

Cnc Designer

Matematiği seviyoruz

FAT16 - SD kartdaki dosyaları okumak.

Bülent Ünalmiş tarafından 27 Kasım 2013 tarihinde gönderildi

[SD kartı herhangi bir işlemciye bağlayarak sektörlerini okuyabiliyoruz.](#)

Bu yazımda adı geçen CMD16, CMD17 gibi terimler size bir şey ifade etmiyorsa yukarıdaki linki tıklayıp SPI + SD kart yazımı okumanız gerekmektedir.

Şimdi ise, PC ortamında içine dosyalar yerleştirilmiş FAT yapısındaki bir SD kartta dilediğimiz herhangi bir dosyayı nasıl okuruz konusunu ele alacağım.

Bir önceki yazımda 8 GB ve 2 GB iki SD kart aldığımı ve 8GB olanı okurken bazı sorunlar yaşadığımı ve bu sorunların üzerine gitmeyip 2GB olanla yetindiğimden bahsetmiştim.

Başlıyoruz...

Sıfıncı sektördeki 0x1BE adresinden İlk "partition"a ait 16 bytlık tabloyu okuyalım.

İlk okunan data Data0 son okunan data Data15 olsun.

Denemelerimde okuduğum hex değerleri sırayla yazayım:

00 00 20 00 06 7d 7f 0b 20 00 00 00 e0 77 3b 00

Data4 6 dan farklı ise bu SD kart, FAT16 formata sahip değil demektir. Bu durumda uyumsuz SD kart hata mesajı ile kullanıcıyı uyaracağım.

Data8, Data9, Data10, Data11 LSB ... MSB şeklinde dizilmiş 32 bitlik bir sayı olup, sektör numarasını tanımlar. (Data11, 2GB kartlarda hep 0x00 dır.)

Adres = [Data8 + (Data9<<8) + (Data10<<16)] x 512

Bu hesaplanan adres, ulaşmak istediğimiz 0x20 numaralı sektörün ("Bios Parameter Block"u barındıran sektörün) fiziksel adresidir.

Benim elde ettiğim verilerle ulaştığım sonuç 0x00000020 * 0x200 = 0x000004000

Şimdi ne yapacağız? Gidip bu 32 nolu sektörü yani 0x00004000 adresindeki verileri okuyacağız. Aradığımız bilgiler ilk 32 byte içinde yer aldığından 32 byte okuyalım.

CMD16(0x00, 0x00, 0x00, 0x20) parametresi ile 32 byte okuyacağımızı bildirelim. Ardından **CMD17(Adres)**

Bunları alınmış sırasına göre Data0, Data1, Data2.....Data32 olarak isimlendiriyorum.

(Ofset 11) 1 sektordeki byte sayısı Data11 + Data12<<8

(Ofset 14) Rezerv edilmiş sektör sayısı Data14 + Data15<<8

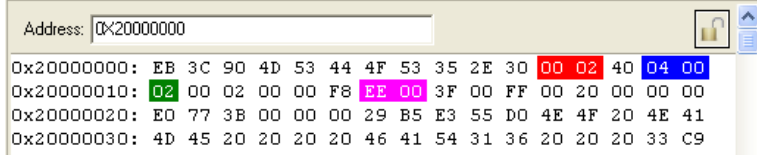
(Ofset 16) FAT sayısı

Data16

(Ofset 22) Fatdeki sektör sayısı

Data22 + Data23<<8

Kendi denemelerimde okuduğum değerler aşağıda görülmektedir. 0x20000000 SD karttan okuduğum verileri yazdığım RAMın adresidir.



Şu anda lazım olmayacak ama Data13=0x40 değerini saklayın.

Adres = Adres + [Data11 + (Data12<<8)] x [Data14 + (Data15<<8)]

Adresde 0x00004000 değeri vardı.

Adres = 0x00004000 + (0x0200) x (0x0004) = 0x00004800 Bu benim FAT adresim.

Yapılan işlem reserve sektörlerin kaç byte olduğunu bulduk ve Adres'e ekledik. Adres şu an bulunduğumuz sektörün (MBR sektörünün) adresiydi. Dolayısı ile şu andaki adres FAT'i işaret etmektedir. (Bu adresi bir yerlerde saklayın ileride tekrardan lazım olacak.)

Adres = Adres + [[Data11+(Data12<<8)] x Data16 x [Data22+(Data23<<8)]]

FAT sayısı ile FAT'deki sektör sayısını çarparsak, FAT'lerin kaç sektörden oluştuğunu bulmuş oluruz. Bunu da bir sektördeki byte sayısı ile çarparsak FAT'lerin kaç byte olduğunu bulmuş oluruz. Bunu da Adres'e eklersek FAT bloğunun arkasındaki Root Directory alanının adresini bulmuş oluruz.

Adres = 0x00004800 + (0x0200) x (2) x (0x00EE) = 0x00040000

Bu da benim Root Directory Adresim.

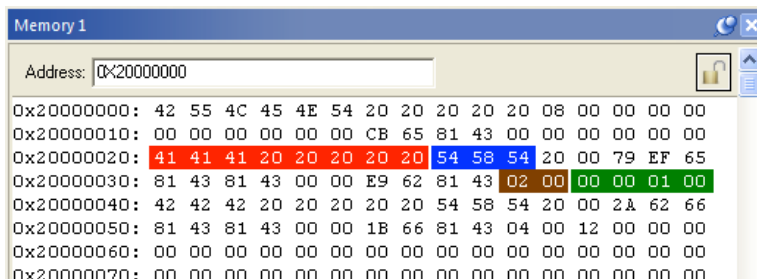
Cevaplanması gereken sorulardan birisi Root Directory kac sektor? Microsoft sadece 16Kbyte yer ayırmış.

Şimdi bu adresten 512 byte okuyayım ve içeriğine bakalım.

Benzer çalışmayı yapmak istiyorsanız (Eğer daha önce SD karta dosya kaydedip sildiyseniz önce formatlayın volume name de verin.) PC'de AAA.TXT adında bir dosya oluşturup içine 65536 Byte uzunluğunda text yerleştirin vs SD karta kopyalayın. Bir de BBB.TXT adında dosya oluşturup içine 0123456789ABCDEF(enter) bilgilerini yerleştirin bunu da SD kartın içine atın.

Bunun için CMD16(0x00000200) ve CMD17(0x00040000) komutlarını işleteceğim.

Komutları işlettiğimde işlemci ramının 0x20000000 adresine ilgili sektör okunmuş oldu.



Bu ulaştığımız sektörde Root Directory altına kopyalanmış dosyaların kimlik bilgileri saklanmaktadır. Her bir dosyaya ait kimlik 32 byte kaplar.

İlk adres 0 adresidir. (0x20000000 yazdığına bakmayın o benim işlemcimin Ram adresi)

Verdiğim "Root Directory" verilerinin bulunduğu resimde 0x00, 0x20, 0x40, 0x60 adreslerinde 4 adet kimlik bilgisi bulunmaktadır.

İlk olarak AAA.TXT dosyamızı okuyalım. AAA dosya adını sonuna gerekiyorsa boşluk karakterleri ekleyerek dosya adını 8 byte uzunluğa çıkartalım. Dosya uzantısını da toplam 3 karakter olacak şekilde uzatalım. Yani AAA TXT

Bu, ASCII veri olduğundan bunun hex karşılığı;

0x41 0x41 0x41 0x20 0x20 0x20 0x20 0x20 0x54 0x58 0x54 olacaktır.

Şimdi bu 11 byte veriyi "Root Directory"den okuduğumuz verilerin içinde aramaya başlayalım. (0x00000000 e bak yoksa 0x00000020 ye bak yoksa 0x00000040 a bak

Hey0000, 0x00000020 de dosyamızın adı var. 0x20000000 okunan dataların RAMdaki adresi olduğuna göre, ofset degerim 0x20000000. Buna göre relatif adresler:

- [0x00 11 byte] Dosya Adı ve uzantısı (8,3) xxxxxxxx.yyy.
- [0x0B 1 byte] Attribute
- [0x0C 1 byte] Rezerv
- [0x0D 3 byte] Dosyanın oluşturulma zamanı
- [0x10 2 byte] Dosyanın oluşturulma tarihi
- [0x12 2 byte] Dosyaya son erişim tarihi
- [0x14 2 byte] Rezerv
- [0x16 2 byte] Dosyada yapılan son değişikliğin saati
- [0x18 2 byte] Dosyada yapılan son değişikliğin tarihi
- [0x1A 2 byte] Dosyanın FAT tablosundaki Başlangıç numarası
- [0x1C 4 byte] Dosyanın uzunluğu

Bu durumda BBB.TXT dosyamızın isminin de 0x00000040 adresinde saklandığını göreceksiniz

Kimlik bilgilerinden bizim için önemli olanlar;

- 0x00 adresindeki 11 byte alana yerleşmiş Dosya Adı ve uzantısı (8,3) xxxxxxxx.yyy.
- 0x1A adresindeki 2 byte alana yerleşmiş Dosyanın ilk "cluster" numarası
- 0x1C adresindeki 4 byte alana yerleşmiş Dosyanın uzunluğu

Hatırlarsanız AAA.TXT dosyasının içine 65536 byte veri yerleştirdim. 0x1C adresinde (0x20000020 adresini ofset kabul edin bu durumda relatif 0x1C adresinin gerçek değeri 0x2000003C olacaktır.)

[0x1C] de 0x00

[0x1D] de 0x00

[0x1E] de 0x01

[0x1F] de 0x00

0x00010000 yani 65536 byte demek. O halde AAA.TXT dosyasının 65536 byte olduğunu da biliyoruz.

Şimdi en önemli konuya geldik. AAA.TXT dosyası nerede?

0x1A ve 0x1B relatif adreslerinde dosyanın hangi "Cluster" da yer aldığı bilgisi var.

[0x1A]=0x02

[0x1B]=0x00

Yani 0x0002 nolu cluster'a bakacağız.

Artık dosya içeriğine bakmanın vakti geldi.

Root Directory alanı için 16KByte byte ayrılmıştır. Bir dosyaya ait kimlik bilgileri 32 Byte olduğuna göre Root Directory'ne en fazla 16384/32 yani toplam 512 dosya + klasör yerleştirebiliriz demektir.

Ben alt klasörlerle uğraşmayacağım ve ne var ne yok tüm dosyalarımı root altına koyacağım. Çünkü 512 dosya benim uygulamalarım için fazlasıyla yeterli.

Unutmadan; Dosya adlarını 8+3 olmak üzere kısa tutacağım ve sadece İngilizce karakterlerden oluşacak. Aksi takdirde bu fat işleri için fazladan kod yazmam gerekecek.

Neyse; Root Directory adresinden 16384 Byte ileri gidersem yani 0x00040000+0x4000=0x00044000 adresinde dosyaların içeriklerine ulaşırım.

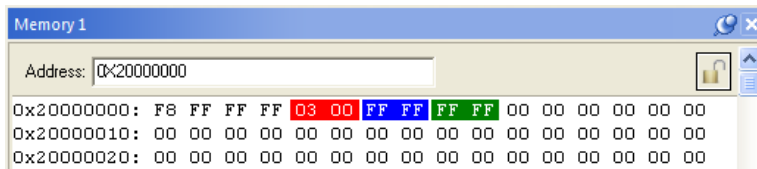
AAA.TXT dosyasının kimlik bilgilerinden uzunluğunun 65536 byte, cluster numarasının ise 2 olduğunu öğrenmiştik.

FAT16 da veriler 2 Byte saklanır. 2 nolu cluster'dan bahsediyoruz o halde buna ilişkin kayıt 2x2=4 adresindedir. Gidip FAT tablosundan 0x04 adresindeki 2 byte veriyi okuyalım.

Hatırlarsanız FAT adresini bir yerlerde saklayın demiştim. Ben bu adresi 0x00004800 olarak bulmuştum. Şimdi FAT tablosunda 2 x 2 = 0004 adresine bakmam gerekiyor. 0x0004, 512 içinde kaldığı için FAT'in ilk sektörünü (512 Byte'ını) okumalıyım.

Bu durumda CMD16(0x00 0x00 0x02 0x00) ve CMD17(0x00 0x00 0x48 0x00) komutlarını işleteceğiz.

"FAT"den hex verileri okuduğumda;



Kimlik bilgisinde adı geçen 2 nolu cluster numarası "FAT"de 4 adresinde 03 00 olarak işlenmiş ve değeri 0x0003 dür.

0x0003'ün anlamı 2 nolu cluster'daki veri alanındaki dataların tamamını oku geri yanıma ("FAT"e) gel 3 üncü kayıdına bak demektir.

Dataları okuduğumuzu varsayın ve hemen "FAT"deki 3. kayıda bakalım. Burada da FF FF var. Bu durumda veri alanında 3 nolu cluster'daki bilgileri okuyacağız ve okuma işlemi bitmiş olacaktı. (Başka veri yok) Yani verilerin tamamı 2. ve 3. cluster'da saklı.

3 nolu cluster'dan kaç byte veri okuyacağız? 65536 byte dosya uzunluğumuz. 1 cluster'da 32K var. O halde 65536-

32768=32768 yani 3.clusterdaki verilerin tamamını okuyacağız.

0xFFFF dosya bitti demektir. Sonuçta FAT bir zincirdir ve eğer dosya devam etseydi 3 nolu kayıta 0xFFFF değil bir başka numara olacaktı. (4 olmak zorunda değil.)

Diyelim ki 3 .kayıt 7 olsaydı ne olacaktı? Bu durumda 3 den 7'ye zıpyacaktık ve veri alanında 7.clusterdan verileri okuyup "FAT"deki yedinci kaydın 0xFFFF mi yoksa bir sonraki kaydın saklı olduğu cluster numarası olup olmadığına bakacaktık. Peki bir clusterda 32K veri olduğunu nereden biliyorum? **Yazımın başlarında Şu anda lazım olmayacak ama Data13=0x40 değerini saklayın.** demiştim. Bu bir clusterdaki sector sayısını verir. O halde 1 clusterda $0x40 \times 0x200 = 0x8000$ yani 32768 byte var demektir.

Sonuç olarak FAT16daki zincirin her bir halkası 16 Bitdir (2 Byte).

Zincir verilerinin anlamı.

0x0000: Boş cluster.
 0x0001: Rezerv edilmiş.
 0x0002.....0xFFEF verileri işaret eden cluster
 0xFFFF0.....0xFFFF6: Rezerv edilmiş.
 0xFFFF7: Bozuk sektor
 0xFFFF8.....0xFFFF Son cluster.

Tamamen atıyorum. FATdeki veriler aşağıdaki gibi ve dosyamızın kimlik bilgilerinde cluster no 2 olsun.

---0-----1-----2-----3-----4-----5-----6

0xFFFF8 0xFFFF 0x0003 0x0005 0x0006 0x0004 0xFFFF

Veri alanındaki 2. Clusterdaki bilgileri oku gel. FATde 2. kayıda bak. 3 yazıyor.
 Veri alanındaki 3. Clusterdaki bilgileri oku gel. FATde 3. kayıda bak. 5 yazıyor.
 Veri alanındaki 5. Clusterdaki bilgileri oku gel. FATde 5. kayıda bak. 4 yazıyor.
 Veri alanındaki 4. Clusterdaki bilgileri oku gel. FATde 4. kayıda bak. 6 yazıyor.
 Veri alanındaki 6. Clusterdaki bilgileri oku gel. FATde 0xFFFF yazıyor. Bitti.

FAT16daki zincir yapısını anladık. O halde gidip şu AAA.TXT dosyasını okuyalım artık.

İyi de nereden okuyalım?

Şimdi yazacaklarım kendi yorumumdur. Elimdeki SD kart için doğru sonuçlar vermektedir.

Root Directorynün adresi 0x00040000 olarak hesaplanmıştı. 16 K ilerisi de 2.cluster bilgilerinin içerir. Bu durumda Base Adres = $0x40000 + 0x4000 - 0x10000$ yazabiliriz.

Base Adres = 0x34000

Bu durumda verilen Cluster numarası için Adres= $0x34000 + (\text{ClusterNo} * 0x8000)$ den hesaplanır.

FAT sayesinde AAA.TXT dosyamızın 2 ve 3 nolu clusterlarda olduğunu biliyoruz.

2. Clusterdaki verilerin adresi= $0x34000 + (2 * 0x8000) = 0x44000$

3. Clusterdaki verilerin adresi= $0x34000 + (3 * 0x8000) = 0x4C000$

Ben dosya okumak için 512 byte buffer kullanacağım. Bu durumda;

CMD16(0x00 0x00 0x02 0x00)

CMD17(0x00 0x04 0x40 0x00) 2.Cluster 0.Sektor

CMD17(0x00 0x04 0x42 0x00) 2.Cluster 1.Sektor

CMD17(0x00 0x04 0x44 0x00) 2.Cluster 2.Sektor

.....

.....

CMD17(0x00 0x04 0xBE 0x00) 2.Cluster 63.Sektor

CMD17(0x00 0x04 0x4C 0x00) 3.Cluster 0.Sektor

CMD17(0x00 0x04 0x4E 0x00) 3.Cluster 1.Sektor

CMD17(0x00 0x04 0x50 0x00) 3.Cluster 2.Sektor

.....

.....

CMD17(0x00 0x05 0x3E 0x00) 3.Cluster 63.Sektor

Sırayla okumam gerekiyor. (1 cluster 64 sektörden oluşuyordu.)

BBB.TXT dosyasını okumak için;

Kimlik bilgilerinden Cluster No 4, Dosya boyu ise 18 byte olarak öğrenilir.

Dosya boyu 32K dan küçük olduğu için bir cluster yer kaplar. Gerçekten de FAT alanından 4.sıradaki değer **0xFFFF** yani 4 nolu clusterı oku yeter demektir.

O halde BBB.TXT dosyas için Adres= $0x34000 + (4 * 0x8000) = 0x54000$ olacaktır.

CMD17(0x00 0x05 0x40 0x00) deki 18 byte BBB.TXT dosya içeriğidir.

Kardeşim benim uygulamamda dosyalar en fazla 32K, zincirle uğraşmak istemiyorum da diyebilirsiniz. Bu durumda inanılmaz kısa kodla dosyalarınızı okuyuverirsiniz.

FATsız göbekler.....

Bu yazı [1- Ürünler](#), [2- Arm ve Asm](#), [3- Elektronik](#) kategorisine gönderilmiş. [Kalıcı bağlantıyı](#) yer imlerinize ekleyin.

Cnc Designer

WordPress ile güçlendirilmiştir.

6 6 6 4 1 8