکنون روی سخت افزار پیاده سازی نشده است!



پیوند دریافت آخرین نسخه CodeVision AVR

<https://soft98.ir/software/programming/3563-CodeVisionAVR.html>

https://www.softgozar.com/codevisionavr-advanced_/

پیوند دریافت آخرین نسخه Proteus

<https://soft98.ir/software/engineering/3535-proteus.html>

https://www.softgozar.com/proteus_/

پیوند دریافت آخرین نسخه OSFMount (کاملاً رایگان)

<https://www.osforensics.com/tools/mount-disk-images.html>

مثال

توضیحات

برنامه از ۴ قسمت تشکیل شده است که در ادامه معرفی می‌شوند:

- ۱- کتابخانه‌های مورد استفاده معرفی شده‌اند تا از توابع موجود در آن‌ها در برنامه استفاده کنیم.
- ۲- متغیرها و ساختارهای مورد استفاده که سعی شده نام‌گذاری آن‌ها متناسب با کارکردشان باشد.
- ۳- وقفه‌ی تایمر که در بازه‌های زمانی کوتاه اقدام به فراخوانی تابع `disk_timeproc()` می‌کند تا MemoryCard به حالت خواب نرود.
- ۴- باقی کدها برای اجرای برنامه

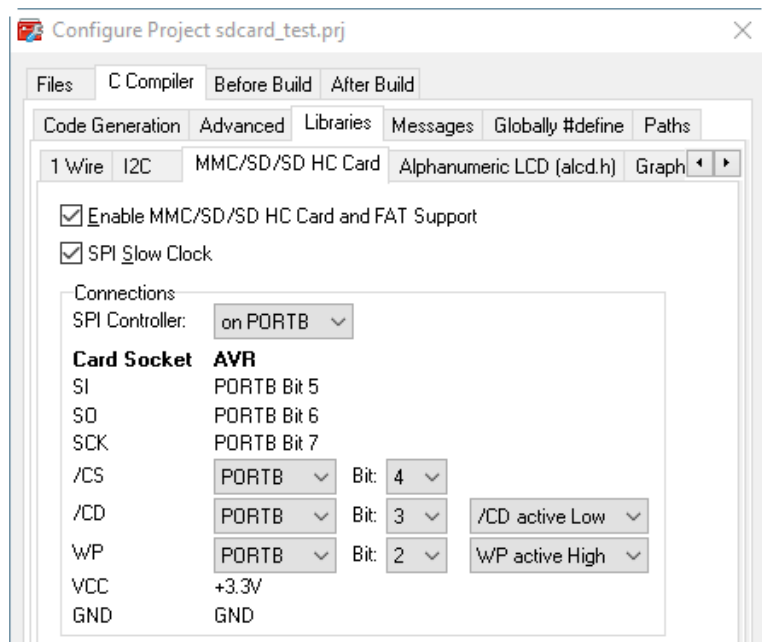
```

D:\Library\Desktop\Test_Projects\SD_Card\CodeVision\sdcard_test.c
Notes sdcard_test.c
19 #include <mega32a.h>
20 #include <alcd.h>
21 #include <ff.h>
22 #include <delay.h>
23
24 unsigned char buffer[16];
25 FRESULT res;
26 unsigned int nbytes;
27 FATFS drive;
28 FIL file;
29 char path[] = "0:/url.txt";
30 char text[] = "http://chipkadeh.ir";
31
32 // Timer 0 overflow interrupt service routine
33 // disk_timerproc avoids MemoryCard sleep!
34 interrupt [TIM0_OVF] void timer0_ovf_isr(void)
35 {
36     TCNT0=0x31;
37     disk_timerproc();
38 }
39
40 void main(void)
41 {
42     lcd_init(16);
43
44     // Timer/Counter 0 initialization
45     // Clock source: System Clock
  
```

تنظیمات تایمر که هم به این شکل انجام شده‌اند.

یادتان باشد این تنظیمات برای شبیه‌سازی انجام شده‌اند و برای واقعیت باید از روی راهنمای کتابخانه محاسبه را انجام دهید. زمان اجرای تابع `disk_timeproc()` را باید هر ۱۰ میلی‌ثانیه تنظیم کنید تا کارت حافظه فعال بماند.

Timer0	Timer1	Timer2	Watchdog
Clock Source: System Clock			
Clock Value: 3.906 kHz			
Mode: Normal top=0xFF			
Output: Disconnected			
☒ Overflow Interrupt ☐ Compare Match Interrupt			
Timer Value: 31 h			
Compare: 0			



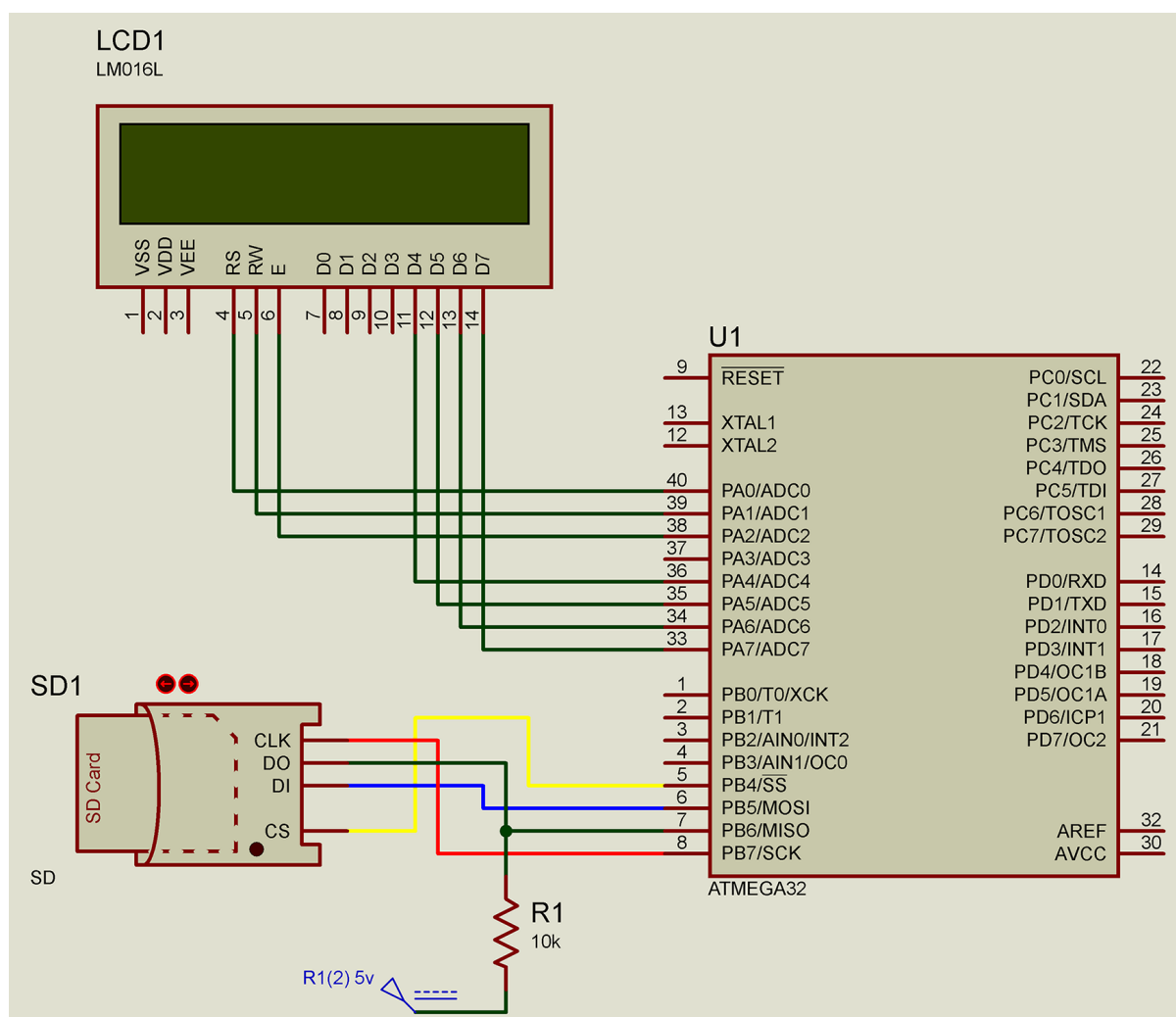
همچنین از منوی

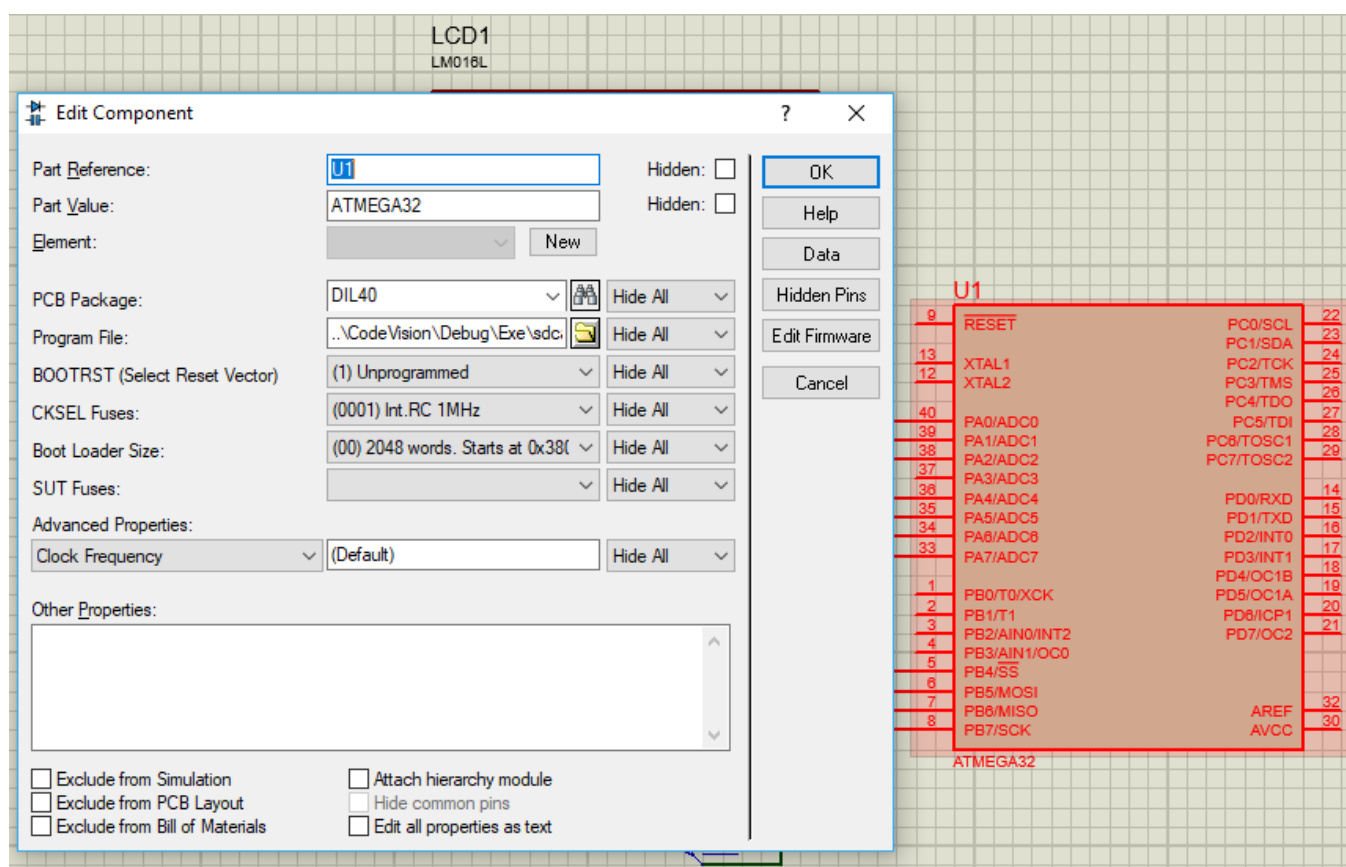
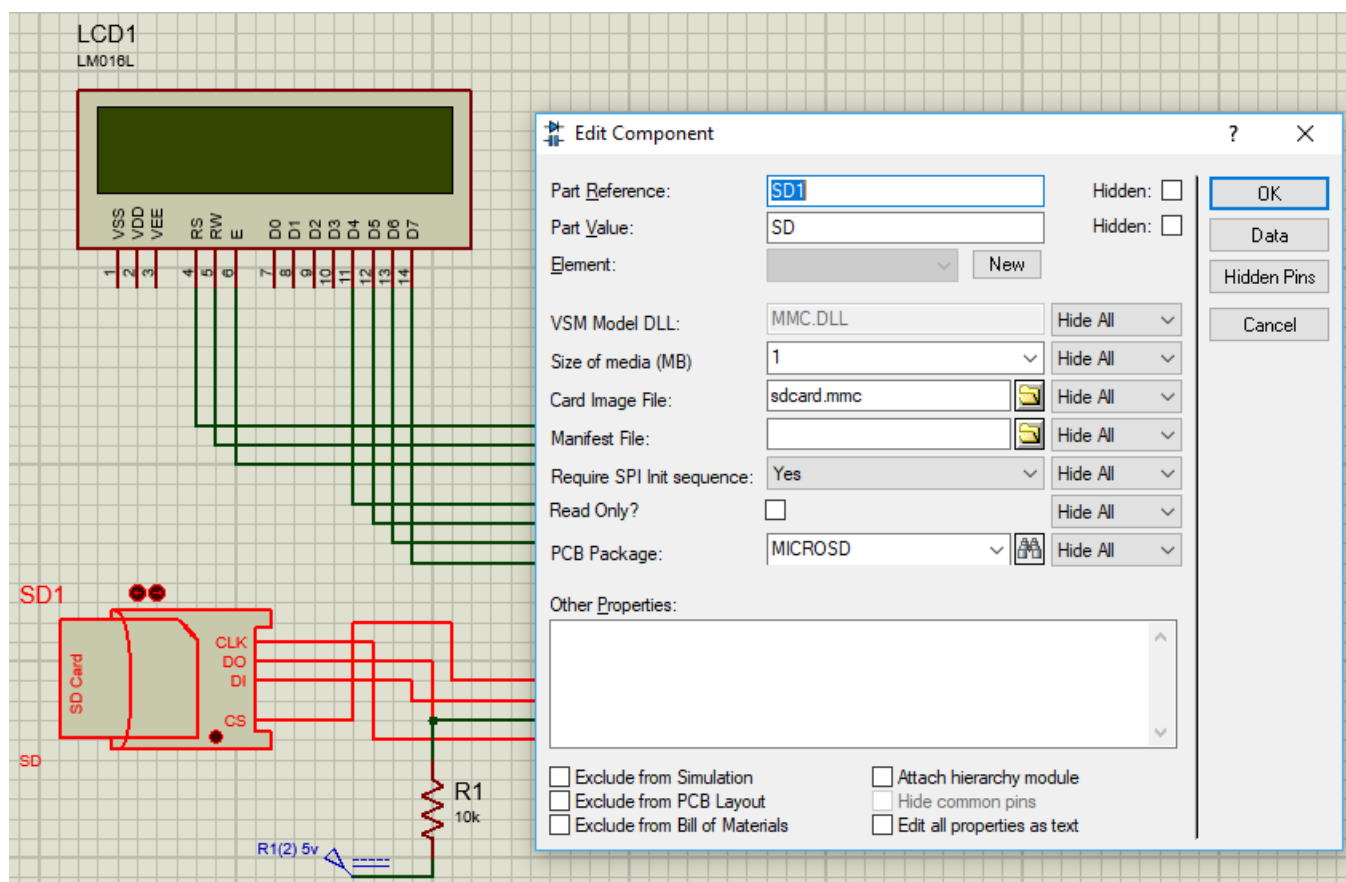
Project → Configure → C Compiler
→ Libraries → MMC/SD/SD HC
Card

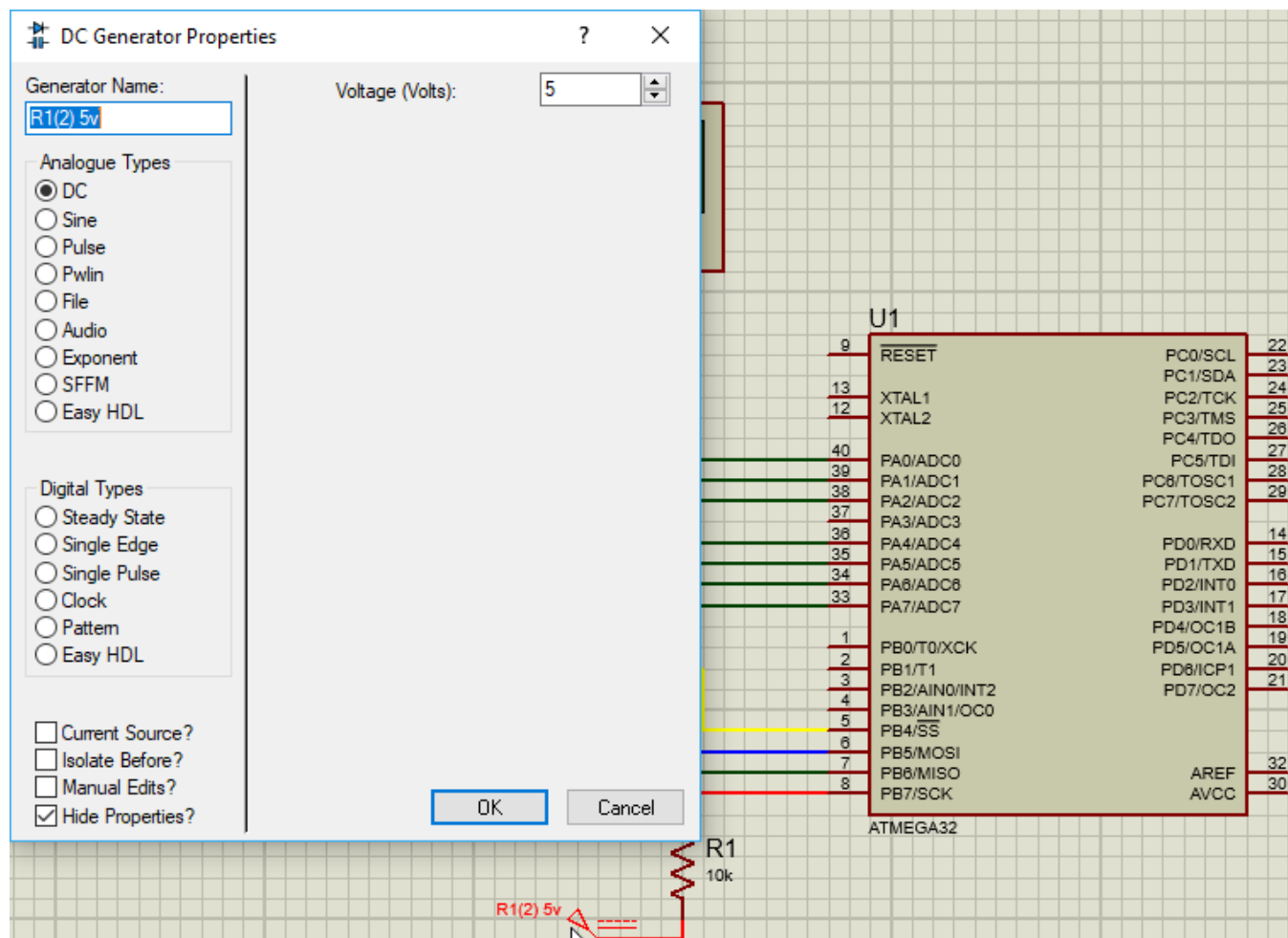
تیک هر دو گزینه‌ی موجود را بزنید و
باقی تنظیمات رو مطابق عکس انجام دهید.

کدهای کامل برنامه هم در ادامه‌ی این
مقاله و هم در ضمیمه آمده‌اند.

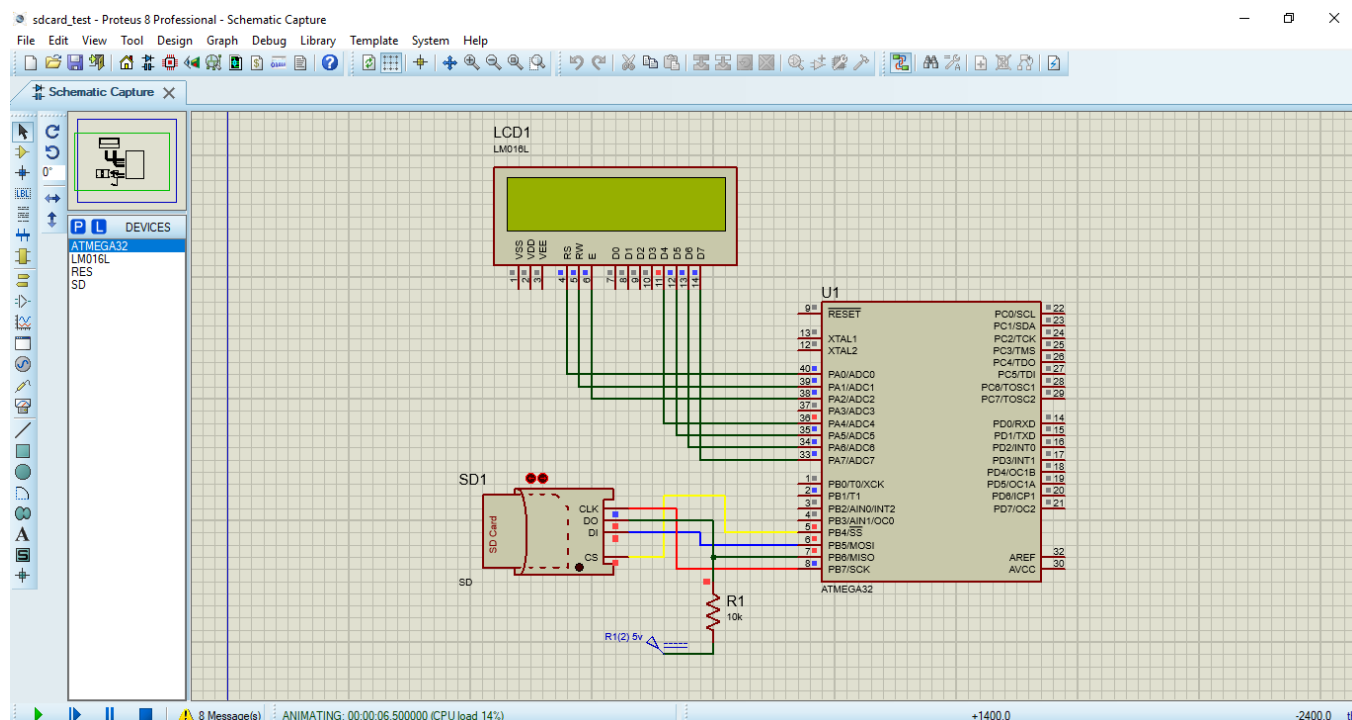
مدار را مطابق شماتیک زیر در پروتئوس ببندید و تنظیمات قطعات را مطابق عکس‌های زیر انجام دهید (حافظه‌ی
مموری کارت 1MB در نظر گرفته شده است تا فایل نهایی پروژه جهت آپلود کم حجم شود):

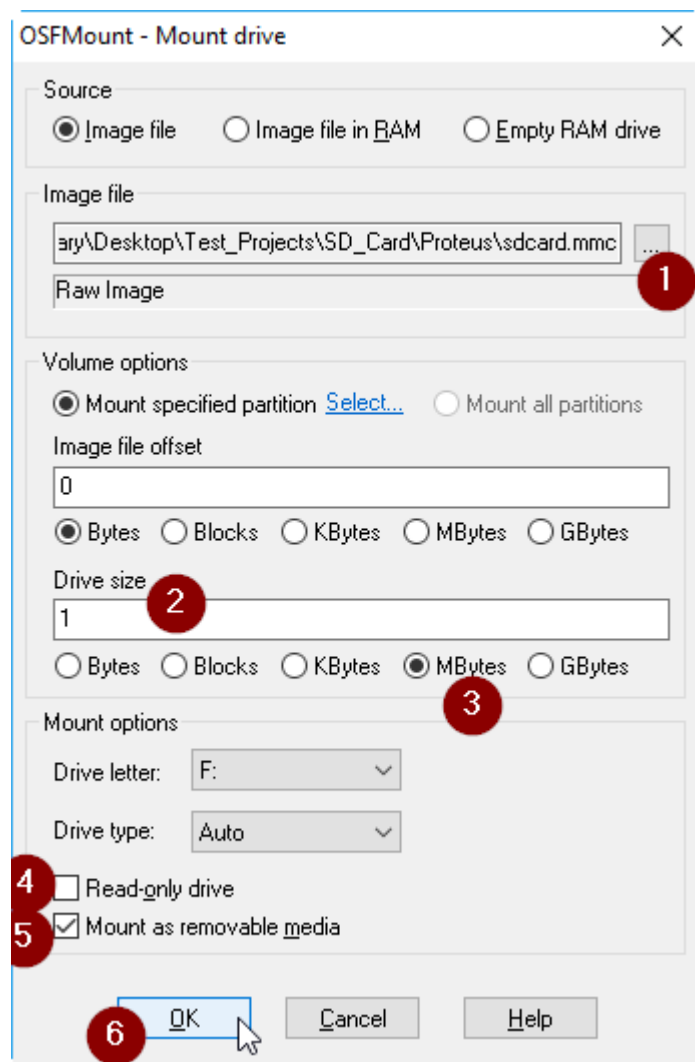






بعد از اولین اجرای شبیه سازی Proteus پیام های خطایی خواهد داد که به دلیل نبودن فایل سیستم است! تصویر زیر را ببینید:

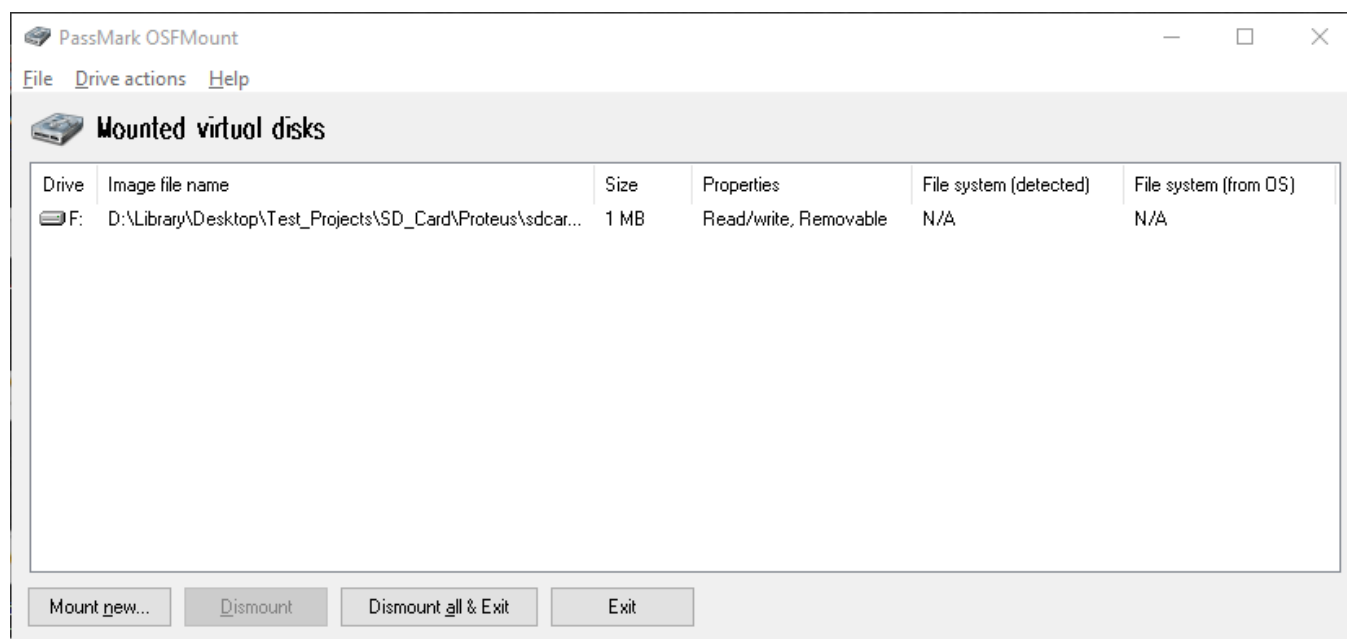




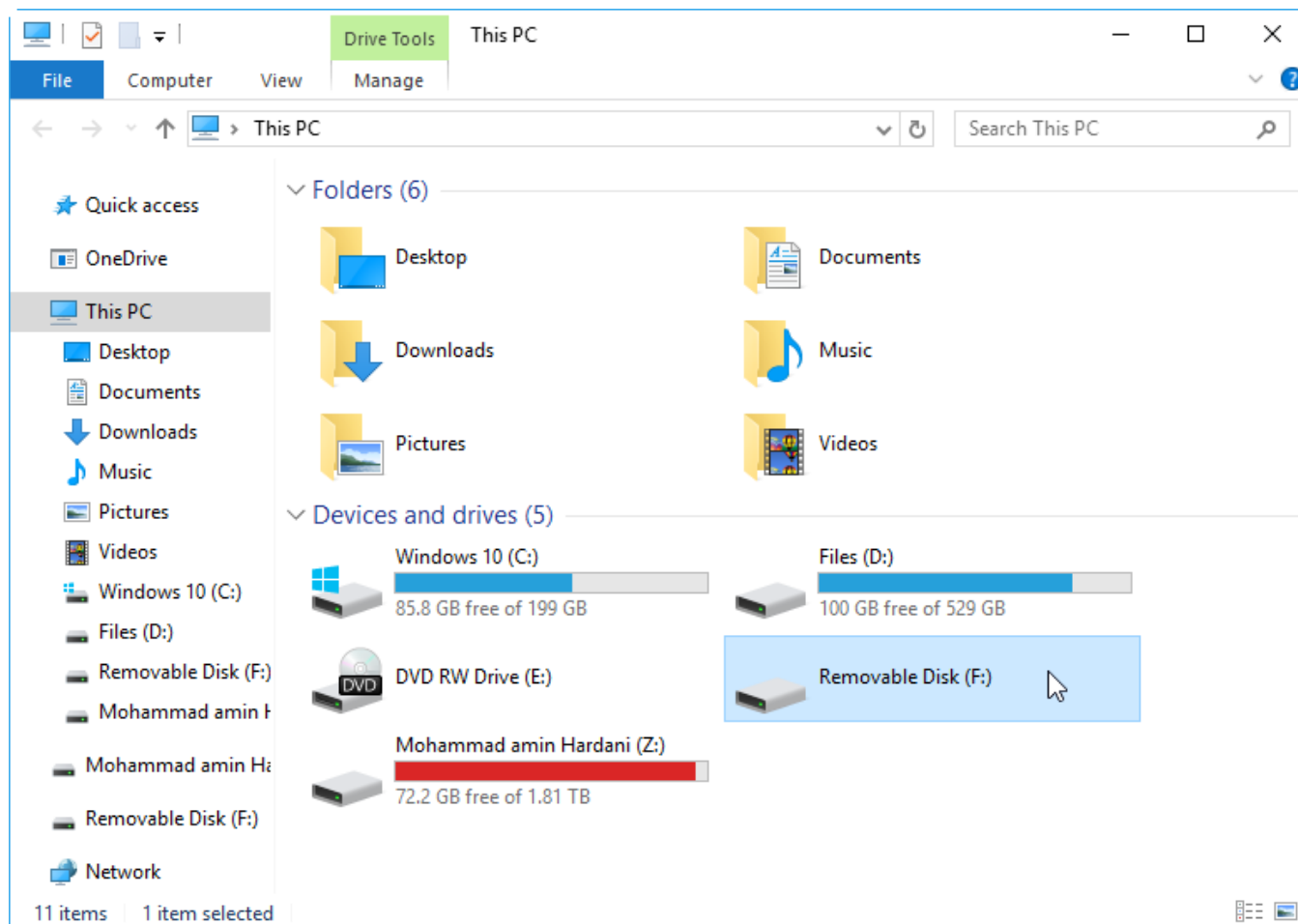
راهکار حل مشکل این است که یک بار با استفاده از برنامه‌ی OSFMount فایل Image کارت حافظه که توسط پروتئوس ایجاد شده است را به ویندوز وصل کنیم تا بتوانیم آن را به فرمت FAT تبدیل کنیم.

ابتدا برنامه‌ی OSFMount را اجرا کنید و سپس از منوی File گزینه‌ی Mount new virtual disk را بزنید تا پنجره‌ی روبرو باز شود. مطابق شکل مقادیر پنجره را تنظیم کنید و روی OK کلیک کنید.

حالا باید فایل ما در ویندوز به شکل زیر متصل شده باشد:

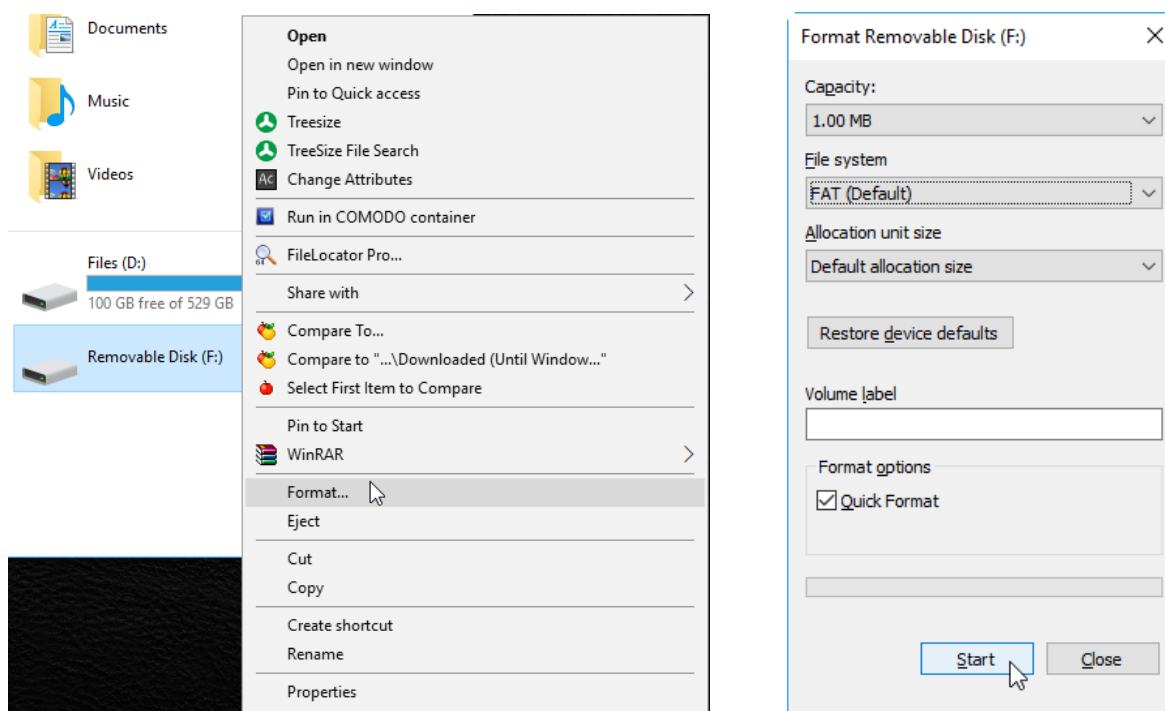


همچنین درایور F نیز به صورت زیر دیده شود:



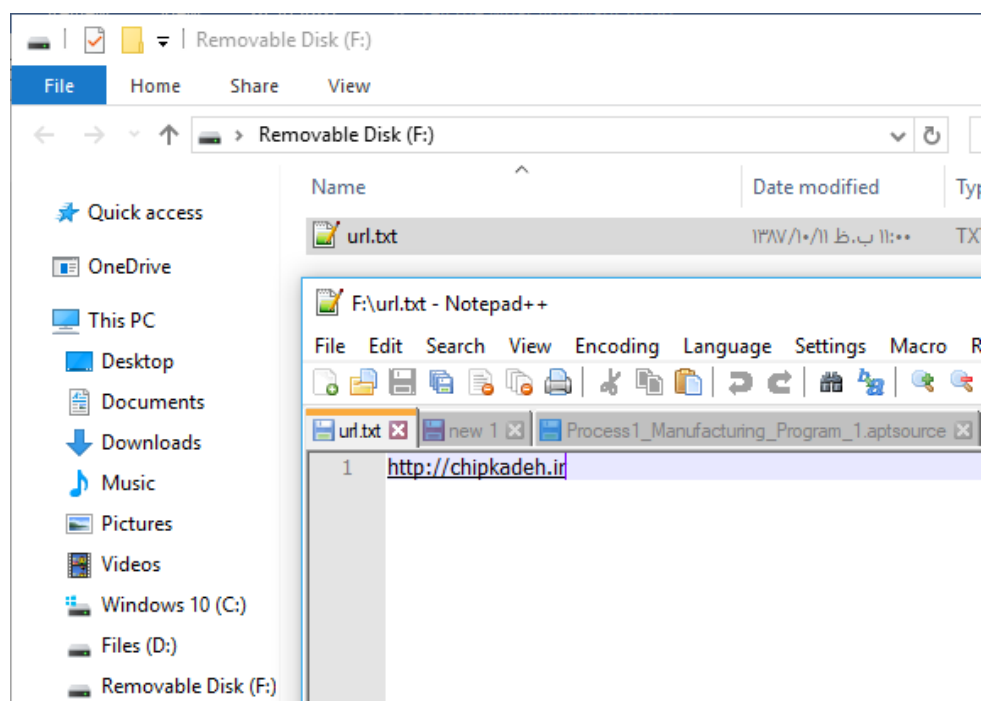
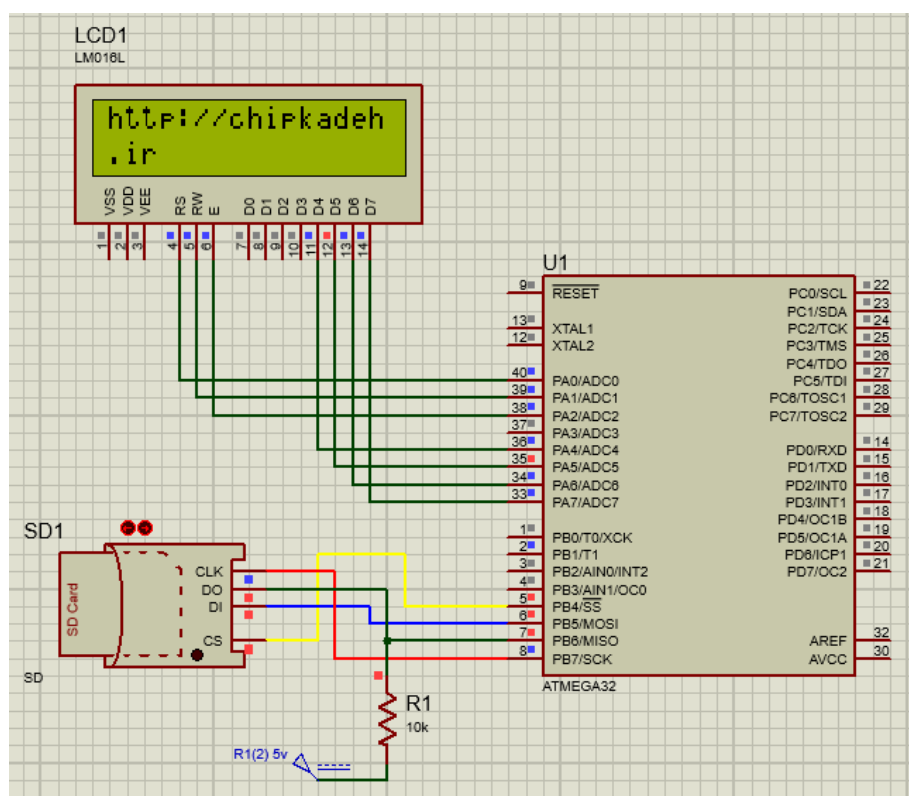
برای استفاده از درایو و تبدیل آن به FAT روی آن کلیک راست کرده و گزینه‌ی Format را انتخاب کنید. سپس تنظیمات را با توجه به شکل انجام دهید و سپس عملیات فرمت را Start کنید. پیام‌های هشدار را هم تأیید

نمایید.



چون حجم درایو را 1MB در نظر گرفتیم، فقط می‌توانیم از FAT12 استفاده کنیم؛ در حالیکه اگر حجم درایو را بیشتر می‌کردیم می‌توانستیم FAT16 و FAT32 رو هم توی گزینه‌ها ببینیم و با آن‌ها فرمت کنیم.

کار ما با درایو مجازی تمام شد! به برنامه OSMount برگردید و روی دکمه‌ی Dismount all & Exit کلیک کنید. با اجرای برنامه‌ی شبیه‌سازی شده در Proteus می‌بینید که ایندفعه برنامه به خوبی کار می‌کند و تا انتها پیش می‌رود. با Mount مجدد فایل SDcard می‌توانیم فایل ایجاد شده در فایل سیستم را هم ببینیم.



```

#include <mega32a.h>
#include <alcd.h>
#include <ff.h>
#include <delay.h>

unsigned char buffer[16];
FRESULT res;
unsigned int nbytes;
FATFS drive;
FIL file;
char path[] = "0:/url.txt";
char text[] = "http://chipkadeh.ir";

// Timer 0 overflow interrupt service routine
// disk_teimerproc avoids MemoryCard sleep!
interrupt[TIM0_OVF] void timer0_ovf_isr(void)
{
    TCNT0 = 0x31;
    disk_timerproc();
}

void main(void)
{
    lcd_init(16);

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: 3.906 kHz
    // Mode: Normal top=0xFF
    // OC0 output: Disconnected
    // Timer Period: 52.992 ms
    TCCR0 = (0 << WGM00) | (0 << COM01) | (0 << COM00) | (0 << WGM01) | (1 << CS02) |
            (0 << CS01) | (0 << CS00);
    TCNT0 = 0x31;
    OCR0 = 0x00;
    // Timer(s)/Counter(s) Interrupt(s) initialization
    TIMSK = (0 << OCIE2) | (0 << TOIE2) | (0 << TICIE1) | (0 << OCIE1A) | (0 << OCIE1B) |
            (0 << TOIE1) | (0 << OCIE0) | (1 << TOIE0);

    #asm("sei")

    DDRB.1 = 1;
    PORTB.1 = 0;

    /* ===== Write to disk ===== */
    f_mount(0, &drive); // mount
    f_open(&file, "url.txt", FA_WRITE | FA_OPEN_ALWAYS | FA_OPEN_EXISTING);
    f_write(&file, text, sizeof(text) - 1, &nbytes); // sizeof()-1 means no "\r"
                                                    // or "\n" character
    f_close(&file);
    f_mount(0, 0); // dismount

    /* ===== Read from disk ===== */
    if ((res = f_mount(0, &drive)) == FR_OK)
        lcd_putsf("FAT mounted = OK");
    delay_ms(1000);

    lcd_clear();
    if ((res = f_open(&file, path, FA_READ)) == FR_OK)
        lcd_putsf("File open: url.txt");
    delay_ms(1000);
}

```

```
lcd_clear();
if ((res = f_read(&file, buffer, 20, &nbytes)) == FR_OK)
    lcd_putsf("Reading Data");
delay_ms(1000);
lcd_clear();
lcd_puts(buffer);
delay_ms(1000);

f_close(&file);
while (1);
}
```

کلام پایانی

پیشاپیش به دلیل خطاهای تایپی احتمالی در این مقاله عذرخواهی می‌کنم. لطفاً در صورت وجود هرگونه ایراد فنی یا تایپی، پیشنهاد یا انتقاد، آن را از طریق راه‌های ارتباطی موجود در صفحه‌ی اول این مقاله انتقال دهید. از آنجا که برای تهیه این مقاله و یافتن و تست کامل نرم‌افزارها زحمت کشیده شده و زمان صرف شده است، استفاده از این مقاله (چه تجاری و چه غیرتجاری) فقط با ذکر منبع مجاز است.

با تشکر
محمد امین حردانی

ضمیمه‌ها

- فایل پروژه CodeVision به صورت کامل
- فایل پروژه شبیه‌سازی شماتیک Proteus به صورت کامل
- برنامه‌ی رایگان OSFMount

منابع

- راهنمای برنامه‌ی CodeVision AVR
- سایت elm-chang.org
- <http://elm-chan.org>
- راه‌اندازی FAT و کار با حافظه‌های MMC و SDC با AVR، نوشته‌ی حمید لطفعلی‌زاده
- <http://ntaco.ir/tag/codevision-راه-اندازی-mmcتنظیمات->