

**Cnc Designer**

Matematiği seviyoruz

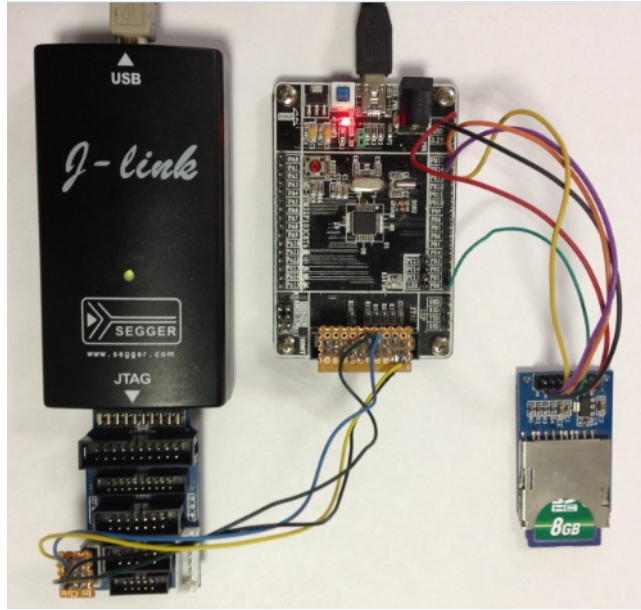
**SPI interface üzerinden SD Card çalışması**

Bülent Ünalmiş tarafından 22 Kasım 2013 tarihinde gönderildi



Mini Cnc makineler için tasarladığım kartlar RS232 yada USB üzerinden bilgisayara bağlanıyordu. Artık PC den bağımsız çalışabilen verileri SD kart üzerinden okuyan cihazlar yapmam gerektiğine karar verdim.

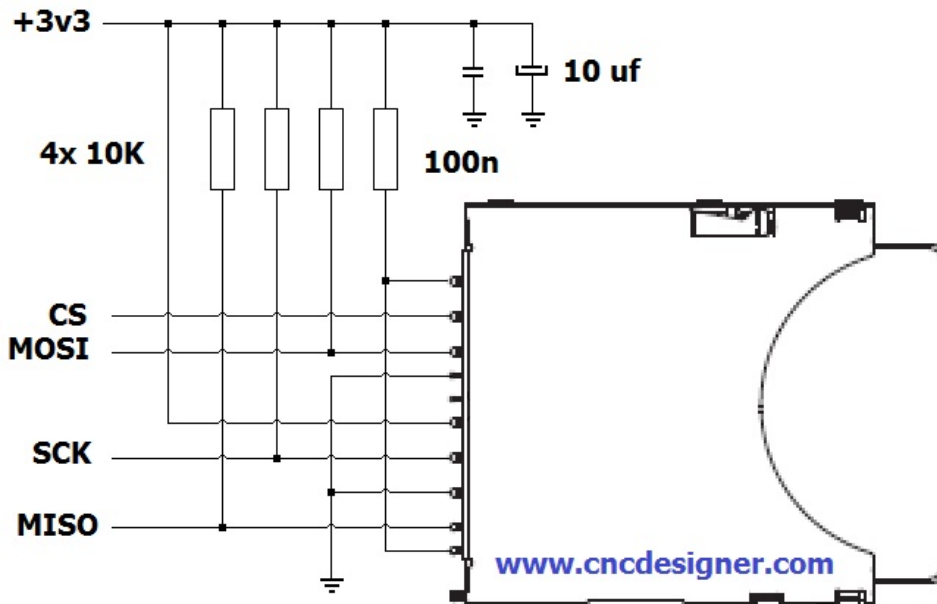
STM32F103C8 kitine SPI üzerinden SD kart konnektörlü minik bir kart bağladım ve bu sistem için kodları yazmaya başladım.



Program yazımında karşılaştığım sorun doküman eksikliği oldu. Aslında ortalıkta pek çok doküman var fakat kafama takılan sorulara cevap almakta zorlandım.

Akla gelen bazı soruların cevabı verilmemiş verilenlerde de detaya boğulmuş.

Emin olun konuyu güzel özetleyeceğim.



SD kart denen malzeme, içinde işlemci (lojik kontrol ünitesi) ve flash barındıran bir elektronik devre. Biz kendi işlemcimizle doğrudan flasha ulaşamıyoruz. SD kart içindeki kontrol ünitesiyle konuşarak isteklerimizi bu içerideki üniteye yaptırıyoruz.

SD kartın bir sürü komutu var. Üstelik zaman içinde SD kart versiyonları ilerlemiş ilk çıkan komut listesine



yeni komut listeleri eklenmiş vs vs. Durum böyle olunca eski versiyona sahip kartlar (ki bu kartlar genelde düşük kapasiteli oluyorlar) için yazılmış programlar yeni kartlarla çalışmayacaktır.

**SD Card ile konuşmaya başlayalım.** CPU kartınızla SD konnektörünü barındıran kartınızı birbirine bağlayın. Bunun için SPI pinlerini kullanacağız.

4 adet sinyale ihtiyacımız var. Bunlar MISO, MOSI, SCLK, bir adet de output pin.

CPU                      SD Card Kiti

MOSI----- SD Card kitinin data girişi

MISO----- SD Card kitinin data çıkışı

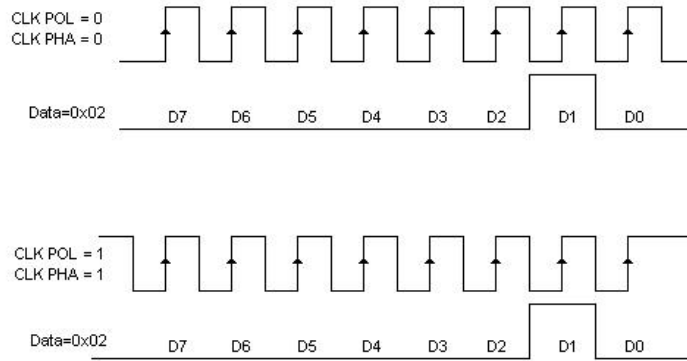
SCLK----- SCLK

SEL ----- CS

+5v ve Gnd de işin içine girince toplam 6 kablo gerekiyor. **Dikkat: Benim uygulamamda SD kite 5v gidiyor fakat SD karta 3.3v gidiyor.**

Burada SEL adını verdiğim sinyal, işlemcinin herhangi bir output pini. Bu pini 1 yaparsak SD kart kiti off, 0 yaparsak SD kart kiti on olacak. (Tabiki power on/off anlamında değil chip select anlamında)

**Peki SCLK ayarlaması nasıl yapılmalı.**



Yukarıda 0x02 datasının Clk eşliğinde data hattındaki yolculuğunu görmekteyiz. İster üstteki modu isterseniz alttaki modu seçebilirsiniz. SPI ünitesini 8 bit modunda çalıştıracacağız.

**İlk işlemiz SD kartı SPI moduna geçirmek olacak.**

Maleşef SD kartının Reset pini yok. Resetleme işlemi dahil herşeyi komutla yapacağız. Bu aşamada Kart ile max **400Khz**'lik CLK ile haberleşebiliriz.

- CS pinini H yapın.
- peşpeşe 10 tane 0xFF yollayın.
- CS pinini L yapın.
- 0x40 0x00 0x00 0x00 0x00 0x95 yollayın. (İlk önce 0x40 yollanacak)
- 0xFF yollayın ve gelen verinin 0xFF olup olmadığına bakın.
- Eğer gelen veri 0xFF ise gene 0xFF yollayın. Bu işlemi 10 kere falan yapın.
- Eğer 0xFF den farklı veri alırsanız 0xFF yollama işini bitirin. 0xFF den farklı değerde okunan değer SD kartın

bize yolladığı durum bilgisidir. Bu bilgi bir yerlerde dursun.

- CS hattını H yapın.
- 0xFF yollayın.

Eğer durum bilgisi dediğim bilgi 10 denemede de alınmadı ise bunun 2 sebebi olabilir.

Birinci neden, SD kartı takmayı unuttunuz.

İkinci neden, Programınız hatalı.

SD kartın gönderdiği durum bilgisi 0x01 olmalıdır.

Eğer SD kartı takdığınız halde 0xFF haricinde data okuyamıyorsanız Kablo bağlantılarınızın doğru olduğunu varsayarsak muhtemelen SPI gönderme alma rutininde yazılım hatası yapmışsınızdır.

SPI dan data gönderme registerine göndermek istediğiniz veriyi yazın. Datanın gitmesini bekleyin ardından gelen datayı okuyun. Datanın gidip gitmediğini SPI Tx Buf Empty benzeri isme sahip bir flag ile takip etmelisiniz. (Delay rutinleriyle falan aman ha uğraşmayın.)

Önerdiğim yazım şekli işlemcinin döngüde kalmasına neden olan kötü bir yazım şeklidir. Fakat olayı kavradıktan sonra bu kısmı interrupt tipi yazım şekline çevirerek işlemciyi boşu boşuna veri giderken bekletmemiş olursunuz.

Sonuç olarak CMD0 komutu 0x40 0x00 0x00 0x00 0x00 0x95 olmak üzere 6 byte yani 48 bit uzunluğundadır.

### **Neden her gönderdiğim veriye karşılık her defasında 0xFF cevabı geliyor?**

MISO pini yani SD kartın gönderdiği bilgileri okuduğumuz pin SD kart tarafında bir dirençle beslemeye bağlıdır. Dolayısı ile SD kart hiç veri yollamasa bile siz bu pini high olarak okursunuz. 8 tane 1 biti 0xFF edeceğinden gönderdiğiniz verilere karşılık 0xFF okursunuz. Ne zaman SD kart denetleyicisi size gerçekten veri yollar işte o zaman 0xFF den farklı data okursunuz. SD kart her gönderdiğiniz veriye cevap vermez. Veri paketlerini okur sonra cevap verir.

Eğer Reset işlemi ardından SD karttan 0x01 durum bilgisi aldıysanız Reset komutunda gönderilen verilerin ne anlama geldiği üzerine konuşalım.

### **CS=High iken en az 72 clk boyunca MOSI den 1 yollamanın amacı.**

SD karta enerji verildiğinde kart SD interface modundadır. Bu interface'in detaylarına girmek istemiyorum. Bizim amacımız SD kartı SPI modunda kullanmak.

Dolayısı ile SD kartın SD moddan SPI moduna geçirilmesi gerekiyor. Bu amaçla CS=1 yapıyor en az 72 clk boyunca karta 1 bilgisi yolluyoruz.

$72/8=9$ , 9 tane FF yollarsan 72 clk boyunca 1 yollamış oluyoruz. Ben 9 yerine 10 tane 0xFF yolladım.

Bu işlem ardından CMD0 denen komutun yollanması gerekiyor. Komutlar 6 bit uzunluğundadır. CMD0 komutunun hex karşılığı 0x00 dır. (Sağdaki 6 adet bit)

Hemen anlatımda anlaşalım, en düşük bit, 0. bitidir. O halde 1 bytelık veride 7 ve 6 bitler boşa kaldı. 6. bit verinin yönünü bildirir. Biz bu biti 1 yaparak verinin SD karta doğru olacağını belirtiyoruz. 7. Bit ise daima sıfırdır.

Bu durumda 0 1 000000 olmak üzere 0x40 elde ederiz. İşte adı geçen 0x40, CMD0 kodu için yola çıkartılacak bu

0x40 dır.

CMD0 komutunu 4 byte'lık parametre bloğu izler ve bu parametreler 0x00 dır. (Kanunu yazanlar öyle demişler)

0x95 ise 0x40 0x00 0x00 0x00 0x00 için hesaplanmış CRC değeridir. SD kart, CMD0 ve CMD8 için kesinlikle doğru hesaplanmış CRC değeri ister. Diğer komutlarda CRCye bakmaz. Fakat istersek CMD59 komutu ile diğer komutlarda da CRC kontrolü yap diyebiliriz.

Unutmadan CRC 7 bit'tir. Bunu 8 bite yayarken sola dayalı olarak yazıyoruz ve en düşük bite (Bit0'a) 1 değeri veriyoruz. (En son gönderilen bit daima 1 olmak zorunda)

Neyse şimdilik CMD8 vs yi unutun. Nerde kaldık en son 0x95 CRC değerini yollamıştık. Artık SD kart içindeki işlemci CMD0 komutunu, 4 parametresini vs CRC değerini aldı artık içeride bir takım işler yapmak için zamana ihtiyacı var. Komutu yorumlayacak, parametresini kontrol edecek, CRC sini hesaplayacak vs vs. Biz SD kart içinde olup bitenleri bilemeyiz. Bu nedenle SD karta anlam içermeyen 0xFF datası yollarız. SPI yapısı gereği her giden 8 bit veriye karşılık karşıdan 8 bit veri alır. Eğer SD kart içindeki işlemci shift registre değer yüklemiyse bize cevaben 0xFF gelecektir. Ne zaman SD kart içindeki işlemci shift registerine veri yazarsa o zaman biz gönderdiğimiz 0xFF değerine karşılık 0xFF den farklı bir değer okuruz. İşte bu an haaa SD kart bize veri yollamış yorumunu yaparız.

0xFF den farklı değeri aldığımızda artık CS hattını high yaparak (Aslında CS değil !CS demek daha doğru olur) SD kartın SPI bağlantısını keseriz.

Açıkcası neden CS=1 yaptıktan sonra bir 8 tane daha 1 biti yolluyoruz bu bilgiye ulaşamadım. Kanunu koyanların kararı olsa gerek. Zira diğer komutlarda da bu böyle.

Şimdi gelelim gelen 0x01 değerinin anlamına.

SD karttan gelen cevaplara kanun yazarlar değişik isimler vermişler. Bunlar R1, R1b, R2, R3.. gibi.

CMD0 komutu **R1** cevabı yollar. **R1** cevabı 8 bit olup birlerinin anlamı aşağıdaki gibidir.

Bit7 = 0

Bit6 = Parametre Hatası

Bit5 = Adres Hatası

Bit4 = Silme komut diziliminde hata var

Bit3 = CRC hatası

Bit2 = Geçersiz komut

Bit1 = Erase Reset (ne demekse)

Bit0 = Aylak modu

Biz 0x01 aldığımızı göre Aylak modundayız demektir. Bu mod kurulumun devam ettiği anlamına geliyor.

Bit0 dışındaki diğer bitlere takılmayın SD karttan bu aşamada 0x01 cevabı almak zorundayız. Alamadıysak bir daha Reset işlemi yaptırın. Ne zaman 0x01 aldınız o zaman işlem tamam demektir.

### **Şimdi karta CMD8 yollayacağız.**

SD kartlar V1.X ve V2.00 olmak üzere 2 öncesi ve sonrası şeklinde ikiye ayrılıyor. Yaşlı olan seri yeni komut listesini tanımıyor. Bu nedenle öncelikle SD kartımızın versiyonunu ve çalıştırmak istediğimiz voltajda sorun çıkartıp çıkartmayacağını bilmemiz gerekir.

Bu amaçla SD karta CMD8 komutu yollayıp bu komuta cevap verilip verilmediğine bakacağız. (Cevap gelmezse kartımız V1.x demektir.)

- 0x48 yolluyoruz.
- 0x00 yolluyoruz
- **0x01** yolluyoruz (Ben kartımı 2.7 - 3.6v aralığında çalıştırmak istiyorum demek)
- **0xAA** yolluyoruz. (Bu bir desen siz isterseniz başka bir şey yazın)
- 0x87 yolluyoruz. (Sola dayalı 7 bit CRC olacak ve Bit0=1)
- SD karttan 6 byte veri gelinceye kadar SD kartı 0xFF ile bombalıyoruz. (Ben her bir byte için 10 kez deneme yapıyorum)

Şimdi 0x48 in ne anlama geldiğini zaten anladınız. 0x40 verinin yönünü 8 de CMD8 in sekizini bildiriyor.

CMD8 komutu da CMD0 komutu gibi toplam 48 bit uzunluğundadır.

8 bitlik 0x48 verisinden sonra 20 tane değerleri 0 olan bitleri yollarız. Etti 28 bit.

4 bitlik voltaj bilgisi yollarız etti 32 bit.

8 Bitlik Kontrol şablonu denilen veri yollarız. Etti 40 Bit.

7 bit de CRC yollarız etti 47 Bit.

En son olarak da 1 bitlik 1 değeri yollarız. (En son bit hep 1 di)

Sonuç olarak CMD8 komutumuz 0x48 0x00 0x00 0x0 **0x1 0xAA** 0x87 olarak gönderilir.

0x00 0x00 0x0 1 takip eden 0x1 voltaj değerini aşağıdaki binary tablodan elde ediyoruz.

0000 Tanımsız

0001 2.7-3.6v

0010 Düşük voltaj aralığı için reserve

0100 Rezerve

1000 Reserve

Diğerleri Tanımsız

Biz 0x1 yolladığımıza göre kartımızın 3.3v da çalışıp çalışmayacağını sorguluyoruz demektir.

Kontrol şablonu herhangi bir değer olabilir. **0xAA** yollamak âdettendir. 0x87 ise bu dizilişe ait hesaplanmış CRC değeridir. (Aslında CRC=43 bunu sola dayalı yazınca 86, sıfıncı biti de 1 yaptık)

SD kartımız V2.00 ve üstü ise CMD8 komutuna toplam 40 bitlik (5 Byte) **R7** adı verilen cevabı verir.

Eğer cevap vermezse versiyonu 2.00 dan düşük demektir.

**R7** cevabında alınan ilk 8 bit CMD0 komutuna gelen **R1** cevabı formatındadır. Sonraki 4 bit versiyon, sonraki 16 bit reserve, sonraki 4 bit voltaj değeri, son 8 bit de kontrol şablonudur.

Ben kendi kartımda CMD8 için 0x01 0x00 0x00 **0x01 0xAA** cevabı aldım.

Bu durumda en soldaki 0x01 den (**R1** cevabı) kartımın hala aylak modda olduğunu anlıyorum.

0x00 0x00 reserve bitler. 0x01 den voltajın da 1 nolu değer olduğunu anlıyorum. Yani SD kart biraz önce verdiğim tabloya göre 2.7..3.6v da çalışabilirmiş. 0xAA cevabı ise aslında benim yolladığım 0xAA nın taa kendisi. Bu bir bakıma hata kontrolü için de kullanılabilir.

Eğer R7 cevabının R1 bileşeninde illegal command biti 1 ise bu durumda SD kartımız V1.x kartıdır. (Yani kart CM8'i tanımamış demektir.)

CMD8 için 0x01 0x00 0x00 0x01 0xAA cevabını aldıysanız artık ACMD41 komutunu gönderebiliriz.

Eğer SD kart CMD8 için illegal command deseydi elimdeki kart V1.X standart kapasitede bir kart demektir. Ben kendi çalışmamda bu tip eski kartlara destek vermeyeceğim. Böyle bir kart takıldığında uyumsuz kart taktınız mesajı ile kullanıcıyı uyaracağım.

### Şimdi karta ACMD41 komutu yollayacağız.

Şu ana kadar komutlarımız CMDx şeklinde isimler alırken Application Specific Command yani uygulamaya özgü komutlar ACMD komut olarak isimlendiriliyor. ACMD41 komutu tek başına yollanmaz. Bu komuttan önce CMD55 yollamak gerekiyor.

- CS hattını 0 yap
- 0x77 0x00 0x00 0x00 0x00 0x01 yolla
- 0xFF yolla, 0xFF geldi ise 0xFF harici veri gelene dek 0xFF yollamaya devam et.
- 0x01 gelmesi gerekiyor. Baska bir şey geldiyse en basa don.
- 0xFF yolla (Yollamazsan bir sonraki dataları yolladığında 0x00 değil 0xE0 geliyor)
- 0x69 0x40 0x00 0x00 0x00 0x01 yolla
- 0x00 gelinceye kadar 0xFF yolla.
- CS hattını 1 yap.
- 0xFF yolla.

Artık SD kartımız Idle durumundan çıktı. Bu en son alınan 0x00 R3 cevabı oluyor.

Şimdiki işimiz kartımızın SD, HCSD olup olmadığını ve çalışma voltaj aralığını öğrenmek.

Bu aşamada SPI clk frekansımı 19Mhz'e yükselttim ve SD kart hala benimle konuşmaya devam ediyor. Fakat aşağıdaki gibi de yapabilirsiniz.

0x49, 0x00, 0x00, 0x00, 0x01 yollayıp 0x00 ve sonrasında 0xFE bekleyin. Şimdi data çekmeye başlayın. 1. gelen değil, 2. gelen değil 3. gelen de değil. 4.gelen veri 0x32 yada 0x5A olacak (benim karttan 0x32 geldi). 32 ve 5A'nın anlamını bulursam yazırım bir ara. Çünkü bunlar SD kartınızın max data transfer hızını bildiriyor.

### Şimdi karta CMD58 yollayacağız.

- CS hattını 0 yap.
- 0x7A 0x00 0x00 0x00 0x00 0x01 yolla.
- 5 byte veri gelinceye kadar 0xFF yolla (ilk byte 0xFF den farklı olacak)
- CS hattını 1 yap.
- 0xFF yolla.

CMD58 komutu R3 adı verilen 5 bytelik (40 Bit) cevabı verir. Bu cevabın ilk byte'i R1 cevabı ile aynı yapıdadır. Geri kalan 32 Bit OCR register içeriğini bildirir.

R1 cevabı 0x00 olmak zorunda.

8GB SD kart bana OCR değeri olarak 0xc0 0x80 0x00 0xff yollarken  
2GB SD kart bana OCR değeri olarak 0x80 0x80 0x00 0xff yolladı.

Bu şartlar altında her iki SD kartın **voltajı istenen değere ulaşmış**, 8GB SD HC (High Capacity), 2GB olan ise Normal kapasitede bir kart imiş.

### OCR registerinin bitlerinin anlamı

OCR31: SD kart voltajı nominal değerine ulaşmışsa 1 aksi halde 0

OCR30: 1 ise kartımız High Capacity, 0 ise SD card demek

OCR29...OCR24:

OCR23: 1 ise SD kart 3.5 - 3.6v aralığında çalışıyor demektir.

OCR22: 1 ise SD kart 3.4 - 3.5v aralığında çalışıyor demektir.

OCR21: 1 ise SD kart 3.3 - 3.4v aralığında çalışıyor demektir.

OCR20: 1 ise SD kart 3.2 - 3.3v aralığında çalışıyor demektir.

OCR19: 1 ise SD kart 3.1 - 3.2v aralığında çalışıyor demektir.

OCR18: 1 ise SD kart 3.0 - 3.1v aralığında çalışıyor demektir.

OCR17: 1 ise SD kart 2.9 - 3.0v aralığında çalışıyor demektir.

OCR16: 1 ise SD kart 2.8 - 2.9v aralığında çalışıyor demektir.

OCR15: 1 ise SD kart 2.7 - 2.8v aralığında çalışıyor demektir.

OCR14 .. OCR8: Rezerve

OCR7: Düşük voltaj aralığı için rezerve

OCR6 .. OCR0: Rezerve

Yazdığım program şu ana kadar yapılan işlemler ışığında kartın takılı olup olmadığını, takılı ise yüksek kapasiteli olup olmadığını söylüyor.

### Şimdiki aşama, sektörden okunacak veri miktarını belirleme işlemi.

Tam karşılığı olmasa da, bir kerede okunabilecek en fazla veri yığını barındıran veri alanına sektör diyoruz. Standart SD kartlarda bu veri yığını 1K kadar olabilirken HC SD kartlarda bu 512 Byte oluyor. Ancak okunacak veri miktarını CMD16 ile daha az olacak şekilde sınırlandırabiliyoruz. Bu durumda örneğin 4 byte veri okuyalım diyebiliyoruz.

CMD16 okuma komutu değil okunacak veri sayısını belirleme komutudur.

Örneğin 256 adet veri okuyacaksam 32 bitlik parametremiz 0x00000100 olacaktır.

Bu komutun 32 bitlik (4 Byte) parametresinin yollanma sıralaması büyük anlamlı byte ile başlar. Daha açık ifade ile 0x0000100 verisinde, önce 0x00 sonra 0x00 sonra 0x01 sonra da 0x00 yollanır.

O halde CMD16 komutunu yollamak istersek;

- CS hattını 0 yap.
- 0x50 yolla
- 32 bitlik parametreyi yolla.
- 0x00 gelinceye kadar 0xFF yolla.
- CS hattını 1 yap.
- 0xFF yolla.

CMD16 komutu da anlamlı CRC istemediği için 0x00 okuyabilmek adına gönderilen 0xFF lerden ilkini CRC



olarak kabul eder.

### **Artık SD karttan verimizi okuyabiliriz.**

Veri okuma işlemi sektör bazında olabileceği gibi peş peşe sektörleri okuma şeklinde de olabilir. Örneğin 0. sektörü okumak yada 0. 1. 2 sektörleri peşi sıra okumak farklı komutlarla yapılır.

PC ye kıyasla göreceli olarak çok düşük boyutlu ram içeren işlemcilerle çalıştığım için 512 byte lık tek sektör okumakla yetineceğim. Aksi takdirde okunan verileri rama yerleştirmek için büyük rama ihtiyacım olacak. Bu nedenle sadece CMD17 okuma komutunu açıklayacağım. Bu komutu kullanmam gerektiğinde okumak istediğim sektörden en fazla 512 byte okuyacağım bu verilerle işlem bittiğinde bir başka sektörü okuyacağım.

### **CMD17 komutuyla sektör okumak.**

CMD17 komutu 0x51 olup bunu 32 bitlik parametre takip eder. Bu komut anlamlı CRC değeri istemediği için 0xFF ile veri çekmek isterken CRC değeri 0xFF olarak yollanır.

2GB kartta

Parametre = Sektör No x 512

Eğer HCSD kart kullanılacaksa (Örneğin 8GB SD Card)

Parametre = Sektör No olarak bildirilir.

*32 bitlik parametre CMD16 parametresinde olduğu gibi MSB den başlanarak yollanır.*

Örneğin 1. sektoru okumak için 32 bitlik Sd kart parametremiz 0x200, HCSD kart parametremiz ise 1 olacaktır.

SD Card parametresi : 0x0000 0000 0000 0200

HCSD Card parametresi: 0x0000 0000 0000 0001

- CS hattını 0 yap.
- 0x51 yolla
- 32 bitlik parametreyi yolla.
- 0x00 gelinceye kadar 0xFF yolla.
- 0xFE gelinceye kadar 0xFF yolla.
- **0xFF yolla veri oku, 0xFF yolla veri oku, 0xFF yolla veri oku .....**
- 2 byte CRC okumak için 0xFF yolla oku, 0xFF yolla oku.
- CS hattını 1 yap.
- 0xFF yolla.

Sektör okumayı da bitirdiğimize göre artık SD kartta dilediğimiz sektöre ulaşım içeriğini öğrenebileceğiz. Ancak PC ile SD kart içine kaydedilmiş bir dosyaya kendi işlemcimizle nasıl ulaşılacağız?

[Bu konuyu FAT üzerinden dosyaya ulaşma başlıklı yazımda ele alacağım.](#)

Şimdilik ihtiyaç duymadığım için SD karta yazma işlemi açıklamıyorum.

Anlatımlarımı dilediğiniz işlemci için koda dönebilirsiniz.

Bu yazının pdf versiyonu: [SPI-SD-Card](#)



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

---

**Cnc Designer**

*WordPress ile güçlendirilmiştir.*

**649559**