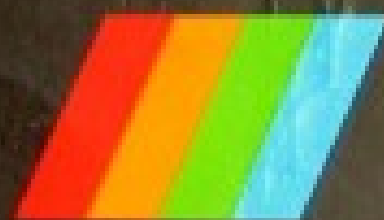


**sinclair**

# **ZX Spectrum+**

## **El Kitabı**



**SPECTRUM YAZILIMLARI**  
Spectrum bilgisayarları için  
kullanılabilen yazılımların tümü yeni  
ZX Spectrum+ 'unuza da uyar.

## ZX SPECTRUM+'u SUNARKEN

Sinclair Research, herkese bilgisayar kullanma olanağı getirmiş olan mikroyonga teknolojisiinde uzun zamandır önde gidiyor. Dünyanın ilk ucuz mikrobilgisayan ZX80'i piyasaya sunduktan sonra, bunu giderek gücü ve değeri artan modellerle, ZX81, ZX Spectrum ve QL bilgisayarlarıyla sürdürdük. Bunların tasarımıda da, çalıştırılmasında da parolamız, kolay kullanım olmuştur. ZX Spectrum+, bu alanda bizi bir adım daha ileri götürüyor. Bunda Spectrum'un en iyi yanları alınmış, daha da geliştirilmiş ve böylece bilgisayarların en güçlüsü, en popüler ve kullanımı en kolayı yaratılmıştır. Bu yeni bilgisayarın size sunduğu geniş olanakları değerlendireceğinizi umuyoruz.

*Chris Smith*

## İÇİNDEKİLER

### İŞE KOYULALIM 3

### PROGRAMLAMAYA BAŞLAYALIM 17

### ZX SPECTRUM+'U TANIYALIM 41

### SINCLAIR BASIC'I ÖĞRENELİM 49

Yazan Neil Ardley  
Yayınlayan Dorling Kindersley Ltd  
Sinclair Research Ltd'in işbirliğiyle

## BU KİTABIN KULLANIMI

ZX Spectrum+'unuzun kılavuzu, aynı aynı renklerde düzenlenmiş dört bölümden oluşuyor. Bir bölümü bulmak için, o rengi taşıyan sayfaları açmak yeterlidir.

## 1 İŞE KOYULALIM

ZX Spectrum+'unuzun çalışmaya hazırlanışı ■  
Televizyonunuzun ayarlanışı ■ Hazırlamada karşılaşılan sorunlar ■ ZX Spectrum+ neler yapabilir ■ Hazır yazılım nasıl kullanılır ■ Program nasıl yüklenir ■ Yazılım yüklemadaki sorunlar

## 2 PROGRAMLAMAYA BAŞLAYALIM

Klavye – bilgisayarınızın komuta kontrol tablosu ■  
Tuşlar nasıl çalışır ■ Televizyon hesap makinası ■  
Renkler ve kullanımı ■ Basit bazı grafikler ■ Resim defteri olan ekran ■ Desen ve resim çizimleri ■  
Bilgisayar karakterlerinin yarabılışı ■ Hareketli resimler ■ Müzik ve çeşitli sesler ■ Programlarınızı nasıl saklayabilirsiniz ■ Program saklamada karşılaşılan sorunlar

## 3 ZX SPECTRUM+'u TANIYALIM

## 4 SINCLAIR BASIC'İ ÖĞRENELİM

Sinclair BASIC komutları için programcının başvuru kılavuzu ■ Spectrum ekranındaki mesajlar ■ BASIC'in ötesi ■ Bilgisayar jargonu – ne anlama geliyor.

## İŞE KOYULALIM

Bu bölüm size, elinizdeki ZX Spectrum+'un geniş olanaklarını araştırmaya nasıl başlayabileceğinizi gösterecek. Bu arada, bilgisayarınızı nasıl istendiği anda çalışmaya hazır durumda tutabileceğinizi de öğreneceksiniz. Burada bir seçim yapma durumundasınız. Ya Spectrum+'unuzun yeteneklerinden yararlanıp, renkli grafiklerini ve çıkarabileceği sesleri kullanarak birkaç program hazırlayabilirsiniz ya da bilgisayar oyunları gibi bazı hazır yazılımları nasıl kullanacağını öğrenebilirsiniz. Seçiminizi ne yönde yaparsanız yapın, kısa zamanda bilgisayarınızı kullanmanın keyfine varacaksınız.



## ZX SPECTRUM+'UN ÇALIŞMAYA HAZIRLANIŞI

Spectrum'unuzu çalıştırmak için, önce aşağıdaki listeyi gözden geçirin, gereksinim duyacağınız bütün gereçlerin bulunup bulunmadığını kontrol edin, sonra da karşı

sayfadaki açıklamaları izleyerek, bağlantıları yapın, bilgisayarınızı da fişe takın. Bu aşamada teypte gereksiniminiz yok.

Bağlantıları iyice yapıp yapmadığınızı kontrol edin. Çünkü Spectrum'unuzu kullanırken, kaza ile bağlantıları yerinden çıkarırsanız, ya da cereyan kesilirse, programınız, Spectrum'daki bilgiler veya sonuçlar silinir.

Bilgisayarla işiniz bitince, fişi prizden çekin.

### Listeniz: Kullanacağınız gereçler tamam mı?

Ambalajı açınca, bulacaklarınız:

1. ZX Spectrum+
2. ZX Adaptör/ Transformör - Spectrum için gerekli 9 voltluk doğrudan akımı sağlar.
3. Anten bağlantısı - Spectrum'unuzu televizyona bağlar.
4. Teyp bağlantısı - Spectrum'unuzu teypte bağlar.
5. Garanti belgesi - Bunu hemen doldurup, postalamana gerekir.
6. Kullanım Kılavuzuna Ek Kaset.
7. Bu elkitabı.

Elinizde olması gereken diğer araçlar:

1. Bir televizyon.
2. Bir kasetli teypt.
3. Bir fiş.



### Hazırlıklara ilişkin sorular ve yanıtlar

**Televizyonumun renkli olması gerekir mi?**  
Hayır. Ancak siyah beyaz televizyonda, Spectrum'un sağlayacağı renkten göremezsiniz.

**Herhangi bir televizyon olur mu?**  
Spectrum'unun herhangi bir televizyonda görüntü oluşturabilmesi gerekir. Görüntü oluşmuyorsa, o zaman bilgisayarınız ve televizyonunuzun sistemleri farklı demektir. Televizyonunuz çok eskiyse, ya da televizyon ve bilgisayar farklı ülkelerden satın alınmışsa, böyle bir durumu karşılayabilirsiniz. Bu konuda bir kuşkunuz varsa, televizyon satıcınıza danışın.

**TV alıcısı yerine monitör kullanabilir miyim?**  
Evet. Satıcıdan, size Spectrum için çok daha üstün bir görüntü verecek bir monitör de alabilirsiniz.

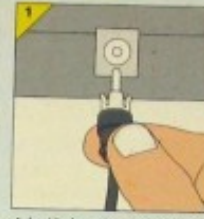
**Nasıl bir elektrik akımı gerekir?**  
Spectrum 240v/50Hz 1.4 amperlik standart akımı kullanır.

**Spectrum parazite yol açar mı?**  
Eğer radyonuz bilgisayara yakınsa, parazite neden olabilir. Ama bu durum radyoya da, bilgisayara da bir zarar vermez.

**Bununla ZX 16K RAM kullanabilir miyim?**  
Hayır. Bu RAM takımı sadece Sinclair ZX81 bilgisayarı ile kullanılabilir.

### ZX Spectrum+'un uzun cereyana bağlanması

Önce adaptörden çıkan, ucu çıplak kabloyu bir fişe bağlayın. Spectrum'unuzun topraklanmasına gerek yoktur. Bundan sonra, yandaki resimleri izleyerek, Spectrum'u elektrikli ve TV alıcısına bağlayın. Bütün bağlantıları tamamladıktan sonra, nasıl ayar yapacağınızı öğrenmek için sayfayı çevirin.



Adaptörden çıkan kablounun ucundaki küçük fişi, Spectrum'da üzerinde 9VDC yazan prize takın.



Adaptörden çıkan diğer kablounun ucundaki normal fişi elektrik prize takın.



Anten bağlantısının ucundaki fişi Spectrum'da üzerinde TV yazan prize takın. Bu bağlantıdaki fişlerden sadece biri bu prize uyar.



Televizyonunuzun anten kablounu çıkarın. Anten bağlantısındaki diğer fişi de televizyonun anten girişine takın.



Televizyonunuzu açın, ses düğmesini sonuna kadar kısın. Arka televizyonunuzu Spectrum'dan gelecek sinyalleri almaya ayarlayabilecek durumdasınız.

### Spectrum fişleri ve bağlantıları



## TELEVİZYONUNUZUN AYARLANMASI

Spectrum'unuz, UHF bandının 36. kanalı üzerinden renkli televizyon video sinyalleri gönderir, bu nedenle bilgisayarın sinyallerini alabilmesi için TV alıcınız bu kanala ayarlanmalıdır.

Spectrum'unuzu ceryana ve TV alıcısına bağladığınızda, alıcınızın istasyon alıcı düğmesini, aşağıdaki ilk ekranda gördüğünüz gibi Sinclair'in copyright sinyalini alacak şekilde ayarlayın. Bunu yaptığınızda, Spectrum'un renklerini sinamaya ve aracınızı kullanmaya hazırsınız demektir. Eğer copyright görüntüsünü alamazsanız ya da renkler iyi değilse, o zaman karşı sayfadaki tablodan bağlantıları kontrol edin.



### Spectrum'un renklerinin kontrolü

Spectrum'un renklerini kontrol etmek için, önce B tuşuna sonra da 1 ile 6 arasında bir rakam tuşuna basın. Copyright görüntüsü kaybolacak, önce BORDER kelimeleri sonra da bir rakam görünecektir. ENTER yazılı tuşa basınca, ekranın,

#### BORDER 4



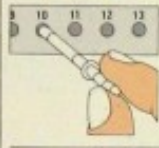
### Ayar düğmeleri

**Döner düğmeli ayar**  
Döner düğme herhangi bir kanala ayarlanabilir. Copyright sinyalini bulana kadar düğmeyi çevirin.



### Tuşlu ayar

Bilgisayar için kullanılacak bir tuş seçin, sonra da copyright sinyalini alacak şekilde ayar yapın. Eğer mümkünse bir tuşa buna ayarın. Böylece Spectrum'u çalıştırmak istediğinizde her seferinde ayar yapma zorunda kalmazsınız.



### Elektronik ayar

Bu sistemde, alıcı kendini istediğiniz kanala ayarlar. Elle ayarlanamayan, sentezle kanal seçimli TV alıcılara bu bilgisayarın kullanımı için elverişli olamaz.

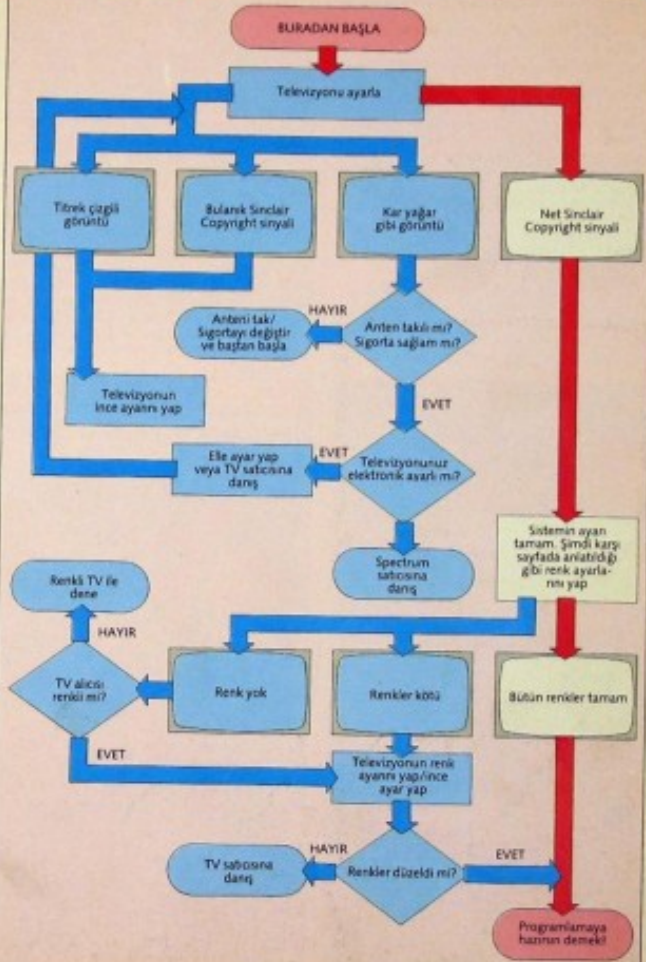


#### BORDER 3



çevresi, rakam tuşunun rengini alacaktır. Aşağıdaki ekranlar önce BORDER 4 ve ENTER, sonra da BORDER 3 ve ENTER tuşlarına basıldığınızda neler olacağını gösteriyor. BORDER 7 ise ekranın çevresini yeniden beyaz yapar.

## Hazırlamada karşılaşılan sorunlar





**Dene**  
(K tuşunu kullanarak) program sıralamasında önce PAPER 7 yazın ve ENTER'e basın. Sabr 20'yi yeniden yazın ama FLASH RND komutlarını koymayın. Halkaların yanıp sönmeleri durur.

## ÇILGIN MOZAIK

```

10 BORDER 0:CLS
20 LET X=15:LET Y=31:LET V=
30 INPUT "X,Y=";X,Y
40 FOR I=1 TO 20
50 PRINT AT Y,I:CHR$ 345
60 LET I=I+1
70 LET Y=Y+2
80 IF Y=31 THEN LET Y=1
90 IF X=15 THEN LET X=31
100 IF Y=31 THEN LET Y=1
110 IF X=31 THEN LET X=15
120 IF Y=31 THEN LET Y=1
130 GO TO 20

```



Renkli bir kare ekranda ileri geri hareket ederek, renkli desenler oluşturur. Programı her başlatıldığında başka desen biçimlenir.

## Dene

Satır 50'de 143'un yerine 42 yazınca, ortaya yıldızlar çıkar. 33'den 142'ye kadar başka rakamları da deneyebilirsiniz. Ne olacağını görmek için 51. sayfadaki tuş tablosuna bakabilirsiniz.

## Bir program nasıl yeniden başlatılır?

YILDIZLAR ÇUBUKLAR gibi bazı programlar sona erince, ekranda son satır olarak OK yazısı belirir. Bu, bütün programın bittiği anlamına gelir. Yeniden başlamak için RIRUN ve ENTER'e basmak yeterlidir.

Diğer programlar ise, ya ÇILGIN MOZAIK gibi kurup gider ya da DOĞAN GÜNEŞ gibi biter bitmez otomatik olarak yeniden başlar. Bu programları durdurmak için BREAK'e basın.



Program durana kadar ve BREAK yazısı belirene kadar tuşa basmaya devam edin. Yeniden başlatmak için RIRUN ve ENTER tuşlarına basmak yeter.

## Hatalar nasıl düzeltilir?

Yanlış bir tuşa basarsanız ya da SHIFT veya EXTEND MODE tuşlarına düzgün basamıyorsanız, endselenmeyin. DELETE tuşuna basıldığında, son komut işaret, harf ya da rakam silinir. Daha önceden de silmek isterseniz DELETE'e basmaya devam edin.

## DELETE

Eğer bir satırda hata yapmış ve ENTER tuşuna da basmışsanız, ekranda yanlış silinen bir satır izlenir. Bu, hataya gösterir. DELETE'e basarak hata yaptığınız noktaya kadar gidin, oradan sonra satırı tamamlayıp, ENTER tuşuna basın.

Eğer belleğe hatalı satır sokarsanız, program durur, ekranın en altında hatalı satır gösteren bir rakam belirir. Bu satırın tümünü yeniden yazıp, sırasıyla ENTER, RIRUN ve ENTER'e basın. Artık programın çalışması gerekir.

## ÇOKGEN

```

10 BORDER 1:POWER 6:CLS
20 INPUT "X,Y=";X,Y
30 FOR I=1 TO 20 STEP 2
40 LET X=X+1:LET Y=Y+1
50 FOR J=1 TO 20 STEP 2
60 LET X=X+1:LET Y=Y+1
70 LET X=X+1:LET Y=Y+1
80 LET X=X+1:LET Y=Y+1
90 LET X=X+1:LET Y=Y+1
100 NEXT J
110 NEXT I
120 NEXT X

```



Boş bir ekran göreceksiniz. Önce 6'ya sonra ENTER'e basın. Ekranda bir altıgen belirir. Bitince programı yeniden başlatın ve kaç kenarlı bir şekil istiyorsanız, o rakama basın.

Dene  
Satır 20'de 2'nin yerine başka bir rakam yazın. Bu rakam ne kadar büyük, çokgenlerin arası ne kadar açık olursa, desen o kadar hızlı olur.

## YILDIZLAR VE ÇUBUKLAR

```

10 ZIN 2
20 POWER 7
30 CLS
40 FOR X=28 TO 148 STEP 20
50 FOR Y=28 TO 31
60 PLOT X,Y:DRAW 215,0
70 NEXT Y
80 NEXT X
90 PLOT 55,28:DRAW 0,131
100 PLOT 215,28:DRAW 0,131
110 DRAW 1
120 FOR X=28 TO 8 STEP 2
130 DRAW AT X,2:*****
140 DRAW AT X,2:*****
150 DRAW AT X,2:*****
160 NEXT X
170 PRINT AT X,2:*****

```



Ekranda Amerika Birleşik Devletleri bayrağı belirir.

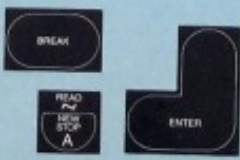
## Dene

Bayrağın renklerini değiştirin. Çubukların rengi satır 10. yıldızların satır 120. yıldızların fonu ise satır 110'dur.

## Yeni bir program nasıl başlatılır?

Bir programla iştin bilince, yerine yepyeni bir program koymak istiyorsanız, programın sona ermesini bekleyin ya da BREAK'e basarak durdurun.

Bir programı bilgisayarın belleğinden silmek için iki yol vardır. Bir A(NEW) ve ardından ENTER tuşlarına basmaktır. Ekrana bir an için kararar, sonra yeniden Copyright mesajı belirir.



Ya da daha kolay, sadece RESET tuşuna basabilirsiniz. Bu da Spectrum'u ceryanını kesip kapatmakla aynı sonucu verir.

## DOĞAN GÜNEŞ

```

10 BORDER 0:CLS
20 INPUT "X,Y=";X,Y
30 POWER 10:DRAW 0
40 CLS
50 LET X=X+1:LET Y=Y+1
60 FOR X=10 TO 274 STEP 2
70 PLOT X,Y:DRAW 135,0
80 DRAW 135,0
90 DRAW 135,0
100 DRAW 135,0
110 DRAW 135,0
120 DRAW 135,0
130 DRAW 135,0
140 DRAW 135,0
150 DRAW 135,0
160 DRAW 135,0
170 DRAW 135,0
180 DRAW 135,0
190 DRAW 135,0
200 DRAW 135,0
210 DRAW 135,0
220 DRAW 135,0
230 DRAW 135,0
240 DRAW 135,0
250 DRAW 135,0
260 DRAW 135,0
270 DRAW 135,0
280 DRAW 135,0
290 DRAW 135,0
300 DRAW 135,0
310 DRAW 135,0
320 DRAW 135,0
330 DRAW 135,0
340 DRAW 135,0
350 DRAW 135,0
360 DRAW 135,0
370 DRAW 135,0
380 DRAW 135,0
390 DRAW 135,0
400 DRAW 135,0
410 DRAW 135,0
420 DRAW 135,0
430 DRAW 135,0
440 DRAW 135,0
450 DRAW 135,0
460 DRAW 135,0
470 DRAW 135,0
480 DRAW 135,0
490 DRAW 135,0
500 DRAW 135,0
510 DRAW 135,0
520 DRAW 135,0
530 DRAW 135,0
540 DRAW 135,0
550 DRAW 135,0
560 DRAW 135,0
570 DRAW 135,0
580 DRAW 135,0
590 DRAW 135,0
600 DRAW 135,0
610 DRAW 135,0
620 DRAW 135,0
630 DRAW 135,0
640 DRAW 135,0
650 DRAW 135,0
660 DRAW 135,0
670 DRAW 135,0
680 DRAW 135,0
690 DRAW 135,0
700 DRAW 135,0
710 DRAW 135,0
720 DRAW 135,0
730 DRAW 135,0
740 DRAW 135,0
750 DRAW 135,0
760 DRAW 135,0
770 DRAW 135,0
780 DRAW 135,0
790 DRAW 135,0
800 DRAW 135,0
810 DRAW 135,0
820 DRAW 135,0
830 DRAW 135,0
840 DRAW 135,0
850 DRAW 135,0
860 DRAW 135,0
870 DRAW 135,0
880 DRAW 135,0
890 DRAW 135,0
900 DRAW 135,0
910 DRAW 135,0
920 DRAW 135,0
930 DRAW 135,0
940 DRAW 135,0
950 DRAW 135,0
960 DRAW 135,0
970 DRAW 135,0
980 DRAW 135,0
990 DRAW 135,0
1000 DRAW 135,0

```



Her birkaç saniyede bir, değişik renklerde doğan güneş andıran bir görüntü belirir. Eğer ekrandaki görüntü kaybolursa bekleyin. Ufuktan yeni bir güneş doğacaktır.

## Dene

Satır 210'daki 200'un yerine bir başka rakam yazarsanız, her gündüğünün ekranda kalış süresini değiştirebilirsiniz. 200, 4 saniyeye eşittir.

## Şimdi ne yapalım?

Çeşitli seçenekler var. Bu programlardan herhangi birini ileride yeniden kullanmak üzere saklamak istiyorsanız, bir kasete kaydedebilirsiniz. Bunu nasıl yapacağınızı öğrenmek için, 38. sayfayı açın. "Programlarınızı nasıl saklayabilirsiniz?" bölümünü okuyun.

Spectrum'unuzla denemeleri sürdürmek istiyorsanız, "Programlamaya başlayın" başlıklı ikinci bölümde programcılık hakkında bilgi bulabilirsiniz. Şu ana kadar, nasıl yapıldığını anlamadan bazı programları denediniz. Spectrum'un programlamasının bazı özellikleri 2. bölümde anlatılacak.

Satır aldığınız bazı bilgisayar oyunlarını denemek isterseniz, o zaman sayfayı çevirip "Hazır yazılımlar nasıl kullanılır"ı okuyun.



## HAZIR YAZILIM NASIL KULLANILIR?

Spectrum'a bir program verirken, tuşlara basınca bir dizi elektronik olarak kodlanmış sinyal üretirsiniz. Bu kodlu sinyaller, Spectrum'un belleğine girer. Bellek bunları, program işleme konusuna bilgisayar kullanabilisin diye saklar. Bu kodlu sinyaller, siz (örneğin NEW ya da RESET tuşuna basarak) bunları silmedikçe ya da Spectrum'u kapatmadıkça, bellekte kalmaya devam eder.

Ancak Spectrum'unuzu her kullanmak istediğinizde yeni bir program vermeniz zorunlu değil. Bunun yerine, bilgisayarınıza doğrudan, otomatik olarak girecek programlar içeren hazır yazılımlar da satın alabilirsiniz. Hazır yazılımlar, sizi Spectrum'u her kullanmak istediğinizde program yazma derdinden kurtarmakla kalmaz, ama aynı zamanda siz kendiniz yazmaya kalksanız günlerinizi, hatta haftalarınızı alacak hazır programlardan oluşan bir kitaplık oluşturmanızı da sağlar. Yazılım imalatçıları, en iyi programcılar tarafından yazılmış çok çeşitli programlar sunarlar ve bunların büyük bir kısmı Spectrum'da kullanılabilir. Ne türden programlardan hoşlanıp, kullanabileceğinizi saptamak için Sinclair Spectrum Software Catalogue (Yazılım Kataloğu) na bakabilirsiniz. Bundan sonra da, gereksinim duyduğunuz, istediğiniz programı işleme koyabilirsiniz.

### Program Spectrum'a nasıl yüklenir?

Bir yazılım kasetindeki kodlu sinyaller, saniyede 1500 adet hızıyla kaydedilmiş alçak ve yüksek bazı seslerden oluşur. Bu yazılım kasetini, teypte çalınca teyp programı oluşturan sesleri yansıtır. Teypi Spectrum'a bağladığınızda, bu kodlu sinyaller doğrudan Spectrum'un belleğine gider. Buna program yükleme denir. Bu iki sayfada, teypin nasıl bağlanacağını göreceksiniz, 14 ve 15. sayfalarda ise bunun nasıl kullanılacağını bulacaksınız.

### Yazılım ilişkin sorular ve yanıtlar

#### "Yazılım" ne demektir?

Yazılım (İngilizce Software), bilgisayarları çalıştırmak için verilen programların genel adı. Donanım ise, (İngilizce Hardware) kullanılan araçlar - bilgisayar ve bu işlemlerde yararlanılan diğer gereçlerdir.

#### Yazılım niye kasetli teyple yapılıyor?

Kasetli teypinin kullanımı kolaydır, başkaca bir özel aygıt gerektirmez. Bu tür yazılımın yüklenmesi için gereken tek şey, ucuz bir kasetli teyptir.

#### Kayda alınmış bir program nasıl bir şey?

Birini, teypi Spectrum'a bağlamadan çalın. Tiz bir düdük sesi duyacaksınız. Bu bilgisayara gidecek yerde oparölöre ulaşan kodlu sinyallerin çıkardığı sesler. Sinyaller teypten Spectrum'a o kadar hızlı aktarılır ki, sesleri teker teker ayırt etmek olanaksızdır.

#### Başka türlü yazılımlar da var mı?

Evet. Kaset yerine ROM kartuşlarda kayıtlı programlar da bulabilirsiniz. Bu kartuşlar, Spectrum'unuzun arkasındaki bir yuvaya girer. ROM kartuşları program, hiç beklemeden anında yüklenir.

Ayrıca Microdrive kartuşlarında da programlar vardır. Bunlarda da programlar, kaset gibi manyetik olarak kaydedilmiştir. Bir kartuşta birkaç program birden olabilir ve kasetin aksine bir programın yüklenmesi dakikalarca değil birkaç saniye sürer. Microdrive kartuşları, Microdrive biriminde çalınabilir. (Bak s.46)

#### En iyi kasetli teyp hangisi?

Spectrum pahalı olmayan herhangi bir portatif kasetli teyple çalışabilir, ama pili değil cırcıyan bağlantılı teypler yeğlenmelidir. Teypin bir ses ayar düğmesi olmalıdır, ama ton düğmesi zorunlu değildir. Ayrıca özel bilgisayar kasetli teypleri de vardır. Bunlar, programların daha güvenli yüklenmesini ve saklanması sağlamaya özenle özel olarak yapılmıştır.

Bir müzik setinin parçasını oluşturan kasetli teypin bilgisayara bağlanması pek kolay olmayacaktır. Ayrıca bu tür araçların ses çıkışları genellikle Spectrum için yeterince güçlü ses sinyali verememektedir.

#### Teypte kaydedilmiş programların özel bakımı gerekir mi?

Diğer manyetik kayıtlar gibi, kasete kayıtlı program da güçlü mıknatıslardan uzak tutulmalıdır. Bu yüzden kasetleri güçlü bir elektrik akımı kullanan hiçbir şeyin yakınında saklamayın. Yazılım kasetlerinin de, tozdan oldukça uzak tutulması gerekir.

#### Her türden yazılım kullanılabilir mi?

Hayır, sadece ZX Spectrum ya da ZX Spectrum+ için yazılmış yazılımlar kullanılabilir.

### Kasetli teyp nasıl bağlanır?

Spectrum'unuzla beraber gelen, teyp kablosu, bunu teypte bağlamak içindir. Bu bağlantının her iki ucunda da ikiye tane küçük fiş vardır. Kasetli teypinizi Spectrum'un yanına yerleştirin ve kabloyu şekilde gösterildiği gibi bağlayın.

Bu işlemi yaparken Spectrum ya da teypin çalışır durumda olup olmaması önemli değil, ama gene teypi açar ya da kaparken içinde kaset bırakmamak iyi olur. Bu, kasete kayıtlı programların korunması açısından yararlıdır.

### Bağlantıları doğru yapın



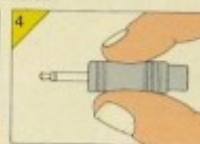
1 Dört fişten birini Spectrum'un arkasındaki EAR girişine sokun



2 Öbür uçtaki aynı renkli fişi teypin, eğer varsa EAR girişine sokun



3 Eğer teypin EAR girişi yoksa, fişi kulaklık girişine, o da yoksa diğer oparölör girişine sokun



4 Eğer bu fişler, teyp girişine uymuyorsa, elektrikli denetçi ya adaptör ya da uygun fişli bir kablo bulmanız gerekir. Spectrum'un EAR girişi 3.5mm'lik jak-fiş ve 1 voltluk bir girdi sinyali için yapılmıştır.

#### EAR ve MIC girişleri

Programları yüklerken, burada da gördüğümüz gibi EAR ve MIC girişleri bağlı durabilir. Ama bir program saklamaya hazırlanıyorsanız (Bak s.38) EAR kablosunu sökmeniz gerekir.

#### Yazılım için öğütler

■ Spectrum'un teyp kablosunda, bilgisayarla teyp arasındaki bağlantılar karışmasın diye, farklı renklerde fişler kullanılmıştır. Teypinizi Spectrum'la kullandığınızda, düzeni değiştirmeyin, bir rengi EAR, bir rengi MIC girişleri için kullanın.

■ Bazı kasetli teypler çevrelerindeki elektrikli araçlardan etkilenirler. Bundan bilgisayar ile teyp arasında gidecek sinyaller de etkilenir ve bu yüzden programlar düzgün yüklenemeyebilir. Eğer teypiniz düzgün çalışmazsa o zaman teypi televizyon ya da bilgisayarın yanından uzaklaştırın.





## PROGRAM NASIL YÜKLENİR?

Kasetli teypi Spectrum'unuza bağladığınızda göre, program yükleyip, işleme koymaya hazırsınız demektir. Ya bir hazır yazılım kaseti ya da kendi programlarınızı içeren bir kaset kullanabilirsiniz. Her iki durumda da yapacağınız aynı işlemdir.

1 Kaseti takın ve en başa sarın.

2 Kasetli teypin ses ve ton düğmelerini gerekli düzeye getirin. Ses düğmesi üçte iki oranında açık olmalı, diğer ton düğmesi varsa, tize getirilmelidir.

3 J'ye basın, ekranda LOAD yazısı gözükmelidir. Sonra programın adını tırmak işaretleri içinde yazın. Örneğin LOAD "Prog 1"



4 ENTER'e basın. Ekran silinecektir.

5 Teypi çalıştırın. Ekranın çevresi kırmızı veya mavi olacak ya da kırmızı ve mavi yarıp sönecektir. Bu Spectrum'un program aradığını gösterir.



6 Birkaç saniye sonra, ekranın çevresindeki kırmızı mavi çizgiler aşağı-yukarı hareket etmeye başlar. Bu Spectrum'un sinyali almaya başladığını belirtir.

### Yazılım yükleme öğütleri

İşte size yükleme zamanı kazandıracak bazı öğütler:

1 Bütün kasetlere, aradığınız programı kolayca bulabileceğiniz biçimde etiket yapıştırın. Eğer bir kasette birden fazla program varsa, etikete sırasıyla bunların hepsini yazın. Programların adını tam bilgisayarın tanıyacağı biçimde yazmayı unutmayın.

Program Adı	Yazılım
1. Program	1. Yazılım
2. Program	2. Yazılım
3. Program	3. Yazılım
4. Program	4. Yazılım
5. Program	5. Yazılım

2 Kasetli teypin sayacı varsa, bunu üzerinde birden fazla program bulunan kasetlerde aradığınızı hızla

bulmak için kullanın. Kasetin başlangıcında sayacı sıfıra getirin. LOAD komutu verin, ardından o kasette bulunmayan bir programın adını tırmak işaretleri içinde yazın. Kaseti çalıştırınızda, Spectrum kasetteki bütün programların adını teker teker ekrana yazacak, ama yüklemeyecektir. Programın adının yanına sayacıdaki rakamı yazın. İleride hangi programın nerede olduğunu kolayca bulabilirsiniz.

7 Ekranda Program sözcüğü, ardından programın adı veya Bytes yazısı, bundan sonra da bir harf ya da rakam görünür. Bu bilgisayarın istenen programı başıyla bulduğunu işaretidir.



8 Bilgisayar programı yüklemek üzere beklerken, kırmızı ve mavi çizgiler yeniden belirlir.

9 Ekranın çevresinde san ve mavi çizgiler belirlir. Bu da Spectrum'un programı yüklemekte olduğunu gösterir. Eğer program çok uzunsa, yükleme birkaç dakika sürebilir.



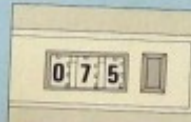
10 Eğer program bölümlere ayrılmışsa, 7, 8 ve 9. adımlar bir veya birkaç kez daha tekrarlanabilir.

11 Program, yüklenir yüklenmez otomatik olarak başlayabilir. Teypi durdurmayı unutmayın.

12 Eğer program yüklenir yüklenmez otomatik olarak başlamazsa, ekran silinir, OK, 0-1 yazısı belirlir. Teypi durdurun.

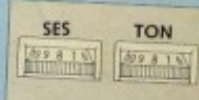


13 R(RUN) ve ENTER'e basın, program başlar.



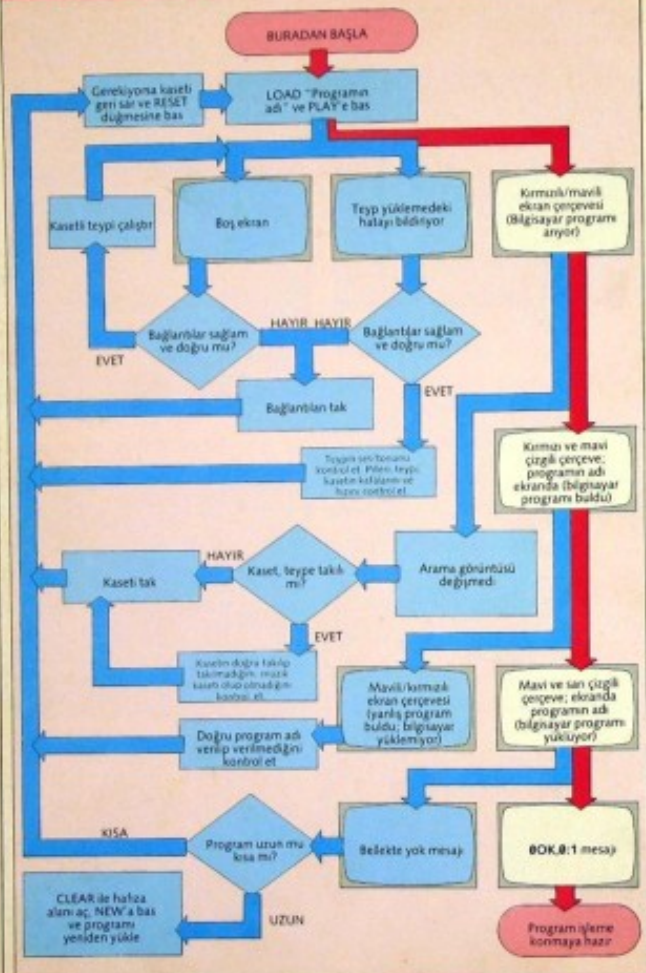
3 Eğer kaset doğru programı bulmuşsa ya da programın adını bilmiyorsanız, LOAD komutundan sonra tırmak içinde

programın adını yazacak yerde, LOAD " " yazın. İki tırmak arasında hiç boşluk bırakmayın. Bu durumda Spectrum'unuz bulduğu ilk programı yükleyecektir. Eğer gördüğünüz program adı, istediğiniz değilse, BREAK'e basın, kaseti sanp gene deneyin.



4 Spectrum'unuzun yüklenmesine yardımcı olacak ses ve ton düzeyine

dikkat edin. Bunları, teypinizi yüklemeye önce yukarıda gösterildiği gibi ayarlayın.



# PROGRAMLAMAYA BASLAYALIM

Bu bölüm, ZX Spectrum+'da program yazmaya bir giriştir. Klavyeyi tanıtarak, Spectrum'unuzu nasıl kullanabileceğinizi anlatmaktadır. Böylece Spectrum'unuzu nasıl çalıştırmaya başlayabileceğinizi göreceksiniz. Bu bölümde deneyeceğiniz kısa programlar, kendi programlarınızı yazarken bilgisayarınızın kapasitesini sonuna kadar kullanabilmeniz için Spectrum'un kendine özgü özellikleri üzerinde yoğunlaştırılmıştır.



# KLAVYE – BİLGİSAYARINIZIN KOMUTA KONTROL TABLOSU

ZX Spectrum'un kendine özgü bir dili vardır – BASIC diye bilinen bilgisayar dili. Talimatlarınıza uyması için, Spectrum'u BASIC dili konuşarak programlamamız gerekir. Bunu, Spectrum'un klavyesini kullanarak yapacaksınız. Ayrıca bu klavye, bilgisayar sizin programınızı işlerken, bilgisayarın kontrol etmenizi de sağlar.

**GRAPH**  
Bu tuş, 1 ile 8 arasındaki şekilleri ve grafik karakterleri seçmede kullanılır. Bu tuşa basıp, sonra da CAPS SHIFT tuşuna basarak veya basmadan bir rakam tuşuna basarsanız, ekranda bir grafik karakter belirecektir.

**NEW**  
Bu tuş, bilgisayarın BASIC dilindeki belleğini temizler, bellekteki bütün programları siler.

**DELETE**  
Yanlış bir tuşa basmış, bir komut, harf, rakam veya işaretli silmek istiyorsanız, bu tuşa basın – bak sayfa 10.

**EDIT**  
Bu tuş, bütün bir programı yeniden yazmak için bunun bir satırının düzeltilmesini sağlar – bak sayfa 21.

**EXTEND MODE**  
Bu tuşun yardımıyla, tuşların üstündeki komutları verebilirsiniz. Bunun ardından SYMBOL SHIFT'i kullanırsanız, üstündeki işaret ya da komutu elde edersiniz – bak sayfa 20-21.



**CAPS SHIFT**  
Bu tuşa basarken, bir harf tuşuna basarsanız, büyük harf elde edersiniz. Yan yana birçok büyük harf kullanacaksanız önce CAPS LOCK'a basın.

**CAPS LOCK**  
Eğer sürekli büyük harf yazmak istiyorsanız, harf tuşlarına basmaya başlamadan buna basarsınız. Büyük harflerle yazdıktan sonra, tekrar buna basıp küçük harflere dönebilirsiniz.

**BEEP**  
Bu tuş, Spectrum'un ses vermesini denetleyen komutunu elde edebilirsiniz.

**TRUE VIDEO ve INV VIDEO**  
Bu tuşlar, program satırlarına, normal ya da ters (dişi) renkler elde etmek üzere kontrol kodlarının yerleştirilmesi için kullanılır.

## Tuşlar ve Komutlar

Komutlar, BASIC dilinde bilgisayara belli bir şey yapması talimatı veren özel sözcüklerdir – PRINT, INPUT gibi. Bilgisayarların çoğunda bu komutları, tipki daktiloda yazarak gibi harfleri teker teker yazmak zorundasınız ve her sözcüğün imlasının da tam doğru olması gerekir. Ancak Spectrum'da bir tek tuşa basarak, bütün komutun ekranda

görünmesini sağlayabilirsiniz.

Sinclair BASIC'te 80 komut vardır. Bunları 36 tuşa (26'sı harf, 10'u rakam) basarak verebilirsiniz. Spectrum bu kadar çok çeşitlilik gösteren BASIC komutları kullandığından, birçok tuşa sadece bir değil, bilgisayarın tanıyacağı çok çeşitli komutlar verebilirsiniz. Aslında birçok tuş size komutların yanısıra harf, rakam, işaret, hatta şekil (grafik karakterler) sağlar. Bütün bunları programlarınızda kullanabilirsiniz.

**Renk tuşları**  
Bu altı tuş, ekrandaki renkleri kontrol eden komutları verir.

## Komutların ve İşaretlerin seçimi

Spectrum'un klavyesinde çok kullanacağınız iki tuş var: EXTEND MODE ve SYMBOL SHIFT. Bunlarla, diğer tuşlardaki komut ya da işaretlerden hangisinin ekranda görünmesini istediğinizi seçebilirsiniz. Bu tuşları nasıl kullanabileceğinizi size daha önce 8. sayfada kısaca anlattık. Şimdi klavyeyi biraz daha tanıdıktan sonra, önümüzdeki iki sayfada, bilgisayarın klavyesinde gördüğünüz herhangi bir şeyi nasıl seçebileceğinizi anlatacağız. Bir defa bunu öğrendiğiniz mi, programlarınızı yazmaya başlayabilirsiniz.

**Rakam tuşları**  
Bunlar, rakamları yazmanın yanısıra, gösterilen renklerin programlara yerleştirilmesi için kontrol kodlarını da verir. (Bak s. 33). 8'de 4'ten 0'a kadarki tuşların hemen üstünde yer alan bu komutlar ancak ZX Microdrive'larla kullanılabilir.

**BREAK**  
Bu tuş, işleme konan bir programı durdurur. Program bilgisayarın belleğinden silinir.

**ENTER**  
Bu tuşa basınca, bir program satırını Spectrum'un belleğine girer. Bu tuş, ayrıca çok sık, bir program sırasında bilgisayara bilgi beslemeye de kullanılır.

**SYMBOL SHIFT**  
Bir tuşun üstündeki ikinci bölümden işaret ya da alt komutu kullanmak isterseniz, bu tuşa SYMBOL SHIFT'i birlikte basarsınız gerekir. Eğer SYMBOL SHIFT'e, EXTEND MODE'dan sonra basarsanız, o zaman bilgisayar, ikinci bölümün hemen üstündeki komuta uyar – bak sayfa 20-21.

**Boşluk tuşu**  
Tipki daktiloda olduğu gibi bu tuşa dokununca bir ara elde edilir.

**Göstergeç kontrolü**  
Bu tuşlara basarak göstergecin (İngilizceci cursor) oların yönünde hareket etmesini sağlayabilirsiniz. Bu tuşlar, programlarda genellikle şekillerin ekrandaki hareketlerini denetlemek için kullanılır. Ayrıca program üzerindeki düzeltmelerde de yararlı olur.

## TUŞLAR NASIL ÇALIŞIR?

ZK Spectrum'un tuşlarından herhangi biriyle altı kadar değişik komut, harf, rakam ya da işaret yazabilirsiniz. Spectrum'un özelliklerinden birini iyice öğrendiğiniz takdirde klavyedeki herhangi bir harf ya da komutu seçmek pek o kadar karışık gelmeyecektir. Bir tuşa bastığınızda, ekranda görünecek olan, o anda bilgisayarınızın hangi modda olduğuna bağlıdır. Farklı karakterleri gibi çok farklı türde bilgileri iletmenizi sağlar. Bunun yaranı şudur: Siz klavyeyi kullanırken, aslında Spectrum, talimat ve bilgileri sırasıyla verebilmeniz için size klavyedeki modları seçmenize yardımcı olur. Bu iki sayfada bu modların ne işe yaradığını göreceksiniz.

### Komut modu

Spectrum'unuzu çalıştırın ya da Copyright mesajını göreceğiniz biçimde düzenleyin ENTER'e basın. Sol alt köşede yanıp sönen bir K harfi belirir. Bu size basacağınız ilk tuşun neresiye yansıyacağını gösterir. K ise bilgisayarın komut modunda olduğunu belirtir. Herhangi bir tuşa basın. Tuşun çıktığı bölümdenki üst komutu ekranda belirir. Örneğin Q'ya basarsanız PLOT sözcüğü belirir. DELETE tuşuna basarak komutu silin ve diğer tuşları deneyin. Rakam tuşları size rakam verecek, ancak bir harf tuşuna basarsanız, çıkıntılı bölümdenki üst komutu göreceksiniz.

Gene DELETE'e basarak, K göstergecinin belirmesini sağlayın. Şimdi de SYMBOL SHIFT tuşlarından birine basın, elinizi orada tutarken, bir başka tuşa daha dokununuz. Bu defa çıkıntılı bölümdenki harfin hemen üstündeki komut ya da işaret belirir. Rakam tuşunda ise, çıkıntılı bölümün sağ üst köşesindeki işaret belirir. Yani, Komut modu tuşun çıkıntılı bölümü ile ilişkilidir.

### Komut, simge ya da karakter nasıl seçilir?

Aşağıda, bir harf ya da rakam tuşundaki herhangi bir komut, işaret ya da karakterin nasıl seçilebileceğini göreceksiniz. Bir

tuştaki herhangi bir işlevi seçerken, bunun tuşun neresinde olduğuna dikkat edin ve aşağıdaki iki örnekten yararlanarak, hangi tuşlara basarak

neyi, nasıl bulabileceğinizi belirleyin. Daha önce ekrandaki göstergece bakıp bilgisayarın hangi modda olduğunu saptayın.

#### Harf tuşu

BIN BRIGHT BORDER B *	Komut (K) modu
Sadece tuş	BORDER
ve tuş	
Genişletilmiş (E) mod	
sonra sadece tuş	BIN
ve tuş	
Harf (L) modu	
Sadece tuş	b
ve tuş	B
ve tuş	
Büyük harf (C) modu	
sonra sadece tuş	B
ve tuş	
Grafik (G) modu	
sonra A ile U arasındaki tuşlar kullanılarak belirlenecek grafik	

#### Sayı tuşu

MIGNA LINE 3 #	Komut (K) modu
Sadece tuş	3
ve tuş	#
Genişletilmiş (E) mod	
sonra sadece tuş	mor
ve tuş	LINE
Harf (L) modu	
Sadece tuş	3
ve tuş	#
Büyük harf (C) modu	
sonra sadece tuş	3
ve tuş	#
Grafik (G) modu	
sonra sadece tuş	
ve tuş	

### Harf ve büyük harf modları

Bilgisayar komut modundayken, bir komut ya da işaret yazdıktan sonra otomatik olarak göstergeci L'ye çevirir. Artık harf moduna geçmiştir. Herhangi bir harf tuşuna basın, bir küçük harf belirir. Rakam tuşuna basın, rakam görürsünüz. Büyük harf yazmak isterseniz, CAPS SHIFT'e basıp bunu tutarken harfe basın.

Eğer hep büyük harf yazacaksanız, önce CAPS LOCK'a basın. Göstergeç C'ye döner. Spectrum'unuz büyük harf moduna geçti, artık hangi harfe basarsanız büyük harf yazarsınız. Rakamlar büyük harf modunda da yazılabilir. Harf (L) moduna dönmek için CAPS LOCK'a bir kez daha basın.

### Genişletilmiş mod

Bu modun adı genişletilmiş mod ve bunu EXTEND MODE'a basarak elde edersiniz. Göstergeç E'ye döner. Herhangi bir harf tuşuna bastığınızda, çıkıntılı bölümün üzerinde bulunan iki komuttan üsttekini

elde edersiniz. Örneğin B'ye basarsanız BIN çıkar. Altındaki komutu ya da işareti elde etmek için ise SYMBOL SHIFT tuşlarından birine basıp tutun ve harf tuşuna dokununuz. B tuşunda örneğin BRIGHT yazısı çıkar. Yani genişletilmiş modda bir tuşa (ya da EXTEND MODE'a) basarsanız, bilgisayar otomatik olarak harf ya da büyük harf moduna geçer.

### Grafik Modu

Beşinci mod, grafik modudur ve GRAPHIC tuşuna basarak geçer. Göstergeç G'ye döner. 1'den 8'e kadar rakamlara basıp bunların üzerindeki grafik karakterlere bakın. Şimdi ise CAPS SHIFT'e basarak sonra 1 ile 8 arasında herhangi bir sayıya basın. Aynı şekli yeniden çıkacaktır ama bu defa siyah ile beyaz yer değiştirmiştir. Grafik modundan çıkmak için mutlaka GRAPH tuşuna yeniden basmanız gerekir, çünkü bilgisayar bu moddan otomatik olarak çıkmaz.

### Spectrum'da düzeltme

Spectrum'a talimat verdiğinizde ya da kendi programınızı yazmaya başladığınızda, komutlarda ya da program satırlarında düzeltmeler yapmak veya bunları değiştirmek isteyebilirsiniz. Bu oldukça kolaydır.

#### Hatanın düzeltilmesi

BASIC'de hata bir satır ya da komut vermeye kalkarsanız, Spectrum bu hatanın önünde bir ? işaretini yakıp söndürmeye başlar. Hatayı düzeltmek için, göstergeci hatanın sağına götürebilmek üzere, göstergeç kontrol tuşlarından sağdakine ya da soldakine sürekli basın. Ardından ya DELETE'e basarak hatayı yok edin ya da gereken komut, harf, rakam ve simgeyi yazın. Ardından ENTER'e basın.

Örneğin bilgisayarın 7 ile 8'i çarpmasını istiyorsanız ama + elde etmek için SYMBOL SHIFT'e basmayı unuttunuz. Bunun yerine

#### PRINT 7\*8

çıkars. Spectrum bu komuta uyamaz, bu nedenle siz ENTER'e basar basmaz hata yaptığınız ?'nin önünde bir soru işareti yanıp sönmeye başlar. Bütün

yapacağınız, göstergeci hatanın sağına götürüp, ardından DELETE'e basıp b'yi yoketmek. Bundan sonra SYMBOL SHIFT ve B'ye basarak + elde edin ve ENTER'e basarak bilgisayarın doğru talimata uymasını sağlayın. Göstergeci yensiden satırın sonuna kadar götürmeniz gerekmez. Spectrum talimatı uygular ve sonucu verir.

#### Bir program satırını nasıl düzeltilir?

Bir program yazarken, sıralama



denilen, birbirini izleyen numaralarla bir dizi satır ardında koyarsınız. Eğer, program yazdıktan sonra, buna K (LIST) ve ENTER tuşlarına basarak "sıralarsanız", programın satırlarından birinin karşısına > işaretini göreceksiniz. Eğer görüyorsanız, göstergeci aşağı yukarı kontrol eden tuşlara basın. EDIT tuşuna basarsanız, bu işaretin karşısındaki satırın eşi ekranın altında bir daha çıkar. Bunu daha önce anlattığımız gibi göstergeç kontrol ve DELETE tuşlarını kullanarak "teğetirebilirsiniz. ENTER'e basınca da o satır eski yerine döner. Başka bir satırı daha düzeltmek istiyorsanız, > işaretini istediğiniz satırın karşısına gelecek şekilde yukarı aşağı oynatın ve EDIT'e basın. Eğer bu uzun sürecektir, sırasıyla LIST, istenen satırın numarası ve EDIT'e basın. Her iki halde de istediğiniz satır ekranın en altında belirir. Bunda düzeltme yapabilirsiniz. Programın bütün bu satırını silmek istiyorsanız, sadece satırın numarasını yazıp ENTER'e basın. Eğer bir hata içeren bir programı işleme koyduysanız, ekranda bunun haberi görürsünüz. Bu 74. sayfada açıklanacaktır.



# TELEVİZYON HESAP MAKİNASI

ZX Spectrum+ son derece hızlı ve doğru hesap yapabilir. Bütün gereken bir kaç rakam ve bu rakamlarla ne yapılmasını istediğinizi gösteren +, - gibi bazı işaretlerdir.

Önce şu talimatı verin (+ işaretini K tuşunda bulacaksınız):

## PRINT 6+2

Bu bir komuttur. ENTER'e basınca komut kaybolur, ekranda cevap, 8 görünür.

Spectrum'unuz, hesaplamada, aritmetik işlemler denilen beş işaret kullanır. Bunların her birinin ne yaptığını aşağıdaki tablo da görebilirsiniz. Bunların hepsini PRINT ile aynı şekilde kullanabilirsiniz.

PRINT 6+2 gibi komutlar vererek, Spectrum'unuzu hesap makinasına dönüştürebilirsiniz. Ayrıca sıradan bir hesap makinasının yapmadığı birçok şey yapabilir. Herşeyden önce, hem hesabı hem sonucunu aynı anda gösterir. Şu komutu verin:

## PRINT "6+2="; 6+2

Bilgisayar bunu şöyle yanıtlar:

6+2=8

Olan şu: PRINT, tırnak işaretleri (") arasındaki herşeyin ekranda görünmesini sağlar, bu nedenle 6+2= görünür. Tırnak işaretleri arasındaki işaretlere dizi denir. Noktalı virgül Spectrum'a, eşit işaretinden hemen sonra sonucu iletmesi talimatını verir.

## Spectrum'un hesaplama işaretleri

Aşağıdaki işaretler ya da "aritmetik işlemler", Spectrum tarafından matematik işlemler için kullanılır. Dikkat edin bilgisayar x ya + işaretlerini kullanmaz.

İşaret	Tuş	İşlevi	Örnek
+	K	İki rakamı toplar	8+2=10
-	J	Bir rakamı diğerinden çıkarır	8-2=6
*	B	İki rakamı çarpır	8*2=16
/	V	Bir rakamı diğerine böler	8/2=4
↑	H	Birinci rakamı diğerinin üstüne çıkarır	8↑2=64

## İlk programınız

Bir komut yerine getirilince, Spectrum'unuz bunu unutur. Bilgisayarın, hesabı tekrarlamasını istiyorsanız, bunu program olarak yazabilirsiniz. Şu talimatı yazıp ENTER'e basın:

## 10 PRINT 6+2

Bu defa talimata derhal uymaz. Sadece talimatı ekrana yansıtır. R(RUN) ve ENTER'e basın. O zaman sonuç, 8, ekranda belirir.

Bütün talimat, şimdi bir bilgisayar talimatı haline gelmiştir. Başına rakam yazarak, bu talimatı Spectrum'unuzun belleğine aktardınız, ama emir alana kadar işleme koymamasını söylediniz. Bu talimata artık komut değil cümle denir ve bir programın numaralanmış bir satırını oluşturur. Programdaki cümleler rakam satır sırasına göre işleme konur ve bunlar gerektiğinde aralarına satır eklenebilir diye daima onlu sayılarla başlar.

Artık Spectrum'unuzu gerçekten çalıştırabilirsiniz. Şu programı verin. Her satır yazdıktan sonra ENTER'e basmayı unutmayın. Bitirince, R(RUN) ve ENTER'e basın. Programı işleme koyunca şuunu göreceksiniz:

## SAYI TABLOSU

10 LET A=1	20 PRINT A; " ";
20 LET A=A+1	40 GO TO 20

1	2	3	4	5	6	7	8	9	0
10	20	30	40	50	60	70	80	90	00
100	200	300	400	500	600	700	800	900	000
1000	2000	3000	4000	5000	6000	7000	8000	9000	0000
10000	20000	30000	40000	50000	60000	70000	80000	90000	00000
100000	200000	300000	400000	500000	600000	700000	800000	900000	000000
1000000	2000000	3000000	4000000	5000000	6000000	7000000	8000000	9000000	0000000
10000000	20000000	30000000	40000000	50000000	60000000	70000000	80000000	90000000	00000000
100000000	200000000	300000000	400000000	500000000	600000000	700000000	800000000	900000000	000000000
1000000000	2000000000	3000000000	4000000000	5000000000	6000000000	7000000000	8000000000	9000000000	0000000000

1'den 203'e kadar bütün rakamlar görülüyor. N, ara tuşu, STOP veya BREAK dışındaki herhangi bir tuşa basın. Bir dizi sayı daha göreceksiniz.

Bu program bir değişken kullanıyor. Bu olayda değişken, n. Değişken olarak

herhangi bir harf ya da sözcük de kullanılabilir. Burada n sayı anlamına geliyor. Değişkene program boyunca değişen bir değer verilir. 10. satırda LET komutu kullanarak bu değer 1 olarak verilmiş. 20. satırda bir boşluk bırakılmış. 30. satırda yine LET kullanılmış, ama bu kez değeri 1 arttırılmış. Bu nedenle n 2 olur. 40. satırda (tek) bir komut GOTO kullanılarak program 20 satıra döndürülmüş. Artık 20. satırda 2 rakamı vardır. Bu, bütün ekran rakamla dolana kadar döne döne tekrarlanır.

## Programın sayı istemesi nasıl sağlanır

BREAK'e basarak programı durdurun. Yeni satır yazın:

## 10 INPUT n

Bu satır eski programdaki 10. satırın yerini alır. İşleme koyduğunuzda, bilgisayar bir rakam vermenizi bekler. Herhangi bir rakama, sonra ENTER'e basın. Bundan sonra görünecek rakamlar, sizin yazdığınız rakam ile başlar çünkü INPUT n, n'nin değerini sizin yazacağınız rakama eşitler. INPUT bilgisayara bir program sırasında bilgi alma talimatı verir.

## Bir çarpım cetveli programlama

Eski programı temizlemek için RESET düğmesine basın ve yeni programı verin. Bu program Spectrum'a çarpım yaptırır. Herhangi bir rakama bastığınızda, ekranda bu rakam için çarpım cetveli verilir. N, BREAK, ara tuşu dışında herhangi bir tuşa basınca tablo devam eder. BREAK'e basıp, yeni bir çarpım cetveli daha yapın. İşte 3 ve 146 ile yapılabilecek çarpım cetvellennin programı

## ÇARPIM CETVELİ

10 LET A=1	20 PRINT A; " ";
20 LET A=A+1	40 GO TO 20

1	2	3	4	5	6	7	8	9	0
10	20	30	40	50	60	70	80	90	00
100	200	300	400	500	600	700	800	900	000
1000	2000	3000	4000	5000	6000	7000	8000	9000	0000
10000	20000	30000	40000	50000	60000	70000	80000	90000	00000
100000	200000	300000	400000	500000	600000	700000	800000	900000	000000
1000000	2000000	3000000	4000000	5000000	6000000	7000000	8000000	9000000	0000000
10000000	20000000	30000000	40000000	50000000	60000000	70000000	80000000	90000000	00000000
100000000	200000000	300000000	400000000	500000000	600000000	700000000	800000000	900000000	000000000
1000000000	2000000000	3000000000	4000000000	5000000000	6000000000	7000000000	8000000000	9000000000	0000000000



## Parantezlerin kullanımı

Zaman zaman hesaplarda parantez kullanmanız gerekecek. Şu iki komutu verin ve sonuçları karşılaştırın:

## PRINT 6+2/4

## PRINT (6+2)/4

İlkinin sonucu 6.5, ikincisinin ise 2. Bu iki farklı sonucun nedeni, bilgisayarın hesaplarda kullandığı bir öncelikler sistemine sahip olması. Önce / , sonra - veya + , en son olarak da \* veya = uygular. Ama daima parantez içindeki işleme öncelik verir. Bu nedenle, yukarıdaki ilk komutta, önce 2'yi 4'e böler, sonra sonuçla (0.5) 6'yi toplar. İkinci komutta ise, 6 ile 2'yi toplayıp, sonra 4'e böler.

## Spectrum'da noktalama işaretleri

Spectrum çok çeşitli noktalama işaretleri kullanır. Bunlar çok önemlidir, çünkü, bunların çoğu aynı zamanda bilgisayarın bir program satırını anlamasını ya da bir sonuç sağlamanın etkileyecek biçimde bilgisayara talimat vermeye de yarar.

- **Noktalı virgül** PRINT ile birlikte kullanınca, bilgisayara iki yanındaki öğelerin ekranda yan yana görünmesi talimatı verir.
- **İki nokta** Program satırındaki cümlelerden birinin sona erdiğini, bir diğerinin başlayacağını haber verir.
- **Tırnak** Tırnak işaretleri arasında yazılanlar, rakam ya da değişken olarak değil, bir ifade olarak değerlendirilir. Dört tırnak işaretleri ile başlayıp biter.
- **Virgül** PRINT ile birlikte kullanınca, bilgisayara izleyen öğeyi ya bir satırın ortasında, ya da yeni satırın başında ekrana yansıtmasını bildirir. Bunu birer ya da milyonlar hanesini ayırmak için kullanmayın.
- **Nokta** Ya bir ondalık işareti ya da noktaadır.

## RENKLER VE KULLANIMI

ZX Spectrum+ ile sekiz değişik renk elde edebilirsiniz ve her rengin değişik bir renk kodu vardır. Renkleri çerçeve rengi, mürekkep rengi ya da kağıt rengi olarak değişik biçimlerde kullanabilirsiniz.

### ZX Spectrum+ renk kodları

Aşağıdaki çizelge, Spectrum'un kullandığı renkleri ve kodları gösteriyor. Bu kodları ezberlemeniz gerekmez; rengi sağlayan tuşun üzerinde rengin adı yazılır. (Bu adlar komut değildir).

Rakam	Renk
0	Siyah
1	Mavi
2	Kırmızı
3	Yeşil
4	Yasıl
5	Açık mavi
6	Sarı
7	Beyaz

Ekranda göreceğiniz renk tonunu ise televizyon alıcınızın renk, kontrast ve tıpkı ayarları belirler. Unutmayın bunun için renkli televizyon gerekir.

### Spectrum'da renk kullanmanın üç yöntemi

Üç tür renk vardır. Çerçeve rengi, orta alanın çevresindeki renktir. Mürekkep rengi, karakterin (harflar, rakamlar, işaretler ve grafik şekillerin) ve çizgilerin noktalarının rengini belirler. Kağıt rengi, ya bütün alanın ya da o harfin etrafındaki küçük karenin tonunun rengidir.

Spectrum'u ilk çalıştırdığınızda, önceden belirlenmiş renkleri kullanır. Mürekkep rengi siyah, çerçeve ve kağıt rengi beyazdır. Klavyedeki tuşlarla vereceğiniz komutla bu renkleri derhal değiştirebilirsiniz. Bunu zaten daha önce 6 ve 7. sayfalarda görmüştünüz. BORDER komutunu kullanarak televizyonunuzun ve Spectrum'unuzun renkleri için doğru bağlanıp bağlanmadığını denetlemiştiniz. Şimdi RESET'e basıp,

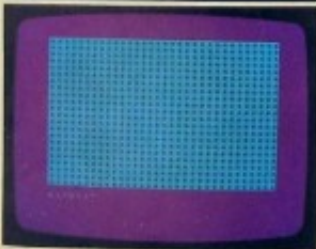
aşağıdaki basit programı verip, işleme koyun.

### RENK TESTİ

```
10 PRINT "RENK TESTİ"
```



Siyah ve beyaz yıldızlardan oluşan bir desen belirecektir. BREAK'e basıp, programı yeniden işleme koyun. Aşağıda iki ekrandan üstte BORDER 4, PAPER 2 ve INK 7, altta ise BORDER 3, PAPER 5 ve INK 1 komutu verilmiştir.



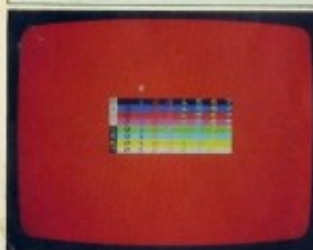
### Renk kullanarak program yazma

Bir programda BORDER, PAPER ve INK komutlarını kullanarak metinler, tablolar ve desenler yapabilirsiniz ve ekranda rengarenk resimler belirebilir. Bir program satırında BORDER kullandıysanız, Spectrum bu satıra gelince çerçevenin rengi değişir. Bir satırdaki INK komutu ardından gelen harf ya da satırlar o rengi alır. PAPER komutu kendi başına verilirse, yalnızca karakterlerin (buna nokta ve çizgiler de dahildir) çevresindeki rengi değiştirir. Bütün fonu belli bir renk yapmak istiyorsanız, PAPER'den sonra CLS'ye basmak gerekir.

PRINT'ten sonra da INK ve PAPER kullanabilirsiniz. Bu durumda, sadece PRINT'ten sonra kullanılmış karakterler INK ve PAPER'in renklerini alır. Şimdi vereceğimiz program, çerçeve, mürekkep ve kağıt renklerinin tüm kullanımlarını, ayrıca INK ve PAPER'in PRINT'ten sonra nasıl kullanılacağını gösteriyor.

### RENK BİLEŞİMLERİ

```
10 FOR I=0 TO 7
20 BORDER 3: PAPER 5: CLS
30 PRINT HT 6,10: INK 0
40 FOR J=0 TO 7
50 PRINT HT 6+J, INK J: DDE
60 BEEP 0.5:KAD-20HP
70 FOR L=0 TO 7
80 PRINT INK L: DDE
90 BEEP 0.5:KAD-20HP
100 NEXT L
110 NEXT J
120 NEXT I
```



Bu programı işleme koyduğunuzda, çerçeve, mürekkep ve kağıt renkleriyle yapılabilecek bütün bileşimleri görebilirsiniz. Programda üç değişken var: b çerçeve, i mürekkep, p kağıt rakamının değeri. BEEP sesi sağlıyor. FOR ve NEXT ile başlayan satırlar, sırasıyla 0 ile 7 arasındaki bütün renk rakamlarının kullanılacağı döngünün başlangıç ve bitişini belirliyor. Döngüde FOR ve NEXT'in kullanımı hakkında 27. sayfada daha ayrıntılı bilgi bulabilirsiniz. Gördüğünüz gibi, INK ve PAPER'in değeri 9 olabilir. Bu, mürekkep ya

da kağıt renginin, herhangi bir fon ya da karakterin üzerinde görülebilmesi için, siyah ya da beyaz olmasını sağlar.

### Renkli çubuklu grafiklerin programlanması

Bu program, Spectrum'un renklerinin kullanılmasıyla bir renkli çubuklu grafik yapımını sağlar. 12 günlük ısıyı, üzerinde rakam yazılı çubuklar olarak gösterir. 60. satırda tırmak işaretlerinin arasında iki boşluk bırakın.

### ÇUBUKLU ÇİZELGE

```
10 BORDER 0: PAPER 1: CLS
20 FOR I=1 TO 12
30 FOR J=1 TO 12
40 READ I
50 FOR L=1 TO 12 STEP -1
60 PRINT BAPER 0: HT L, I: CLS
70 NEXT L
80 PRINT INK 0: HT 20-I, I: CLS
90 NEXT I
100 NEXT J
110 DATA 20, 15, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10
```



Şimdi aşağıdaki satırları ekleyin ve 110. satırı gösterdiği gibi yeniden yazın. Artık çizelge iki renkli olmuştur. READ ve DATA hakkında daha fazla bilgi için 33. sayfaya bakın.

### ÇİFT ÇUBUKLU ÇİZELGE

```
20 READ I
30 FOR L=1 TO 12 STEP -1
40 PRINT PAPER 0: HT L, I: CLS
50 PRINT INK 1: PAPER 0: HT 20-I, I: CLS
60 DATA 20, 15, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10
```





## BASİT BAZI GRAFİKLER

ZX Spectrum+'ınuzunla alçak ya da yüksek tanımlı bazı çizimler yapabilirsiniz. Her ikisi de aynı anda ekrana yansıtılabilir. Alçak tanımlı çizimler renkli bazı karelerden oluşur. Bu iki sayfada bu karelerin klavyeden nasıl elde edilebileceğini ve bunların ekrana nasıl yansıtılabileceğini göreceksiniz.

### Alçak tanımlı çizimlerde ekran

Alçak tanımlı çizimlerde ekrana enlemesine 32 karakter yerleştirilebilir. Boylamasına ise 22 karakter alır. Ekrandaki her pozisyonu iki ayrı rakam belirler. Önce dikey satır rakamı verilir. Bu, aşağıda kaç satır kaldığını gösterir. En üstteki satırın rakamı 0, en alttakini ise 21'dir. Bunu yatay sütun rakamı izler. Sol baştaki sütunun rakamı 0, en sağdakini ise 31'dir. (80. sayfada alçak tanımlı çizim çizilmesinin tümünü görebilirsiniz). Şimdi göreceğiniz program, bu pozisyonların renklerle nasıl doldurulacağını gösteriyor. R tuşundaki RND komutu rastsal bir mürekkep rengi seçer.

### RASTSAI KARFLER

```
10 BORDER 1: INK RND 47
20 DO UNTIL
30 DO TO 30
```



Bütün ekrani kareler kaplar. Herhangi bir karakterin belli bir pozisyonda belirmesi için, PRINT ile birlikte AT komutunu kullanmanız gerekir. PRINT'in ardından AT, bundan sonra da önce satır rakamı, virgül, sütun rakamı ve noktalı virgül tuşlarına basın. Örneğin, şu komutta

```
PRINT AT 11,16:"*"
```

11. satır, 16. sütunda bir yıldız çıkacaktır.

### Gökkuşağı deseni nasıl çizilir?

Renkli desenler elde etmenin iyi bir yolu, çizim programınızda FOR NEXT döngüsünü kullanmaktır. FOR NEXT döngüsü, kendisini belirli kez tekrarlayan bir program parçasıdır. Bunu yaparken, ekrana belli karakterleri yerleştirmesi de istenebilir.

### Grafik karakterlerin seçimi

ZX Spectrum+'ınuzun klavyesinde bazı grafiklerin programlanmasını kolaylaştıran bir dizi grafik karakter vardır. Bunları 1-8 arasındaki tuşlarda görebilirsiniz.

Grafik karakterleri ekrana aktarmak için, GRAPH tuşu ardından 1'den 8'e kadar olan tuşlara, aralarında ara tuşuna da dokunarak, basın. Grafik karakterler ekranın altında belirecektir. Bu karakterlerin tuşlarını üzerinde beyaz olan bölümü ekrana mürekkep renginde, siyah bölümü de kâğıt renginde yansıtır. Bu defa CAPS SHIFT'e basarak, bu tuşlara bir kez daha dokunur. Mürekkep ve kâğıt renkleri yer değiştirir.

Program satırlarına grafik karakterleri aynen böyle yerleştirirsiniz. Grafik karakterlerden rakamlara dönmek isterseniz, GRAPH'a bir kez daha basın.

8 tuşu  
Bu tuş, genellikle GRAPH ve CAPS SHIFT ile birlikte tam bir renkli kare elde etmede kullanılır.

GRAPH  
Bu tuş Spectrum'u grafik moduna geçirmede kullanılır.



Her defasında sadece bir döngü kullanmakla sınırlı değilsiniz. Bir döngüyü, bir diğerinin içine yerleştirebilirsiniz ve bu çoğu kez yararlı sonuçlar verir. Aşağıda (biri diğerinin içine "yuvalanmış") iki FOR NEXT döngüsünde INK ve AT'in sağlayacağı pozisyonların ve renklerin değişebileceğini göreceksiniz. Bu döngülerin nasıl hazırlanacağını bu sayfanın sonundaki karede bulabilirsiniz.

### GÖKKUŞAĞI

```
10 BORDER 0: PAPER 5: CLS
20 LET X=1
30 FOR I=0 TO 21
40 FOR J=1 TO 8
50 PRINT INK C: AT I, J:
60 NEXT J
70 NEXT I
```



### Resimlerin programlanması

Alçak tanımlı çizimlerde, grafik karakterlerin pozisyonları ve renkleri ile oynayarak resimleri "boyayabilirsiniz". 80. sayfadaki basit çizim çizilmesinden yararlanarak resimlerinizi programlayabilirsiniz. Bundan sonra, karşı sayfada gösterildiği gibi grafik karakterleri seçerek, resmi oluşturacak program satırlarını birer birer verebilirsiniz.

Yandaki program nasıl bir sonuç elde edeceğinizi gösterecektir. Buradaki bütün şekiller rakam tuşlarındadır. Programı işleme koymak için, tümünü verene kadar bekleyebileceğiniz gibi, her satırı verdikçe işleme koyacak olursanız, robotun çeşitli bölümlerinin nasıl oluştuğunu görebilirsiniz. (Unutmayın eğer yanlış grafik karakter verecek olursanız, bunları da tipki yanlış rakam veya harfleri düzelttiğiniz gibi düzeltebilirsiniz.)

### ZX ROBOT

```
10 BORDER 2: PAPER 1: CLS
20 LET X=1
30 FOR I=0 TO 21
40 FOR J=1 TO 8
50 PRINT INK C: AT I, J:
60 NEXT J
70 NEXT I
```



70. satırda PRINT'ten sonra görülen TAB komutu, bir karakteri bilgisayarı o anda üzerinde çalıştığı satırda belli bir yere yerleştirmek için kullanılır. TAB'ı izleyen 0 ile 31 arasındaki rakam, sütunu gösterir.

### FOR NEXT döngüsünün kullanımı

FOR NEXT döngüsü daima FOR ve TO komutlarıyla, bir değişken ve bunun başlangıç ve bitiş değerlerini içeren bir satırla başlar. Örnek:

```
30 FOR C=1 TO 6
```

Burada değişken c. Bununla başlayacak döngü, bilgisayarı bu programı tekrarlamasını sağlayacak satırları içerecektir. Bunlarda da c değişkeni kullanılabilir. FOR NEXT döngüsü, daima NEXT komutu ve değişkenle sona erer. Örnek

```
50 NEXT c
```

Program işleme konunca, FOR'dan NEXT'e kadar ki bölüm kendisi defalarca tekrarlar. Değişken TO'dan önceki değeriyle başlar ve TO'dan sonraki limite kadar her seferinde 1 artar. Bu durumda kendini altı kez tekrarlayacak, c başlangıçta 1, sonra 2,3,4,5 ve son olarak da 6 olacaktır.

25. sayfadaki ilk programda, bir "yuva" da üç döngü kullanılmıştır. Bu demektir ki, "dış" döngünün her turunda, "orta"daki bir tur yaparken de "iç" döngü bütün devrelerini yapmaktadır.

## RESİM DEFTERİ OLAN EKRA

ZX Spectrum + ile yapılabilecek çizimler, iri parçalardan oluşan alçak tanımlı çizimlerle sınırlı değildir. Karmaşık yetenekleri nedeniyle Spectrum'u düz veya eğri hatlı bazı şekillerin ayrıntılı görüntülerini çizmek için de kullanabilirsiniz.

Yüksek tanımlı çizimler, bir çizgi oluşturmak ya da düz renkli bir şekli doldurmak için yanyana dizilmiş noktalarla oluşur. Her nokta, alçak tanımlı çizimlerde kullandığımız karelerin 64'te biri büyüklüğündedir. Şu komutu verirsiniz

**PLOT 128,87**

bunlardan biri ekranın tam ortasında görülebilir.

Yüksek tanımlı çizimlerde kullanılan noktalara pixel denir. Bu İngilizce picture cell'in, resim hücrelerinin kısaltılmışıdır. Tipik alçak tanımlı çizimde olduğu gibi pixel'in yerini de iki rakam belirler. Ancak bunlar alçak tanımlı çizimlerdeki rakamlar değildir.

### Yüksek tanımlı çizim çizelgesi

Yüksek tanımlı çizim çizelgesinde yatay 256, dikey 176 pixel vardır. Ancak basit çizimin tersine, ilk rakam yatay koordinatı, yani ekranda enlemesine yerini gösterir. Bu rakamlar sol köşede 0, sağ köşede 255'tir. İkinci rakam olan dikey koordinatta ise rakamlar aşağıdan yukarıya doğru büyür. Aşağıda 0, yukarıda 175'tir. 0,0 pozisyonu alçak tanımlı çizimde olduğu gibi sol üst değil, sol alt köşedir. Yüksek tanımlı çizim çizelgesi için sayfa 80'e bakın.

### Yer belirleme ve çizme

Yüksek tanımlı çizimler için üç komut yeterlidir: PLOT, DRAW ve CIRCLE. PLOT'u virgülle ayrılmış yatay ve dikey koordinatlar izler ve komut buraya bir pixel kondurur. DRAW'u da virgülle ayrılmış iki rakam izler ama bunlar koordinat değildir. Bunlar pixel olarak, iki noktanın arasındaki yatay, dikey ya da çarpazlama mesafesidir. DRAW komutu iki nokta arasında bir çizgi çeker.

Programda eğer daha PLOT ve DRAW kullanılmamışsa ilk pozisyon 0,0'dir.

Kullanılmışsa, hangisi en son kullanılmışsa, PLOT'daki son nokta ya da DRAW ile ulaşılan son noktadır. Bundan sonraki DRAW komutu yeni noktaya bir çizgi çeker. Eğer çizgi sola ya da aşağı gidecekse, yatay veya dikey mesafeler eksi (-) değerle verilir. Şu programı deneyin:

### YILDIZ

```
10 INK 0
20 PLOT 0,0
30 DRAW 70,-140
40 DRAW 180,0
50 DRAW 140,140
60 DRAW -180,0
70 DRAW -70,-140
```



PLOT başlangıç noktasını ekranın tepesine götürür, ardından DRAW komutları beş kırmızı çizgi çizer.

Şimdi programa şu satırları ekleyin:

```
4 BORDER 1: PAPER 6: INK 1: CLS
5 CIRCLE 128,87
```

Programı yeniden işleme koyun, renkli zeminli bir daire içinde kırmızı çizilmiş bir yıldız görürsünüz.



CIRCLE için üç değer gerekir. İlk ikisi dairenin merkezinin konumunu, üçüncüsü ise yarıçapı belirler. DRAW komutlarına da bir üçüncü değer ekleyebilirsiniz. Programda 2

ile -2 arasındaki değerleri deneyin ve bakın ne oluyor.

### Şekiller nasıl doldurulur?

Alçak tanımlı çizimde, şekillerin içini, birçok çizgiyi yanyana çizerek kolayca doldurabilirsiniz. Bunu DRAW pozisyonlarını her defasında 1 arttıran FOR NEXT döngüsü ile elde edebilirsiniz.

### DOLU ÜÇGEN

```
10 BORDER 1: PAPER 6: INK 2: CLS
20 FOR X=100 TO 100
30 PLOT 100,100
40 DRAW X,-100
50 NEXT X
```



Eğer çizgileri biraz aralıklı çizerseniz ilginç bir sonuç elde edersiniz. Bunu FOR satırına STEP komutu ve bir rakam da ekleyerek yaparsınız. Bu teknik 11. sayfadaki doğan güneş programında kullanılmaktadır. Üçgen programında da aynı işi yapacaktır. Satır 20'yi değiştirip yeniden işleme koyun.

```
20 FOR X=100 TO 100 STEP 4
```

Bu defa aşağıda gördüğünüz yelpaze benzeri şekil oluşur. Bunun nedeni STEP'in her çizgi çizildikçe x'i 1 ilerletecek yerde 4 sıçratmasıdır.



### Ekrandaki resim defteriniz

Spectrum'unuz çok yeteneklidir. Bir resim ya da şekil çizdiğiniz zaman her defasında bir program yazmanız gerekmez. Bunun yerine ekrana doğrudan resim çizmenizi mümkün kılacak bir program yazabilirsiniz. İşte buna olanak sağlayacak basit bir program.

Program size mürekkep rengi soracak olan INPUT komutu ile başlıyor. Ardından (bu defa bir diziyi belirlemek üzere \$ işaretini kullanarak) INPUT komutu ile dört tuşa - u, d, l ve r - basılığa bilgisayarın kısa bir çizgi çizmesini sağlıyor.

### RESİM DEFTERİ VE ÖRNEK

```
10 INPUT "INK: "; I
20 BORDER 1: PAPER 6: INK I: CLS
30 PLOT 80,20
40 LET X=0
50 INPUT "X: "; X
60 IF X=0 THEN DRAW 0,X
70 IF X=1 THEN DRAW 0,-X
80 IF X=2 THEN DRAW X,0
90 IF X=3 THEN DRAW -X,0
100 GOTO 50
```



### IF ve THEN ile karar verme

Resim defteri programının 60-90 satırları IF THEN cümlesi içeriyor. Bu Spectrum'unuzun karar almasını sağlar. Bu durumda bilgisayar u, d, l ve r tuşlarına mı basıldığını bakar. IF (eğer) bu tuşlara basılmışsa, THEN (o zaman) bilgisayara bir çizgi çizmesi söylenmiş demektir. Başlığınız büyük harfle çizgi çizmez.

IF her zaman, Spectrum'un yapıp yapmadığını sınavacağı bir olgu ider - belli bir tuşa basılıp basılmadığı gibi. Gerçekse ya da oluyorsa, bu durumda THEN'de belirtilen iş yapılır. Değilse, program bir sonraki satıra geçer.

THEN'i izleyen satırdaki her şey bir karara bağlıdır. Şu satırda

```
110 IF B=5 THEN PRINT "+"; GOTO 200
```

b ancak 5 ise bilgisayar 200. satıra gidecektir.



## DESEN VE RESİM ÇİZİMLERİ

Alçak veya yüksek tanımlı çizimleri ya da her ikisini de kullanarak ZX Spectrum+ ile her tür desen ya da resim çizilebilir. Bu sorunun üstesinden gelmenin en iyi yolu, deseni 80. sayfadaki çizilgenin bir kopyası üzerine çizmektir. Bundan sonra istenen yerde çizgi ve biçimler oluşturacak programı çıkarmak gerekir.

Desen ya da resimleri çizerken, programın belli bir bölümünü birkaç kez tekrarlayacak FOR NEXT döngülerini kullanabilirsiniz. Her defasında pozisyonları, karakterlerin ya da çizgilerin renklerini, genellikle düzenli bir biçimde değiştirebilirsiniz. İşte, bu tekniği kullanan bir program:

### KARELER

```
10 BORDER 0: PAPER 0: CLS
20 FOR x=7 TO 0 STEP -1
30 INK x
40 FOR y=11-x TO 11+x
50 FOR c=10-x TO 10+x
60 PRINT AT 1,1: " "
70 NEXT c
80 NEXT y
90 NEXT x
```



Bu programda üç FOR NEXT döngüsü içeriyor. x döngüsü, yapılan büyük karelerin rengini ve boyutunu değiştirirken, l döngüsü ile c döngüsü her defasında küçük karenin sabit ve sütun olarak pozisyonunu değiştirir. 60 satırdaki kareyi bir yıldıza çevirin ya da klavyedeki başka herhangi bir karakterle değiştirin.

### Rastsal etkiler ve alt programlar

Döngülerin kullanılması her defasında benzer şekiller elde edilmesi anlamına gelmiyor. Döngülerde RND (İngilizce RANDOM, rastsalın kısaltılmışı) komutunu kullanarak, her defasında renkten,

pozisyonları ve diğer görüntüleri değiştirebilirsiniz. 10. sayfadaki mozaığa bakın. Bu, mürekkep rengi RND 7 olduğu için bu sonucu veriyor. Çünkü bu 0 ile 7 arasında herhangi bir ondalık sayı demek. İNK bunu en yakın tam sayıya çeviriyor. Bu nedenle her defasında bir kare görününce, rengi İNK 0 ile İNK 7 arasında bir renk oluyor.

Bir sonraki program, ekranda grafik karakterlerle simetrik desenler çiziyor. Bu karakterleri ve renklerini değiştirmek için RND kullanıyor. i ve p değişimleri mürekkep ve kağıt renklerini veriyor, a ise kaç desen çizileceğini (bu örnekte 4 kez) gösteriyor. n değişimini her desende kaç karakter olduğunu belirliyor. x ise 129 ile 142 arasında rastsal bir rakam.

50. satırdaki GOSUB 1000 bilgisayar alt programı gönderir.

### SİMETRİK DESENLER

```
10 BORDER 0: PAPER 0: CLS
20 LET i=4: LET p=0
30 FOR a=1 TO 4
40 LET n=RND*13+129
50 FOR x=1 TO 40: GOSUB 1000
60 NEXT x
70 LET i=i+1: LET p=p+1
80 IF i=5 THEN i=1: IF p=5 THEN p=1
90 NEXT a
100 STOP
1000 LET i=INT (RND*13)
1010 LET c=INT (RND*16)
1020 INK i: PAPER c
1030 PRINT AT 1,1: CHR$ x
1040 PRINT AT 1,1: CHR$ x
1050 PRINT AT 1,1: CHR$ x
1060 PRINT AT 1,1: CHR$ x
1070 PRINT AT 1,1: CHR$ x
1080 RETURN
```



Alt program, program içinde program gibi hareket eden bir grup satırdır. Bu programda, alt program 1000. satırdan başlıyor. Ekranın dört çeyreğine de, hepsi merkezden (11,16 pozisyonu) aynı uzaklıkta bir dizi grafik karakter yansıtıyor. 1000. ve 1010. satırlarda verilen mesafelerde, i satır, c de sütun olarak mesafeyi belirliyor. İNT rastsal sayısı, i'nin 0 ile 10, c'nin de 0 ile 15 arasında olabilmesi için bir tam sayıya çeviriyor. 1030'dan 1060. satıra kadarkiler kodu x olan grafik karakterleri belirliyor. (51. sayfadaki karakter listesine bakın).

BEEP tonu bulunduğu noktaya göre değişen bir ses çıkarıyor, ardından 1080. satırdaki RETURN, programı 50. satırdaki GOSUB'dan sonra gelen son cümleye gönderiyor.

60. satır mürekkep ve kağıt rengini değiştiriyor, 70. satırdaki PAUSE 100, programı döngü yeniden başlamadan 2 saniye durduruyor. 90. satırdaki STOP ise dördüncü döngüden sonra programın derhal alt programa geçmesini önlemek için gerekli.

30. satırdaki 4 ve 50. satırdaki 40'ı başka rakamlara çevirerek, bu programı değiştirebilirsiniz. 40. satırdaki x'in alanını daha genişletirsiniz, başka karakterlerin ekrana yansımaları önlersiniz. i ve p'nin de 7'den büyük olmaması gerekir.

### ÇİZİMLERDE FOR NEXT DÖNGÜSÜNÜN KULLANIMI

FOR NEXT döngüleri, yüksek tanımlı çizimlerde düzgün şekil ve çizgilerden oluşan resimlerin yapımında çok iyi kullanılabilir. Aşağıdaki programı yazıp, işleme koyun. Sadece PLOT ve DRAW komutlarıyla, iki FOR NEXT döngüsü, önce yerdeki çizgileri, sonra da içi dolu beş üçgen ya da piramidi çizer.

### PIRAMİTLER

```
10 BORDER 0: PAPER 1: INK 0
20 CLS
30 FOR y=0 TO 20 STEP 2
40 PLOT 0
50 DRAW 255,0
60 NEXT y
70 FOR x=100 TO 220 STEP 30
80 FOR n=10 TO 10 TO 10+n/10
90 PLOT 0: DRAW 10
100 DRAW x: n+2
110 NEXT n
120 NEXT x
```



Şimdi de yan sütundaki satırları ekleyin. İşleme koyduğunuzda, gece karanlığında yerden yansıyan bir lazer ışınının göğün yıldızlarla kaplanmasına yol açtığını göreceksiniz. Çizgi, ekranın köşesinden x,y pozisyonuna çiziliyor. x,y değişimlen rastsal sayılır. Bunlar daha sonra yıldızlar için alçak tanımlı çizimdeki pozisyon rakamlarına dönüştürülür.

```
120 LET x=RND*255
130 LET y=RND*100+73
140 LET z=INT (12/0.1)
150 PLOT 0: DRAW OVER 1,x,y
160 DEEP 0:0:1,x,y
170 PLOT 0:0:1,x,y
180 PLOT 0:0:1,x,y
190 PRINT AT 1,1: " "
200 GO TO 120
```



160 ve 180. satırlardaki OVER 1, ilk satırın lazer ışını çizdikten sonra, ikinci satırın onu silmesini, ancak resmin geri kalanının bozulmamasını sağlıyor. İleride gerekeceği için bu programı saklayın.

### FLASH, BRIGHT ve INVERSE

Bu üç komut Spectrum'un renklerini gerçekten harekete getirebilir. Her komutu 0 veya 1 izler. 0 veya 1'den sonra noktalı virgül kullanmak koşuluyla, bunları PRINT cümleleri olarak verebilirsiniz. FLASH 1 karakterlerin bulunduğu yerlerin, mürekkep ve kağıt rengi arasında yapıp sönmelerini, BRIGHT 1 renklerin parlaklaşmasını sağlar. INVERSE 1 mürekkebi kağıt rengi, kağıdın mürekkep rengi yapar. Bu komutlardan sonra 0 kullanımı ekranı normal hale getirir.

Bu komutların ne sonuç verdiğini görmek için, bu iki sayfadaki programlarda şu değişiklikler yapın. Kareler programında 60. satırdaki kareyi yıldıza çevirip, şunu ekleyin:

### 15 INVERSE 1

Değişen mürekkep renkli bantlar üzerinde siyah (kağıt rengi) yıldızlar belirir. Devam etmeden INVERSE 0 ekleyin.

Simetrik Desenler programında BRIGHT ve FLASH'ın çalışmasını görmek için şu satırları ekleyin:

### 15 BRIGHT 1

### 16 FLASH 1

FLASH deseni sanki ileri geri gidiyormuş gibi gösterir. Ekrandaki yapıp sönmeyi durdurmak için FLASH 0: CLS komutu verin.

Bu değişiklikler her programdaki bütün görüntüyü etkiler. FLASH, BRIGHT ve INVERSE'i bir PRINT cümlesinde kullanırsanız, bu üç komutun etkisi o satırda yazılanlarla sınırlı kalır.



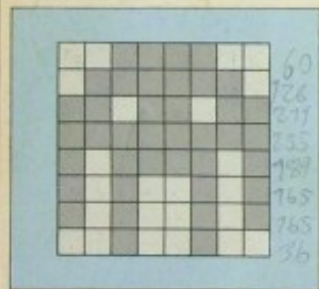
## BİLGİSAYAR KARAKTERLERİNİN YARATILIŞI

ZX Spectrum +1'unuz, sadece klavyedeki tuşlarla yapılan grafik karakterlerle sınırlı değildir. Belleğinin özel bir bölümünde sizin oluşturacağınız karakterleri de saklayabilir. Bunlara kullanarak belirleyeceğiniz grafik karakterler denir ve her programda bunlardan en çok 21 tane kullanılabilir.

Her karakteri mürekkep rengindeki 64 küçük nokta ya da pixel'den oluşur. Bunlar her bir sırada 8 tane olmak üzere sekiz sıra halinde yer ve her karakteri alçak tanımlı çizim çizelgesindeki bir karakterin yerini işgal eder - tıpkı klavyedeki standart grafik karakterler gibi.

### Bir grafik karakterin hazırlanışı

Önce aşağıda gördüğünüz gibi 8x8'lik bir çizelge hazırlayın. Sonra karakterleri bazı kareleri doldurarak biçimlendirin. Bu kareler, mürekkep rengindeki pixel'lerdir. Sonra ya yazarak ya da kafanızdan, dolu kareleri 1, boşları da 0 olarak alın. Şu bir örümcek çizimi:



Kullanının belirlediği grafik karakterlerden her biri a ile u (ya da A ile U - hiç fark etmez) arasında bir harfle tanımlanır. Karakter programlamak için sekiz POKE USR cümlesi vermeniz gerekir. Bunlar BIN ile başlar, ardından da çizilgenin her satırına karşılık 8 tane 1 veya 0'dan oluşan bir ikil rakam ile sona erer. Örnecek karakteri s olsun,

[illegible]

Şimdi programı işleme koyun, ardından GRAPH ve s'ye basın. Ekranda s yerine bir örümcek belirir. Bir de şu satırları ekleyin. 30. satırdaki örümceği de anlattığımız yoldan elde edin. Programı işleme koyun. Bütün ekranı örümcekler kaplar.

## ÖRÜMCEKLER

20 BORDER 1: PAPER 0, CL5  
30 PRINT 148: RND+7: 0-1  
40 GO TO 30



Kendi karakterlerinizi oluştururken, unutmayın, bunları tanımlayan programı işleme koymadıkça bunları ekranda göremeyeceksiniz. O zamana kadar bunlar harflerden oluşan sıralamalar halinde kalacaktır.

Renklerin benekli karelerle "karıştırılması"

Spectrum'unuzda renkleri kolayca karşılaştırabilirsiniz. Bunu yapmak için bir kareyi yüzde 50'si mürekkep, yüzde 50'si kağıt renginde dolduracak bir karakter oluşturmak gerekir.

```

10 FOR x=0 TO 6 STEP 2
20 POKE USH - 4 + x, BIN 10101010
30 POKE USH - 3 + x + 1, BIN 010101
40 NEXT x

```

Bütün yapmanız gereken, iki pixel satın  
belirleyip bilgisayara, bunları karakterde  
dönüştürme kullanma talimatı vermektir. Programı  
işleme koyunca benekli bir kare göreceksiniz.  
Eğer aynı teknoloji renk komutları içeren bir  
programda kullanırsanız, benekli kare,  
programın mürekkep ve kağıt renklerinin  
kaybını yansıtabilir.

### READ ve DATA ile çizimleri basitleştirme

Bilgisayar grafiklerinizi oluşturmanın daha kolay bir yolu vardır. O da READ ve DATA ile ondaki sayılar kullanılmaktadır. Önce 0 ile 1'lerden oluşan sekiz ikil sayıyı ondaki sayıya çevirin. Bunu PRINT BIN ardından sayıyı vererek yapın. Örneğin

## PRINT BIN 00111100

Spectrum'da 60 görünür, bu 00111100'in ondalık halidir. Örümceğin sabırlarının ondalık sayıları 60, 126, 219, 255, 189, 165, 165 ve 36'dır.

Artık READ ve DATA'yı kullanabilirsiniz. Bu iki komut, bir programa değışen olarak rakam gibi birçok değeri kolayca verebilmeyi sağlar. READ'yı bir değışen izler. Bu, eğer rakamsa, bir veya birkaç harf, bir dizi ise \$'i izleyen tek bir harftir. Burada öndaki sayıları okumak (READ) istediğimize göre, rakamsal bir değışen seçeriz. Buna v diyoruz.

Spectrum'unuz READ ile karşılaştıra, programdaki ilk DATA (veri) cümlesine bakar. Bu cümle, virgüllerle ayrılmış bir dizi değerden oluşmaktadır. Bilgisayar buradaki ilk değeri alır, READ'e geldiğinde ise ikinci değer verir ve bu sırayla böyle gider.

İşte size drumcek yapacak yeni bir program.

```

10 FOR x=0 TO 7
20 READ Y
30 DUKE CSR "S" + x, Y
40 NEXT x
50 DATA 00,120,219,288,169,165
,188,36

```

Aslında bu program bir karakter oluşturmak üzere beşleçe herhangi sekiz ondalık sayıyı yazdırabilir. 30. satırdaki `5` yi istediğiniz harfe çevin. 50. satırdaki `DATA` dan sonra herbin virgül ile ayrılmış sekiz sayı yazın. Programı işleme koyduktan sonra `GRAPH` ile seçtiğiniz harfe basın.

## Renk kontrol kodlarının kullanımı

INK (mürekkep) ve PAPER (kağıt) gibi komutları kullanacağınıza, PRINT cümlelerini ilk tırnaktan sonra kontrol kıldanını koyabilirsiniz.

Ekrana yansıtacak karakterler, bu durumda, sıralamada ekranda alacakları renkte görünürler.  
Kodları elde etmek için önce

EXTEND MODE'a sonra da CAP SHIFT'le birlikte veya tek başına bir rakam tuşuna basın. Bu çizelge herhangi bir rengi nasıl seçeceğinizi gösteriyor:

Satranç tahtası çizimi

Bu, ekrana satranç tahtası çizip, taşların oynanmaya hazır yerleştirilen bir programdır. Aşağıdaki çerçevede gibi, renk kontrol kodlarını kullanarak, renklerin bir sıralamada yer almasını sağlayarak bunu yapabilirsiniz.

## SATRANC

```

10 FOR X=1 TO 8
20 READ A$
30 GO SUB 500
40 NEXT X
50 BORDER 4, PAPER 1, CLS
60 FOR I=7 TO 14 STEP 2: FOR
  J=1 TO 19 STEP 2
70 PRINT AT I,C:PRINT J+I,

```

```

90 NEXT C NEXT I
90 PRINT AT 7:12; " ";
100 PRINT AT 8:12; " ";
110 PRINT AT 9:12; " ";
120 PRINT AT 13:12; " ";
130 GO TO 130
900 FOR n=8 TO 7
910 READ Y
920 POKE USR aa+n,Y
930 NEXT n
940 RETURN

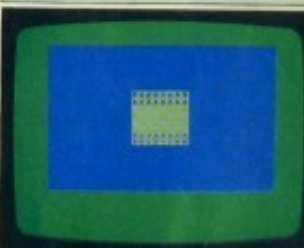
```

500. satırda başlayan alt program oyun taşlarını belirler.

```

550 DATA F,0.0,16,50,50,16,12
560 DATA F,0.04,124,50,50,124
570 DATA A,0.16,50,124,124,50
580 DATA B,0.16,40,60,120,50
590 DATA K,0.16,50,16,50,50.5
600 DATA L,0.04,40,16,100,124
610

```





## HAREKETLİ GÖRÜNTÜLER

Bilgisayar çizimleri, eğer karakterler ve çizimler ekranda hareket edebilirse çok daha hoş görünür. Spectrum'unuzda bu hareketliliği yaratmak çok zor değildir. Bütün yapmanız gereken, bir karakterin ya da çizimin bulunduğu yeri sürekli değiştirmektir. Bunu yapmanın en iyi yolu da bir ya da daha fazla FOR NEXT döngüsü kullanmaktır.

### Dikey ve yatay hareket

Bu programı yazıp, işleme koyun. Eğer 32. sayfadaki örümcek gibi grafik karakteri yaptıktan sonra, Spectrum'unuzu kapatmadınızsa ya da resef'e getirmediğinizse 10 ile 50 arasındaki sabırları yazmanız gerekmez. Bu grafik karakterin hâlâ bellekte "c" harfinden bulunması gerekir.

## İNEN ÖRÜMCEK

```

8 BORDER 3, PAPER 5: CLS
10 FOR I=0 TO 70
20 READ V
30 PRINT USING "5 * * * V"
40 NEXT I
50 DATA 60,126,219,255,169,165
100 FOR X=0 TO 7
70 READ D
80 PRINT USING "5 * * * V"
90 NEXT X
100 DATA 16,16,16,16,16,16,16,16
#
110 FOR I=0 TO 50
120 PRINT AT I*3, I: INK 0;"| "
130 NEXT I
140 NEXT I

```

Bu programı her işleme koyuşunuzda  
örümceğin iplikten indiğini göreceksiniz.

Programda 60'tan 100. satıra kadar olanlar bir başka grafik karakter ("I") olan işpli yapıyor. Hareketi, satır sayısı 1'i 0 ile 20 arasında değtiren FOR NEXT döngüsünü oluşturan 110 ile 140 arasındaki satırlar sağlıyor. Döngünün kendini her tekrarlayışında, belli bir pozisyonda bir miktar işpli oluşuyor ve örmek bir sonraki pozisyonda beliyor. Bir sonraki turda bir miktarda işpli daha oluşuyor ve örmek yeniden aşağıda beliyor. Böylece örmek 21. satırda yarı görüntünün en altına kadar hızla inlinin ucunda iniyor.

Spectrum'unuz yeni pozisyonları hızla hesaplayabilir ve bu nedenle örümcek hızla iner. Bunu yavaşlatmak için şu sabrı verin:

135 PAUSE 10

Bu, bilgisayarın, örümceği bir sonraki pozisyonunda yansıtmadan önce saniyenin beşte biri kadar beklemesine neden olur.  $10^8$  u başka bir rakama çevirin ve etkisini izleyin.

Şimdi de programa şu satırları ekleyin ve yeniden işleme koyun, bu defa hareketi farklı bir yönde göreceksiniz.

## KACAN ÖRÜMCEK

```
150 FOR C=3 TO 30
160 PRINT AT 21,C:INK 21;"0"
170 PRINT AT 21,C+1:INK 21;"0"
180 NEXT C
```



Bu defa örümcek yere değer değmez bir tarafa kaçıyor. Ek satırız süzünü pozisyonu 'c'yi denetleyin bir başka FOR NEXT döngüsü oluşturuyor. Bakon önce bir boşluk belirliyor, ardından örümcek bir sonraki sütun pozisyonunda ortaya çıkıyor. Böylece örümcek sağa giderken, bir pozisyonda kaybolup, bir diğerinde belirliyor. Bir karakterin eski bulunduğu pozisyonda yok etmeden, yeni bir yere yansıtılmaktansa, eskisini yok ettikten sonra yeniden yansıtılmak saha iyidir. Bu, hareketli çizimlerde titremeyi yok eder ya da azaltır.

### Hedef uygulaması

Birçok bilgisayar oyunlarında, olay, ya hareket halindeki iki şekil çarpınca ya da bir şekil ışınla isabet alırsa meydana gelir. Bilgisayar çarpışma ya patlamanın olacağını nasıl biliyor?

Çarpışma olacağını kestirmek güç değil. Eğer iki karakterden biri (sabit), c(sütun) pozisyonunda, diğeri de v(dikey), h(yatay) pozisyonundaysa ve  $l=v$ ,  $c=h$  ise, ikisi de aynı pozisyonlarda demektir. Bunu şöyle bir cümle olarak yazabilirsiniz:

```
160 IF l=v AND c=h THEN PRINT
    "CRASH!"
```

Çarpışmaları kontrol etmenin bir başka yolu da renkleri kullanmaktır. NEW komutuyla örümcek programının tümünü verin, ya da bunu teypte sakladınızsa bilgisayara yükleyin. Şimdi bunu bir üst düzeye çıkarabilir ve (eğer bu arada kapatmadıysanız) ya da reset'e getirmediyse bilginin belleğindeki

örümceğinizle birleştirerek yeni bir program yaratabilirsiniz.

Önce grafik bir patlama, "e", yaratacak aşağıdaki satırları ekleyin.

```

5 FOR x=0 TO 7
6 READ y
7 POKE USR "E" x,y
8 NEXT x
9 DATA 145,82,44,121,100,92,7

```

Şimdi de 190. satın alıp, aşağıdakileri de ekleyin ya da değiştirin.

### ÖRÜMCEKLER VE PİRAMİTLER

```

114 LET H=RND*40
115 FOR V=0 TO 20
116 PRINT AT V,H: "  RE V+1,H;
END 4
117 NEXT V
118 PRINT AT 21,H: FLASH 1-2MR
119 GOTO 61.5
120 GO TO 114

```



Artık yıldız patlamaları olmaz. Bunun yerine gökten düşecek örümcekler, piramitleri ve yeri yiyeceklerdir. Yaptığınız v ve h'nin örümceğin pozisyonunu belirlediği, bir FOR NEXT döngüsü eklemek oldu. h değişimi raststaldır. Bu nedenle örümcekler ekranın değişik yerlerinde dikey düşmeye başlarlar. Şimdi de şu satırları ekleyin.

## PATLAYAN ÖRÜMCEKLER

```

100 IF ATTR (V+1,N)=14 THEN GO
TO 500
500 PRINT AT V+1,N: FLASH 1, PA
PER 2,"E"
510 PAUSE 100
520 GO TO 114

```



Lazer ışını örümceklere isabet sağladıkça, kısa bir süre için bunlar sarı olacak. DRAW'ı oluşturduğunuz çizgi örümcek karakterinin pozisyonuna ulaşınca, mürekkep rengi, çizginin rengi olan sarıya döner. 190. satırda, ATTR örümceğin sarıya dönüştüğünü fark edince, bilgisayarı 500. satırdaki patlama alt programına gönderir.

## Zıplayan top

Bir çok grafik programda cisimler ekranın kenarından yansır. Bu program, bunun nasıl yapılacağını gösteriyor. v ve h değişimleri patlayın örümcekler programındaki gibi işler, ama toplan aşağı yukarı sağa sola gitmesi için bunlara +1 veya -1 eklenir. SCREEN, c,h pozisyonunda bir X olup olmadığını kontrol eder.

## ZIPLAYAN TOP

[illegible]

## Gönderme kullanımı.

ATTR komutu, ekrandaki belli bir pozisyona "gönderme" işi sağlar. "Gönderme" ler, mürekkep ve kağıt renginden, belli bir pozisyonda yangın söndürme ya da parlatıcı olup olmamasından kaynaklanır. Patlayan Örümler programında, ATTR, örümceğin san renk altına geçmesini sağlar. Burada gönderme mürekkep rengi (6 rakamdır). Kağıt rengi mavimsi (1 rakam)dır ve örümcek parlatıcı ya da yangın söndürme değildir. Bu durumda göndermelerin toplamı 14'tür. Programcılar Başpuru Kılavuzu bu rakamların nasıl bulunduğunu gösterir.



## MÜZİK VE ÇEŞİTLİ SESLERİN YAPIMI

ZX Spectrum+'un, programlarınıza çok çeşitli müzikal sesler ve özel ses efektleri ekleyip, canlılık kazandıracak bir ses sentesizeri vardır. Müzik bilginiz az da olsa, hiç olmasa da, kullanımı çok kolaydır. Sentesizerin çıkardığı sesler size Spectrum'un iç oparöründen ulaşır.

### Seslerin programlanması

Spectrum'unuzdan ses çıkartabilmek için tek bir komut yeterlidir: BEEP. Bunu iki sayı veya bunları temsil eden değişkenler izler. İlki bilgisayara bu sesin (saniye olarak) ne kadar süreceğini bildirir, ikincisi ise sesin tınısının ne kadar alçak ya da yüksek olduğunu gösterir. Örnekteki C(do)nin değeri 0, C(do) diyeyizinki 1, B(si) bemolunki -1'dir.

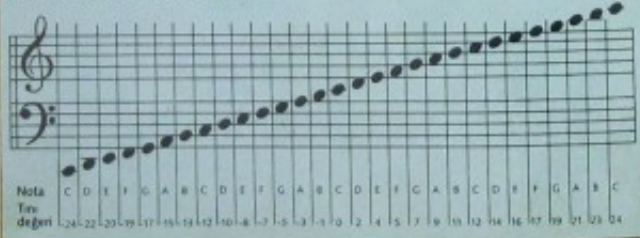
Spectrum'unuzun çıkaracağı bütün sesleri duyabilmek için şu programı verin:

```
10 FOR P=0 TO 100 STEP 1
20 BEEP P,2:P
30 PRINT AT P,0: "T:HT P,0:P
40 NEXT P
```

Spectrum en yüksekinden (69) en alçağına (-60) bütün notaları çaldı. En yüksek notaları duyamadığınızı, en alçakları ise

### Müzik yapmada tınısların değeri

İşte Spectrum'un en baştan en tize kadar bütün tınısları. Diyiz için 1 ekleyip, bemol için 1 çıkartın.



tıkırtı gibi geldiğini farkedebilirsiniz. Çünkü bu sesler insan kulağının duyabilme sınırlarının ötesindeki seslerdir. Bu sayfanın altındaki çizelge, notaların tını değerini gösteriyor. Böylece yazılı bir müziği, Spectrum'un programına dönüştürebilirsiniz.

### Ses efektleri

BEEP'i tını değerini hızla değiştirecek bir döngüye yerleştirerek, Spectrum'unuzdan çok çeşitli sesler, efektler elde edebilirsiniz. Bu programları deneyin ve bunlarla kendi seslerinizi geliştirmeye çalışın. Dikkat ederseniz, ses süreleri, saniyenin yüzde biri gibi çok kısa. Döngülü programları durdurmak için BREAK'e basın.

### COŞKULU

```
10 LET P=INT (1000*(RND-0.5))
20 BEEP P,0.02:P
30 GOTO 10
```

Bu program, üç notalık bir gruba rastsal tınılarda üst üste geliyor. Tını alanı hayli geniş, ama 10. satırdaki değerlerle oynayarak, bu alanı da değiştirebilirsiniz.

### MAKİNA

```
10 FOR K=10 TO 30
20 BEEP -0.1:P
30 BEEP -0.1,24-P
40 NEXT K
```

Bu program, bir giderek yükselen, diğeri alçalan tınıda iki ses çıkıyor. Bunun nedeni iki BEEP cümlesinin, sadece saniyenin yüzde biri kadar aralarla, farklı tınılarda iki ses üst üste çıkması.

### KALKIŞ

```
10 FOR P=10 TO 40 STEP 0.2
20 BEEP P,0.2:P
30 NEXT P
```

Bu, makina programına benzer, ancak iki nota da birbirine 6 yarımlı nota uzaklıkta birlikte yükselir. Bu arada tını değeri her defasında 0,2 - yarımlı notanın beşte biri - kadar değişir. Bu da sesin tınısının daha yavaş yükselmesine yol açar. STEP değerini değiştirerek, diğer tını değişikliklerini de deneyin.

### KLAVYE DEĞİŞİKLİĞİ

```
10 LET P=CODE INKEY$
20 IF P=0 THEN GOTO 10
30 BEEP P,0.1:GOTO 10
```

Bu program sizin herhangi bir tuşa basmanızı bekler. Basınca her biri değişik bir ses çıkarır. Bir tuşa basarken, CAPS SHIFT'e basarsanız, sesin başlangıcını göreceksiniz. Bu program, her tuşa basıldığında CODE INKEY\$, P'nin değerini değiştirdiğinden bu sonucu verir. İkinci satır ise, bir başka tuşa basılmazsa bilgisayarı susturur. 51. sayfadaki karakterler tablosunda, CODE'un vereceği değerleri görebilirsiniz.

### Spectrum'un sesini nasıl yükseltebilirsiniz?

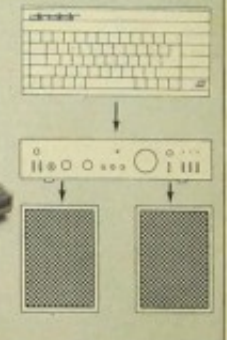
Spectrum'unuzun sesini daha yükseltmek için EAR girişine kulaklık ya da amplifikatör ve oparör bağlayabilirsiniz. Amplifikatörün ses ayar düğmesini oynatarak sesi yükseltip kısabilirsiniz.

Bunu yapmanın en kolay yolu, Spectrum'un kaset bağlantısı ile EAR girişini kasetli teypin AMIC girişine bağlamaktır. Gerekliyse kaseti çıkarıp, sonra da teypin PLAY, REWIND (geri sarma) ve FAST FORWARD (ileri

sarma) düğmelerine basın. Kasetli teypin ses düğmesini ayarlarken, bilgisayarı sesinin teypin oparöründen geldiğini duyacaksınız. Ya da kulaklığınızı da kasetli teype takabilirsiniz.

Eğer sesleri tam olarak duymak isterseniz, Spectrum'u bir müzik seline de bağlayabilirsiniz. Bunun için bir ucunda Spectrum'a girecek 3.5mm'lik bir jak-fiş, diğer ucunda müzik setinin girişine takılacak özel bir bağlantı

gerekir. Spectrum, kasetli teyp ya da diğer teyp gibi düz sinyaller ürettiğinden, amplifikatördeki REPLAY ya da LINE IN girişleri kullanılabilir. Eğer bir sorunuz olursa, bu tür malzeme satan bir mağazaya danışın.



### Ses ve Görüntü

Spectrum'un çıkardığı seslerin ekrandaki görüntü ile uyumu çok hoş olur. Programlara nasıl ses ekleyebileceğinizi görmek için 34. sayfadaki Kaçan Örümcek programına dönün.

Unutmayın 135. satıra hareketi yavaşlatmak için bir PAUSE cümlesi eklemiştiniz. Programı böyle yavaşlatacak yerde, ses çıkaracak bir duraklama da programlayabilirsiniz. 135. satırı şöyle değiştirin:

### 135 GOSUB 500

Şimdi de programa şu satırları ekleyin:

```
200 STOP
300 FOR P=40-1 TO 30-1 STEP -1
410 BEEP P,0.02:P
500 NEXT P
510 RETURN
```

Programı işleme koyun, örümcek aşağı inerken bir vizitli çıkaracaktır. Alt program, örümceğin her bir alttaki pozisyona geçişinde çok hızlı üç nota çalar. 500. satır ve 510 satırdaki 0.02'yi değiştirip notaları hızlandırıp yavaşlatarak yeni notaları ekleyin.



## PROGRAMLARINIZI NASIL SAKLAYABİLİRSİNİZ?

Bir süre sonra, kendi programlarınızı kasette saklamak isteyeceksiniz. Bunu yapmak için, Spectrum'unuza bir kasetli teyp bağlayın ve

### Kendi programlarınızı kaydedin

- 1 Önce Spectrum'unuzu 14. sayfada anlatıldığı gibi uygun bir kasetli teyp, kaset bağlantısı ile bağlayın. Ancak Spectrum'un sadece MIC girişinin kasetli teyp'e bağlı olmasına dikkat edin.
- 2 Eğer kasetli teypin bir kayıt ses kontrol düğmesi varsa, bunu üçte iki oranında açık tutun. Yoksa, merak etmeyin, ses kayıt düzeyi otomatik olarak ayarlanacaktır.
- 3 SAVE'e basın, ardından da programın adını tırnak içinde yazın-örneğin:

SAVE "prog2"



Ad için, on harf ya da rakamdan oluşan herhangi bir bölüm kullanılabilir. ENTER'e basın. SAVE satırı kaybolacak ve ekranda Spectrum'dan gelen kasetli teyp kullanma talimatlarını göreceksiniz.



bilgisayardaki programı burada saklayın. Spectrum, programı teyp'e, kasete kaydedilebilecek biçimde yollar. Bundan sonra programı her kullanmak istediğinizde, bunu 14 ve 15. sayfalarda anlatıldığı gibi bilgisayara yükleyerek kullanabilirsiniz. Bu iki sayfada, programların nasıl saklanabileceğini ve programın doğru olarak kaydedilip edilmediğini nasıl kontrol edebileceğinizi göreceksiniz.

- 4 Teypi, kayıt durumuna getirin. Bu, genellikle RECORD ve PLAY'e aynı anda basarak yapılır.
- 5 Siz beklerken, Spectrum programı aktarılır. Önce ekranın çevresinde aşağı yukarı oynayan mavi-kırmızı çizgiler göreceksiniz.



Sonra kısa süreyle mavi-sarı çizgiler görünür. Bu, Spectrum programının adını teyp'e aktarıldığını gösterir.

- 6 Bunu kısa bir ara, sonra da tekrar mavi-kırmızı çizgiler izler. Ardından, Spectrum programı teyp'e yollarken yeniden mavi-sarı çizgiler belirir. Uzun bir programın aktarımı epey zaman olabilir.



- 7 Program teyp'e aktarıldığında, OK, 0:1 mesajı görünür. Teypi durdurun. Program artık saklanmıştır. Eğer isterseniz bunu kontrol edebilirsiniz veya "doğrultabilirsiniz."

### Programınızı nasıl doğrultabilirsiniz?

Bilgisayarın programı kasetli teyp'e aktarmış olmasına rağmen, programın kasete başıyla kaydedildiğinden emin olamazsınız. Ancak, Spectrum'unuz bunu sizin için kontrol edebilir.

Bu süreç, doğrultma denir. Önce kaseti en başa sarın. Sonra Spectrum'un EAR girişini teypin EAR girişine bağlayın. (İsterseniz MIC girişine de bağlı kalabilir.) Ardından VERIFY tuşuna basıp, tırnak içinde programın adını yazın. ENTER'e basın ve teypi çalıştırın. Aynı sırayla mavi-kırmızı ve mavi-sarı çizgilerin görünmesi gerekir. Programın adı da ekranda belirecek ve doğrultma işlemi sonuna kadar ekranda kalacaktır.



İkinci mavi-sarı bölümün sonunda, şu mesaj belirecektir.

OK, 0:1

Bu, Spectrum'unuzun belleğinde olan programı, teyp'teki program ile karşılaştırıp, tastaman aynı bulunduğunu gösterir.



Artık programınızın, kasete geçtiğinden kesinlikle emin olabilirsiniz.

### Yazılım saklamada öğütler

1. Saklayacağınız programın adını kasetin etiketine ya da bir karta yazın. Büyük ya da küçük harfleri aynı ekranda görüldüğü gibi kullanın. Eğer teypinizin sayacı varsa, bundan yararlanıp programın yerini saytayı, adının yanına sayıyı yazın.
2. Programı saklamadan önce, adını bir REM cümlesi ile programa ekleyin - örneğin

S REM SPIDER program Version 3

Program işleme konunca, bilgisayar bütün REM cümlelerini gözard eder. Programın istediğiniz yerlerine REM'i kullanarak, görüş ve düğüncelerinizi ekleyebilirsiniz.

Eğer bu mesajı alamazsanız, bir aksilik var demektir. Önce 16. sayfadaki **Yazılım yüklemesindeki sorunlar** tablosuna bakın. Çünkü program kasete aktarılmış ama bir hatadan ötürü, doğrultma için bilgisayara yüklenemiyor olabilir. Eğer burada bir hata varsa, hatayı düzeltin, kaseti başa sarın ve programı yeniden doğrultun. Bilgisayar programı hâlâ doğrultmıyorsa, o zaman bir sonraki sayfadaki **Yazılım saklamadaki sorunlar** tablosuna bakın. Bu arada NEW ya da reset'e basmayın veya bilgisayarı kapatmayın, aksi halde elinizde güvenilir bir kopya olmadan bellekteki programı kaybedersiniz.

### Programın otomatik başlaması

SAVE'in ardından, programın adını ve LINE 1 yazabilirsiniz - örnek:

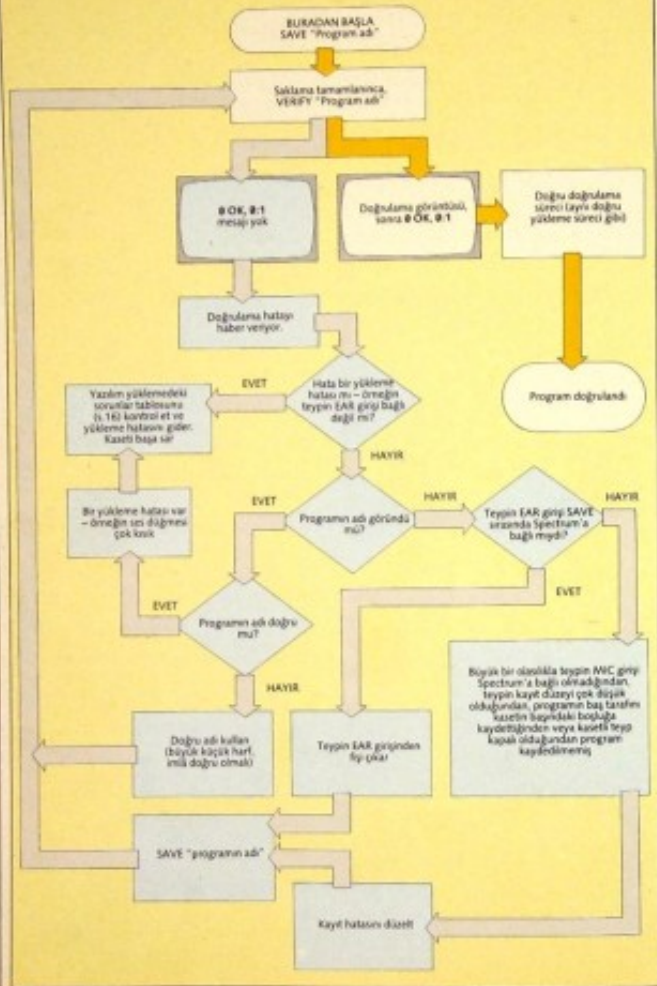
SAVE "SPIDER" LINE 1

Saklama süreci ilkinden farklı değildir. Ancak bu defa doğrulturken, VERIFY ve programın adından sonra LINE 1'i dahil etmeyin. LINE 1 ile saklanan programlar, yüklendiğinde otomatik olarak başlar. RUN'i kullanmaya gerek yoktur, (ama program başlatmaya teypi durdurmayı unutmayın).

Olan şu: Program 1. satırda başlıyor, ama 1. satır yoksa, bilgisayar hemen programın ilk satırına geçiyor. 1'i bir başka rakama çevirirseniz, program otomatik olarak bu rakama taşıyan satırdan başlar.

### CODE, SCREEN\$ ve DATA'nın saklanması

SAVE, CODE ve SCREEN\$ ile Spectrum'un belleğinin bir bölümünü ve DATA ile de bir düzünü saklamak için kullanılır. Bunlar için Programının Başvuru Kılavuzuna bakın.



## ZX SPECTRUM+'U TANIYALIM

Bu bölümde, ZX Spectrum+'un içine girecek, klavyenin altındaki çeşitli öğelerin neye yaradığını ve bilgisayarın çalışabilmesi için bunların birbirine nasıl bağlandığını öğreneceksiniz. Ayrıca yan donanımları, yani Spectrum'unuzu tam bir bilgisayar sistemi haline getirecek, eklenebilir parçaları nasıl kullanacağınızı göreceksiniz. Nihayet, burada bilgisayarınızın teknik yönleri - örneğin belleğin düzenleniş, Spectrum'un teknik özellikleri - hakkında daha çok bilgi bulacaksınız.





## İÇİNDE NELER VAR?

ZX Spectrum + 'unuzu nasıl çalıştırdığını öğrenmek için, içini açıp bakmaya kalkmayın, bu bölümü okuyun. İçini açarsanız, garantiniz geçersiz hale gelir, ciddi zarara neden olabilirsiniz.

Kasanın içinde klavyeyi Spectrum'ün diğer parçalarına bağlayan iki kurdela bağlantı vardır. Bu parçaların hepsi bir tek basılı devre tablosunun üzerine yerleştirilmiştir. Tabloda, direnç, kapasitör gibi standart elektrik parçaları da vardır, ama gen kalkanların çoğu, tek başına ya da bloklar halinde yerleştirilmiş siyah dikdörtgen mikroyongalardır.

### Yonganın içinde

Bir mikroyonganın çalışan bölümü aslında içinde bulunduğu plastik ambalajdan çok daha küçüktür. Bu kaplama, esas olarak yonga için gerekli bağlantıların yapılabilmesini ve bunun devre tablasındaki soketlere takılmasını mümkün kılacak biçimde düzenlenmiştir. Yonganın kendisi ise, üzerinde binlerce elektrik devresi olan çok ince bir silikon düzlemidir. Her devre, kendisine ulaşan elektrik sinyallerini durduran, aktaran veya depolayan bir elektrik anahtarı görevi yapar. Bu oldukça basit bir işlem olmakla birlikte, aynı anda birlikte hareket eden o kadar çok devre vardır ki, bunlar akıllamaz bir hız ve kesinlikle bilgi depolayan ya da işleyen sinyaller üretebilirler. ZX Spectrum + 'da her biri bilgisayarın işleyişinde belli rol üstlenmek üzere tasarlanmış çok farklı yongalar vardır.

### Yongalar nasıl bağlanır?

Bir bakıma, sonuçta, Spectrum'unuz son derece karmaşık bir elektrik devresidir. Yongaların ve parçaların içindeki ve arasındaki yollardan sürekli gidip gelen kalp atışı gibi elektrik akımlarından oluşan kodlu sinyaller bilgisayarın çalışmasını sağlar. Peki, nasıl oluyor da, doğru sinyal tam zamanında, tam yerine ulaşabiliyor? Bilgisayarın yongalarından birinin içine gizlenmiş bir saat vardır. Bu, çevreye kalp atışı gibi elektrik akımları saçarak çalışır - saniyede 3,5 milyon atış. Bunlar, düzenli olarak bu devrelerde dolaşarak, her parçanın hareketini denetleyen, her şeyin düzgün akmasını sağlayan kodlu sinyalleri üretir.

**ZX Spectrum + 'unuzun içi**  
Spectrum'un devre tablosunun bu çiziminde, klavye ile bağlantı sağlayan kurdela bağlantılar sökülüştür.

**Spectrum kullanırken, bir tuşa basılması klavyenin altındaki iki kablolu temasını sağlar. Bu da CPU'ya kodlu bir sinyal gönderir.**

**Bağılantısız Mantık Düzeni (ULA)**  
Bu yonga, RAM'da bulunan bilgilerden görüntüler üretir, ayrıca sistem denetçisi olarak görev yapar.

**Klavye bağlantı noktası**  
Klavyeden gelen kurdela bağlantılardan biri buraya bağlanır.

**TV kodlayıcısı**  
Bu, bilgisayarın devrelerinin ürettiği sinyalleri, renkli televizyon sinyallerine çevirir.

**Rastgele Kullanım Belleği (RAM)**  
Bu yongalar, bilgisayarın değerleri gibi, programların gereksinim duyduğu belirli bilgileri bilgisayara besleyen programı içerir. RAM'ın 48 K'lık içeriği klavyeden değiştirilebilir veya reset'e getirerek ve bilgisayar kapatılarak tımden silinebilir.

**TV çıkışı**  
Bu, televizyon alıcısına giden sinyalleri üretir.

**Teyp soketleri**  
Bunlar, bellekten bilgi ve programların bir teypte gönderilmesi ve teypten yeniden belleğe verilmesi için kullanılır.

**Mantık Yonaları**  
Bu yongalar, CPU ile RAM arasında bilgi akışını denetleyen görevi yapar.

**Merkez İşlem Birimi (CPU)**  
Bilgisayarın "beyni". CPU, bir 280 mikro işleyicidir. Butün hesapları yapar ve Spectrum'un bütün işlemlerini denetler.

**9 VDC girişi**  
Elektrik bağlantısını sağlar.

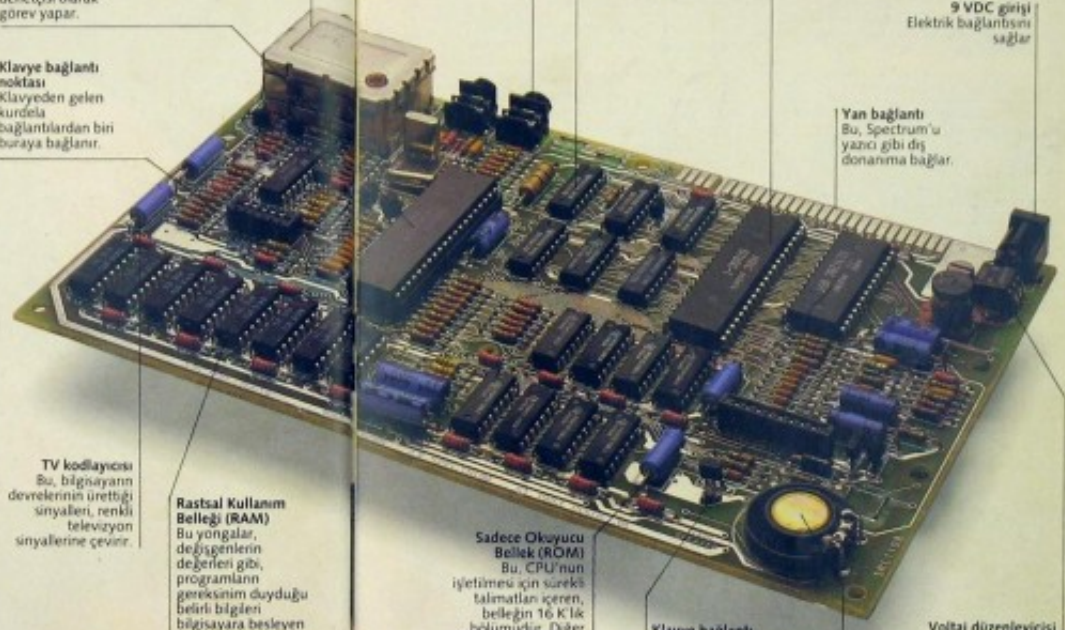
**Yan bağlantı**  
Bu, Spectrum'u yazıcı gibi dış donanıma bağlar.

**Sadece Okuyucu Bellek (ROM)**  
Bu, CPU'nun işletimisi için sürekli talimatları içeren, belleğin 16 K'lık bölümüdür. Diğer işlevlerin yanı sıra bu talimatlar, BASIC komutlarını CPU'nun anlayacağı biçime sokar. Bu bellek yongalarının içeriği, klavyeden değiştirilemez.

**Klavye bağlantı noktası**  
Klavyeden gelen kurdela bağlantılardan biri buraya bağlanır.

**Volajaj düzenleyicisi**  
Bu parça voltajdaki değişimin bilgisayarın etkilemesini önler.

**Oparlör**  
Gerektiğinde ses üretir.



## ZX SPECTRUM+ 'UNUZ NASIL ÇALIŞIR?

Diğer mikrobilgisayarlar gibi, ZX Spectrum + 'un işleyişi dört ana parçaya oluşur. Bunlar, bilgisayar bilgi veya bir program veren klavye gibi *girdi birimleri*; bilgi, program ve işletme talimatları depolayan geçici veya daimi  *bellekler*; bilgiler üzerine program talimatlarını uygulayan *Merkezi İşlem Birimi (CPU)* ve sonucu veren *çıkış birimleri* dir.

### Program verip işleme koyma

Çok basit bir program verip işleme koyunca, Spectrum'un içinde ne olur? İşte tek satırlık bir örnek.

```
10 PRINT 6+2
```

Önce klavyeyi kullanırsınız. Tuşların altı, kafes gibi tellerle örülmüştür. Her tuşa basışınızda, bir çift tel temas eder ve CPU'ya kodlu bir sinyal gönderir. Bunun üzerine CPU bu kodu depolanacağı RAM'a yollar.

Programı işleme koyunca, CPU, depolanmış kodları, programdaki sırasıyla teker teker RAM'dan alır. Önce, ona ROM'dan belli bir işlem kodu almasını söyleyen PRINT kodunu alır. Bu işlem kodu CPU'ya gider ve CUP, ekranda bir değer

## İkil kodlar

Spectrum'unuzu çalıştıran bütün kodlar ikil biçimdedir. Bunlar iki tür sinyalden oluştuğu için bunlara ikil denir. Bunlar ikil sayılar, yani iki rakamdan 0 ve 1'den oluşan sayılarla ifade edilebilir. Örneğin 6'nın ikil sayısı 00000110'dır.

Spectrum'unuzun içinde, bu kodlar hızlı elektrik abislerinden oluşan bölümler haline gelir. Ateş belli bir sürede gelmezse, bu 0'dir. Bu nedenle bilgisayar kodunda 6, Gelmedi-gelmedi-gelmedi-gelmedi-gelmedi-geldi-geldi-geldi-geldi dir.

Yandaki şekilde, farklı kodların bilgisayar tarafından bilgiyi bir yerden bir başkasına aktarmak için nasıl kullanıldığını görebilirsiniz.

yanıtsızca hareketleri yapmaya hazırlanır. CPU, RAM'dan 6 değeri alır. Bu da, bir tür kod biçimindedir ve CPU bunu "küçük" denen küçük ikili belleğe geçirir. Bunu toplama kodu izler ve CPU gene gerekli işlem kodunu ROM'dan alır. Son olarak da CPU, 2. içim RAM'dan bir kod alır. Bu kodu, "Küçük" teki değere ekleyip sonucu (8) elide eder. CPU, sonra bu sonucu başka bir elide kodu çevirip, bunu görüntü kesimine gönderir. Bu, RAM in ekranda gördüğünüz her şeyi çizen kodları bulunduran kesimdir ve ekranda 8 rakamı görünür.

Bir program depolaması

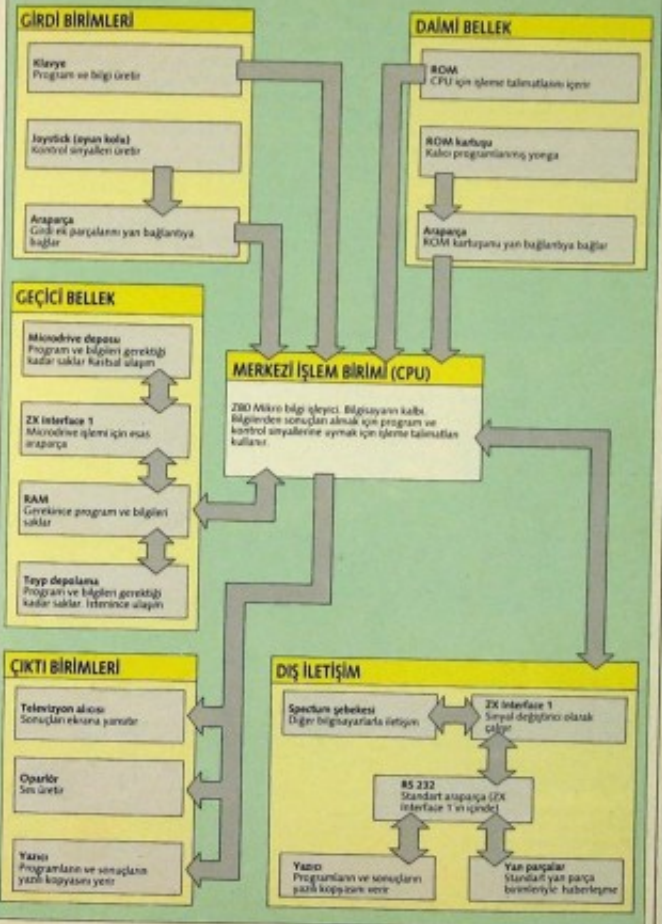
Spectrum'dan bir programı kasette saklamasını isterseniz, CPU gene kodları RAM'dan alır. Ama buna uygun hareket edecek yerde, kodları ses sinyallerine dönüştürecek bir değişim birimine gönderir. Bu sinyaller sonra teypte yollanıp, kasete kaydedilir.

Daha sonra programı yüklediğinizde, teypten gelen ses sinyalleri, değiştirim biriminde bilgisayar kodlarına çevrilir. CPU, bunları gerek olana kadar saklanacakları RAM'a yollar.

Spectrum'un girdi-çıkış yolları

Bu çizelge, kodlanmış bilgilerin, klavye girdi birimlerinden, Spectrum'un işleme sisteminden geçerek, televizyon ekranı gibi çıktı birimlerine

nasıl ulaştığını gösteriyor.  
Tek uçlu oklar, tek yönlü çalışan yolları, çift uçlu  
oklar ise iki yönlü çalışan yolları gösteriyor.





## DONANIM BAĞLANIŞI

Sinclair'in ya da Spectrum'la kullanılabilecek diğer yapımcıların yan donanımlarından yararlanarak ZX Spectrum + 'unuzu tam ve güçlü bir bilgisayar sistemi haline getirebilirsiniz. Bu sistemin merkezini ZX Interface 1 oluşturmaktadır. Programların ve verilerin daha hızlı ve daha kolay işlenmesini sağlayan Microdrive'ları ve diğer Spectrum'lar da dahil olmak üzere çok çeşitli diğer donanımları bilgisayara bağlamanızı bu mümkün kılar. Bu araparçadan yararlanarak, Spectrum'unuzu standart yazıcılara bağlayabilir, telefon hatları aracılığıyla program ve veri alıp vermenizi sağlayan modemle bağlantı kurabilirsiniz. Bilgisayara anında program yüklemek için kullanılan ROM kartuşları ile bağlantı sağlayan başka araparçalar da vardır. Oyun oynamanızı kolaylaştıran oyun

kolu(joystick) da gene bunlar aracılığıyla bağlanır.

### Spectrum'a uyan yazıcılar

Bazı yazıcılar, doğrudan Spectrum'un yan bağlantısına bağlanabilirler. Örneğin elinizde Sinclair ZX yazıcınız varsa, bunu bir araparça kullanmadan bilgisayarınıza bağlayabilirsiniz. Bu tür bir yazıcı aynı zamanda ZX Interface 1'in de arkasına takılabilir. Ancak RS232 çıkışlı bir yazıcı için, ZX Interface 1'deki D girişini kullanmanız gerekir.

### ZX Interface 1

ZX Interface 1 birimi Spectrum'un altına ve arkasına bağlanır. Bu, Spectrum'unuzu sekiz kadar Microdrive'a, 63 kadar diğer Spectrum bilgisayarına ve RS232 standart araparça birimi ile pek çok sayıda diğer standart donanıma bağlar.

Microdrive'lar ve Microdrive kartuşları, program ve verilerin depolanmasında kasetli teypin ve kasetlerin yerini alır. Microdrive kartuşu takarak, programlarınızı birkaç

sanideye depolayabilir, doğrultabilir ve yükleyebilirsiniz. Her kartuş 85K kadar veri depolayabilir ve en fazla sekiz Microdrive kullanarak, Spectrum'unuz 680K'lık bir depolama kapasitesine ulaşır!

Araparça birimi ile verilen şebeke bağlantısını kullanarak bilgisayarınızı bir başka ZX Spectrum ya da ZX Spectrum + 'a bağlayabilirsiniz. Bu şebeke, en çok 63 başka Spectrum'u kapsayacak kadar genişletilebilir. Bunların arasında saniyede 10,000 hızıyla bilgi alışverişi yapılabilir.

ZX Interface 1 birimi ayrıca 9 adet D girişli bir RS232 araparçası da içerir. Ve bu girişlerle modemler ve diğer bilgisayarlar Spectrum'unuza bağlanabilir. Ayrıca bir de standart araparça kablosu da vardır.



Standart yazıcılar ZX Interface 1 aracılığıyla bağlanır.

### ROM kartuşları ve oyun kolları

ZX Interface 2 gibi araparçalar, ROM kartuşlarının ve oyun kollarının bağlanmasını sağlar. ROM kartuşları, kasetten yüklenmesi hayli uzun zaman alacak programların derhal yüklenmesini mümkün kılar.



**Microdrive'in yüklenmesi**  
Microdrive kartuşları, sürücünün önündeki yarığa sokulur.



**ROM kartuşunun takılması**  
Kartuş, araparça soketine sokulur. Bilgisayar cereyanına bağlanınca, program bilgisayarın içindeki ROM'u atlayarak otomatik olarak yüklenir.

### Donanımların bağlanması

İşlem sırasında, ZX Interface 1 bilgisayarın altına ve arkasına gelecek şekilde yan bağlantıya girer. Bu resimde, sistemin bilgisayara bağlanmadan önceki durumu görülmektedir.



**Microdrive birimleri**  
Bir Spectrum'a bu depolama birimlerinden sekiz tane bağlanabilir.

**Kurdela kablo**  
Bu, Microdrive'i bilgisayara ZX Interface 1 aracılığıyla bağlar.

### Dipnot

ZX Spectrum + 'un klavyesine eğim verecek iki ayağı vardır. ZX Interface 1 takıldığında zaman ayakların kullanılması gerekmez.



ROM kartuşu/oyun kolu araparçası

Yan bağlantı  
Yan donanımlar bilgisayara buradan bağlanır.

**Uyarı**  
Yan donanımlar, daima cereyan bağlantısı yapılmadan bağlanmalıdır.

## ZX SPECTRUM+'UN BELLEK HARİTASI

42 ve 43. sayfalardaki Spectrum'un içini gösteren resme baktığınızda, bir ROM ve 16 daha küçük RAM yongası olduğunu göreceksiniz. Bu yongalar Spectrum'un belleğini oluşturur. Bellekte, her biri bir byte (0 ile 255 arasında bir sayı) içeren 65536 bilgi birimi bulunmaktadır. Her birim, adres denilen bir numara ile tanımlanır.

ROM, Sadece Okuyucu Bellek demektir. Belleğin bu bölümü Merkezi İşlem Birimi için işletme talimatları içerir. Bu 16 K'lık bir ROM'dur, yani  $16 \times 1024$  (16384) byte ya da adres taşımaktadır. Bu bellek, byte'ları sadece okur, yani bunlar değiştirilemez. (Değiştirilirse bilgisayar çalışmaz olur.) Herhangi bir adresteki byte'ı PEEK'i kullanarak elde edebilirsiniz.

RAM, Rastsal Ulaşım Belleği anlamına gelir ve bilgisayara verilen programları ve bilgileri içerir. Spectrum'un 48 K'lık bir RAM'i vardır, yani  $48 \times 1024$  (49152) byte ya da adres taşıyabilir. Rastsal ulaşım, buradaki herhangi bir adresteki bir byte'ın değiştirilebileceği anlamına gelir ve bu sadece POKE kullanılarak yapılabilir.

Bellekteki adresler 0'dan 65535'e kadar uzanır. Bunların ilk çeyreği ROM, geri kalanı RAM'dir.

### Sistem değişkenleri

Yandaki sütun, Spectrum'un belleğinin nasıl düzenlendiğini göstermektedir. Bunda, bilgisayar denetleyen belli kesimlerin nerede yer aldığını görebilirsiniz. Bunlardan bazılarının yerleri değiştirilebilir, bunun sınırlarını ise sistem değişkenleri belirler.

Spectrum'un sistem değişkenleri, BASIC'te kullanılan değişkenlere benzemez. Aslında bunlar, bellekteki belli adreslerde ya da yerlerde bulunan bazı yararlı değerler için verilmiş isimlerdir. Bu ismin amacı, burada depolanmış belli bir değerin anlamını hatırlatmaya yardımcı olmaktır. Örneğin RAMTOP sistem değişkeni, ROM'un en başındaki adrestir. Belleğin bu bölümünde bir BASIC programı ile değişkenlerinin değerleri bulunmaktadır. RAMTOP'un adresi 23730'dur.

### Bellek haritası

Kullanılan tanımlanmış grafikler	
GOSUB yığını	RAMTOP
Yedek	
Hesaplama yığını	STKEND
Geçici çalışma alanı	STKBOT
INPUT verileri	
Düzeltilen emir ya da satır	WORKSP
Değişkenler	E-LINE
BASIC programı	VARS
Bilgi kanalı	PROG
Microdrive haritaları	CHANS
Sistem değişkenleri	23734
Yazıcı tamponu	23552
Göndermeler	23296
Görüntü dosyası	22528
	16384
16K ROM	

48K RAM

## SINCLAIR BASIC'I ÖĞRENELİM

Bu bölüm, Sinclair BASIC'i tam olarak tanımlamaktadır. Burada, her komutun kullanış biçiminin bir özetini bulacak, Sinclair BASIC'in nasıl çalıştığını daha ayrıntılı olarak öğreneceksiniz. Burada verilen bilgi çok temel olanlardan, en ileri BASIC programlama için gerekli olanlara kadar değişmektedir. Bu, baştan sona okunacak bir bölüm değildir. Aslında bu, sizin Spectrum'un yeteneklerini sonuna kadar kullanmanızı mümkün kılacak bir programcı sözlüğüdür.





# SINCLAIR BASIC'IN KOMUTLARI İÇİN PROGRAMCININ BAŞVURU KILAVUZU

## Komut türleri

Komutlar, aşağıdaki dört gruptan birine girerler:

### Emir

Bir hareketin yapılmasını sağlayan ve doğrudan emir biçiminde komutlardır. Enter'e basarak basmaz işleme konur. Örnek - RUN, LOAD

### Cümle

Bir hareketin yapılmasını sağlayan ve bir program satırında kullanılabilecek komutlardır. Ancak program işleme girince uygulanır. Örnek - DRAW, INPUT

### İşlev

Bir tür değer üreten komutlardır. Bir emrin veya cümlenin bir parçasını oluşturur. Örnek: RND, INT.

### Mantık işlemi

Bir cümle ya da emirde mantık ifade üzere kullanılan komuttur. Belli koşulların geçerliğini veya doğruluğunu belirler ya da değiştirir. Spectrum'da üç mantık işlemi vardır - AND, OR ve NOT.

## Komut biçimi

Komut biçimi her komutun iç dizisini ifade eder - yani, komutla değerler ve değişkenler gibi diğer öğelerin doğru biçimini gösterir.

Kısaltmalar	Açıklama	Örnek
num-const	Sayısal değişmez (rakam)	24.5
num-var	Sayısal değişken (rakam da içerebilen değişken)	sum
num-expr	Sayısal ifade (bir rakam veren komutlar, sayısal değişmezler ve değişkenlerden oluşan geçerli herhangi bir bileşim)	sum * 24.5 RND.7
int-num-const	En yakın tam sayıya yuvarlanmış bir sayısal değişmez, değişken ya da ifade	
int-num-var		
int-num-expr		
string-const	Bir dizi değişmez ya da dizi (zincir karakterleri içinde karakterlerin herhangi bir bileşimi)	"ZX Spectrum +"
string-var	Dizi değişken (dizi içeren değişken)	a\$
string-exp	Dizi ifade (dizi veren komutlar, değişmezler ve değişkenlerden oluşan geçerli herhangi bir bileşim)	a\$ + "ZX Spectrum +" a\$(6 TO 8) Y * X
letter	Herhangi bir büyük veya küçük harf	B\$A\$
letter\$	Herhangi bir büyük veya küçük harfi içeren \$	B\$A\$
cond	Bir koşul veya koşul içinde bir altkoşul	X = 10 AND 1 < 10
statement	Başka bir cümle ile kullanıldığında geçerli olan herhangi bir BASIC cümlesi	IF T > 10 THEN STOP PRINTINK 2: X

[ ] Tekrarlanabilecek seçmeli bir öğe

NOT Metinde sayısal değer ve dizi değeri sayısal veya dizi öğeleri için kullanılmıştır.

Bu kılavuz, ZX Spectrum + 'da bulunan bütün BASIC komutlarının tüm tanımını içermektedir. Her birinin altında

- Komutun yeri
- Komutun türü
- Komutun amacı
- Komutun kullanımı
- Programlama biçimi

verilmektedir. Yer, amaç ve kullanım konusunda verilen ayrıntılar, kolayca anlaşılabilir. Tür ve biçim daha karmaşıktır. Bu bakımdan, bu kılavuzdan tam yararlanmak istiyorsanız, önce bu sayfadaki bilgileri dikkatle okumanız gerekir.

## Sayılar ve değişkenler

### Sayılar

9 ya da 10 haneli bir sayıyı tastamam saklayabilir. İşleyebildiği sayılar ise  $10^{38}$  ile  $4 \times 10^{-39}$  arasında yer almaktadır.

### Kabul edebildiği değişkenler

Sayı bir harfle başlamak üzere istenen uzunlukta. Boşlukları dikkate almaz ve bütün harfleri küçük harfe dönüştürür. Büyük küçük harf farkı gözlemez.

Dizi '\$' in ardından gelen herhangi bir tek harf. Büyük küçük harf aynıyı yapmaz.

Düzen Düzen değişken ve altöge (subscript) leni için, DIM maddesine bakın.

Biçimi gösterirken, aşağıdaki kısaltmalar kullanılmıştır.

## Sinclair BASIC'in işaretleri

İşaret	Yeri	İşlevi/Kullanımı	İşaret	Yeri	İşlevi/Kullanımı
\$	4	Dizi değişkeni	=	L	Eşittir
%	7	Yeni satır başlatır	Z	8	Bir program satırında cümleleri ayırır
(	8	Parantez açar	/	V	Bölme
)	9	Parantez kapatır	*	8	Çarpma
<=	Q	Daha az veya eşit	-	Kendi tuşu	Öndelik işareti
<>	W	Eşit değil	-	Kendi tuşu	Yan satırında gösterir Program cümlesindeki cümleleri ayırır
>=	E	Daha büyük veya eşit	*	Kendi tuşu	Dizi başlatıp bitirir
<	R	Daha az	-	Kendi tuşu	0 veya 16. satırda gösterir Komutlardan sonra değeriyle ayırır
>	T	Daha büyük			
↑	H	Üssüne yükseltir			
→	J	Çarpma/olumsuz			
+	K	Toplama/olumsuz Dizi kumedi			

## ZX Spectrum + character set

	0	1	2	3	4	5	6	7	8	9
0					TRUE VIDEO	INV VIDEO	PRINT sayı	EDIT	gösterge ışık	gösterge ışık
10	gösterge ışık	gösterge ışık	DELETE	ENTER	sayı	GRAPHICS MODE	INK kontrol	PAPER kontrol	FLASH kontrol	BRIGHT kontrol
20	INVERSE kontrol	OVER kontrol	AT kontrol	TAB kontrol						
30			boşluk	!	"	#	\$	%	&	'
40	(	)	*	+	-	.	/	0	1	
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	]	^	_		a	b	c	
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	-			
130										
140				¡	¢	£	¤	¥	¦	§
150	GRAPHICS G	GRAPHICS H	GRAPHICS I	GRAPHICS J	GRAPHICS K	GRAPHICS L	GRAPHICS M	GRAPHICS N	GRAPHICS O	GRAPHICS P
160	GRAPHICS Q	GRAPHICS R	GRAPHICS S	GRAPHICS T	GRAPHICS U	BMD	INKETS	PI	PN	POINT
170	SCREENS	ATTR	AT	TAB	VALS	CODE	VAL	LEN	SIN	COS
180	TAN	ASN	ACS	ATN	LN	EXP	INT	SGR	SGN	ABS
190	PRIN	IV	USR	STRS	CHRS	NOT	SIN	OR	AND	<=
200	>=	<>	LINE	THEN	TO	STEP	DRFFN	CAT	FORMAT	MOVE
210	ERASE	OPEN +	CLOSE +	MERGE	VERIFY	BEEP	CIRCLE	INK	PAPER	FLASH
220	BRIGHT	INVERSE	OVER	OUT	LPRINT	LIST	STOP	READ	DATA	RESTORE
230	NEW	BORDER	CON-TINUE	DIM	REM	FOR	GO TO	GO SUB	INPUT	LOAD
240	LIST	LET	PAUSE	NEXT	POKE	PRINT	PLOT	RUN	SAVE	RANDOM-IZE
250	IF	CLS	DRAW	CLEAR	RETURN	COPY				

**ABS** Absolute value

**Klavyede yeni**  
EXTEND MODE  
G

İşlev  
ABS bir sayısal değerin mutlak değerini, yani artı ya da eksi işaretini olmaksızın değeri verir.

**ABS'nin kullanımı**

ABS'yi bir sayısal değer izler. Ifade parantez içine alınmalıdır. Örneğin:

50 LET x=ABS (y-z)

ABS sayısal değerin mutlak değerini verir.

**Örnek**

Emir

PRINT ABS -34.2

34.2 görüntü verir.

**Bicim**

ABS num-const

ABS num-var

ABS (num-expr)

**ACS** ArcCosine

**Klavyede yeni**  
EXTEND MODE  
SYMBOL SHIFT W

İşlev

ACS, kosinüsünden bir açının değerini hesaplar.

**ACS'nin kullanımı**

ACS'yi bir sayısal değer izler. Ifade parantez içine alınmalıdır. Örneğin:

60 LET x=ACS (y-z)

ACS'yi izleyen değer (yukarıda y-z) istenen açının kosinüsüdür ve -1 ile 1 arasında olabilir. ACS açının değerini radian olarak verir. Radiana bir dereceye çevirmek için ACS'nin verdiği değeri 180/PI ile çarpın.

**Örnek**

Emir

PRINT 180/PI \* ACS 0.5

60 verir, bu kosinüsü 0.5 olan açının derecesidir.

**Bicim**

ACS num-const

ACS num-var

ACS (num-expr)

**AND**

**Klavyede yeni**  
SYMBOL SHIFT Y

Mantık işlemi/İşlev

AND, bir kodların birleşiminin gerektiklerini inşaatı mantık işlemi olarak görev yapar. Ancak bütün kodların gerektiklerini, birleşimin tümü gerektirir. AND ayrıca iki sayısal ya da dizi değeri üzerinde ikili işlemler yapmak için bir işlev olarak da kullanılır.

**AND'in kullanımı**

Mantık işlemi olarak AND, tümünün gerektiklerini inşaatı bir cümlede iki kodlu bağlar. Örneğin:

90 IF x=y & AND time < 10 THEN

PRINT "Correct"

Ancak her iki koşul da (x=y & az ve time < 10) gerçekleşirse, ekranda Correct yazısı görülecektir. Eğer biri ya da her ikisi de yanlışsa, o zaman birleşimin tümü yanlış ve program bir sonraki satıra geçer.

**İşlev olarak AND**

İşlev olarak AND iki sayısal değeri çarpabilir. Örneğin:

50 LET x=y AND z

AND ancak ikili değeri 0'a eşit değilse ikili değeri 1'i verir, ikili değeri 0 ise 0 zaman 0 verir.

AND, kendinden önce gelmesi kodlarıyla bir dizi değeri de işleyebilir. AND'ı mutlaka bir sayısal değerin izlemesi gerekir. Örneğin:

50 LET a5=b5 AND z

AND, ikinci değeri (z) sıfır değilse iki (b5) verir, ikinci değeri (z) 0 ise boş dizi (" ") verir.

Dikkat edin, ZX Spectrum + doğru bir kodla 1 değeri, yanlış olana da 0 verir. Sıfır olmayan değeri gerçek, 0'ı yanlış olarak kabul eder. Sayısal değerlerin birleşimini standart gerçek tablosu uyarınca değeri vermez.

**Örnekler**

60 LET correct=(x=y & z) AND

time < 10

70 LET score=score + 10 \* (1 AND

correct)

80 LET a5=(1 "Out Of Time Or

Not" AND NOT

correct) + "Correct"

Eğer 60. satırdaki iki koşul gerçekleşirse, sayısal değeri correct'e 1 değeri verir. O zaman score 10 artar ve a5 "Correct" olur. Eğer koşullardan biri yanlışsa, o zaman correct'in değeri 0 olur, score değeri 10 artmaz ve a5 "Out Of Time Or Not Correct" (zamanınız ya da doğru değil) olur.

**Bicim**

cond AND cond

num-expr AND num-expr

string-expr AND num-expr

**ASN** ArcSine

**Klavyede yeni**  
EXTEND MODE  
SYMBOL SHIFT Q

İşlev

ASN, sinüsünden bir açının değerini hesaplar.

**ASN'nin kullanımı**

ASN'yi bir sayısal değer izler. Ifade parantez içine alınmalıdır. Örneğin:

60 LET x=ASN (y-z)

ASN'yi izleyen değer (yukarıda y-z) istenen açının sinüsüdür ve -1 ile 1 arasında olabilir. Bu durumda ASN açının

değerini radian olarak verir. Radiana bir dereceye çevirmek için ASN'nin verdiği değeri 180/PI ile çarpın.

**Örnek**

Emir

PRINT 180/PI \* ASN 0.5

Ekran 30 görünür. Bu sinüsü 0.5 olan açının derecesidir.

**Bicim**

ASN num-const

ASN num-var

ASN (num-expr)

**AT**

**Klavyede yeni**  
SYMBOL SHIFT I

Bak INPUT/LPRINT/PRINT

**ATN** ArcTangent

**Klavyede yeni**  
EXTEND MODE  
SYMBOL SHIFT E

İşlev

ATN bir açının değerini tanjantından hesaplar.

**ATN'nin kullanımı**

ATN'yi bir sayısal değer izler. Ifade parantez içine alınmalıdır. Örneğin:

60 LET x=ATN(y-z)

ATN'yi izleyen değer (yukarıda y-z) istenen açının tanjantıdır. ATN değeri radian olarak verir. Radiana bir dereceye çevirmek için ATN'nin verdiği değeri 180/PI ile çarpın.

**Örnek**

Emir

PRINT 180/PI \* ATN 1

Ekran 45 görünür. Bu, tanjantı 1 olan açının derecesidir.

**Bicim**

ATN num-const

ATN num-var

ATN (num-expr)

**ATTR** ATTRibute

**Klavyede yeni**  
EXTEND MODE  
SYMBOL SHIFT

İşlev

ATTR ekrandaki belirli bir karakter pozisyonunun gördüklerini verir. Bu gördükleri, bu pozisyonun karakterinin murekkep ve kağıt rengi, parlaklığı ve yazı yönüne durumudur.

**ATTR'nin kullanımı**

ATTR'yi bir birinden vücutla aynı ve tümü parantez içine alınmış iki sayısal değer izler. Örneğin:

150 IF ATTR(y-h)=115 THEN

GOSUB 2000

ATTR'yi izleyen ilk değer (yukarıda y) 0 ile 23 arasında olabilir ve ekrandaki bir pozisyonun satır numarasıdır. İkinci değer (yukarıda

h) 0 ile 31 arasında olabilir ve ekrandaki pozisyonun sütun numarasıdır. Bunun üzerine ATTR 0 ile 255 arasında bir rakam verir. Bu rakam, belli bir pozisyonun gördüklerini topladığı ve böyle hesaplanır:

Murekkep rengi: Renk kodu (0-7)

Kağıt rengi: Renk kodunun 8 katı

Parlak: 64

Yazı yönü: 128

**Örnek**

Eğer 11.16 pozisyonundaki karakterin murekkep rengi 3 (mor), kağıt rengi 6 (sarı) ve karakter parlak ama yazı yönü yoksa, o zaman

PRINT ATTR (11,16)

emir, 115 verir: (3 + 8\*6 + 0)

**İkili biçimde ATTR**

ATTR bir byte a verildiğinde, 7. bit (en önemli) yazı yönüne için 1, normal için 0, 6. bit parlak için 1, normal için 0, 5. bit 3. aradaki iki rakam yazı yönü kağıt rengi, 2. ile 0. aradaki murekkep rengidir.

**Bicim**

ATTR(num-expr,num-expr)

**BEEP**

**Klavyede yeni**  
EXTEND MODE  
SYMBOL SHIFT Z

Cümle/Emir

BEEP, verilen uzunlukta ve tonda bir sesin oparörden çıkmasını sağlar.

**BEEP'in kullanımı**

BEEP bir program cümlesi ya da doğrudan bir emir olarak kullanılabilir. BEEP'i vücutla aynı iki sayısal değer izler. Örneğin:

80 BEEP x,y

İkili değeri (x) 0 ile 10 arasında olabilir ve notanın saniye olarak süresi belirler. İkinci değeri (y) -60 ile 69 arasında olabilir ve yazı tonları halinde notanın süresi belirler. Ortadaki C'nin (do) alındıysa eksi, üstündeyse artı değeri verir.

**Örnek**

Emir

BEEP 0.5,1

Ortak C'nin (do) bir üstündeki C diyez sesinin yarı saniye süreyle çıkmasını sağlar.

**Bicim**

BEEP num-expr, num-expr

**BIN** Binary number

**Klavyede yeni**  
EXTEND MODE  
B

BIN bir iki sayısal değeri sayıya çevirir.

**BIN'in kullanımı**

BIN'ı, 1. terden ve 0'lardan oluşan en fazla onaltı haneli bir iki sayı izler. Örneğin:

**50 POKE USR "a", BIN 1010 1010**

BIN bu iki sayısal değeri karşılığı verir. Bu, genellikle yukarıda da olduğu gibi POKE ve USR ile birlikte, kullanımları tanımladığı grafiklerini yapımında kullanılır. Burada 1 murekkep, 0 ise kağıt renginde bir pixel e işaret eder.

**Örnek**

Emir

PRINT BIN 111111110

Ekran, bu iki sayısal değeri karşılığı olan 254 görünür.

**Bicim**

BIN [1] [0]

**BORDER**

**Klavyede yeni**  
B

Cümle/Emir

BORDER, ekranın çevresindeki alanın rengini belirler.

**BORDER'in kullanımı**

BORDER, doğrudan bir emir ya da bir program cümlesi olarak kullanılabilir. Bunu bir sayısal değer izler. Örneğin:

30 BORDER RND \* 7

BORDER'i izleyen değeri en yakın tam sayıya yuvarlanacak ve bu da aşağıda gösterildiği gibi ekran çerçevesinin rengini verecektir:

0 Siyah

1 Mavi

2 Kırmızı

3 Mor

4 Yeşil

5 Açık mavi

6 Sarı

7 Beyaz

Ayrıca BORDER ekranın alt bölümünün kağıt rengini de belirler.

INK ve PAPER dan farklı olarak BORDER cümlesi, bir PRINT cümlesinin içine yerleştirilemez.

**Bicim**

BORDER int-num-expr

**BRIGHT**

**Klavyede yeni**  
EXTEND MODE  
SYMBOL SHIFT B

Cümle/Emir

BRIGHT, karakterlerin ekranda normalden daha parlak görünmesini sağlar.

**BRIGHT'in kullanımı**

BRIGHT doğrudan bir emir olarak kullanılabilir, ama genel kullanımı bir program cümlesi biçimindedir. Bunu bir değer izler. Örneğin:

80 BRIGHT 1

Gerekirse BRIGHT'ı izleyen değeri en yakın tam sayıya yuvarlanabilir. Bu sayı 0, 1 veya 8 olabilir. Değeri 1 olursa, bunu izleyen PRINT veya

INPUT cümleleri de ekrana yazdıracak bütün karakterlerin murekkep ve kağıt renginin daha parlak görünmesini sağlar. Değeri 8 olursa, yeni karakterler yazıldığı anda, ekranda parlak karakterlerin bulunduğu pozisyonların, parlak, normalden bulunduğu pozisyonların da normal kalmasını sağlar. BRIGHT'ı izleyen 0 ise hem BRIGHT'ı hem de BRIGHT'ı izleyen, bundan sonra gelen bütün karakterler normal görünür.

BRIGHT ayrıca PRINT, INPUT, PLOT, DRAW ve CIRCLE gibi ekranda görüntü yaratan komutlarla oluşturulan cümlelere de yerleştirilebilir. BRIGHT komutları sonra, ama veri ya da görüntü öğelerinden önce gelir. BRIGHT'ı kendi değeri ve bir noktalı virgül izler. Örneğin:

50 PRINT BRIGHT 1:

"WARNING"

Bu durumda BRIGHT'in etkisi eniririr ve ancak kendisini izleyen karakterleri, ya da bu cümle ile komutları, yazıya geçirir. BRIGHT'ı izleyen sonra, ama veri ya da görüntü öğelerinden önce gelir. BRIGHT'ı kendi değeri ve bir noktalı virgül izler. Örneğin:

50 PRINT BRIGHT 1:

"WARNING"

**CAT** CAtalogue

Microdrive dosya işleme emri. Microdrive ve Interface 1 in el kitabına bakın.

**CHR\$** Character string

**Klavyede yeni**  
EXTEND MODE  
U

İşlev

Spectrum'un karakter tablosu, klavyede bulunan karakter ve komutların yanı sıra kullanımları tanımladığı grafik karakterlerinden oluşur. CHR\$ ve bir kod numarası kullanarak, bunlardan herhangi birini bir dizi olarak elde edilebilir. Karakter tablosu ayrıca, karakterlerin ekrana yazılışını etkileyen bazı kontrol kodları da içerir. CHR\$ den önce PRINT kullanılarak bu kodlar harekete geçirilip karakterler ekrana yazdırılabilir. Karakter tablosunun tamamı ve kod numaraları 51 sayfa da bulabilirsiniz.

**CHR\$'nin kullanımı**

CHR\$'yi bir sayısal değer izler. Örneğin:

80 PRINT CHR\$(x)

Ifade parantez içine alınmalıdır. CHR\$'yi izleyen değeri (yukarıda x) en yakın tam sayıya yuvarlanır. 32 ile 255 arasında ise, CHR\$ bir dizi











**Örnek**  
Emir







## MOVE

Microdrive dosya işleme emri. Microdrive ve Interface 1 el kitabına bakın.

## NEW

Klavyede yeni A

Emir/Cümle

NEW, BASIC bellek alanını (RAMTOP'a kadar olan alan) temizlerken, belleğin bu komuda bulunan herhangi bir program yok eder.

**NEW'nin kullanımı**  
NEW normal olarak bir doğrudan emir olarak kullanılır ama bir program cümlesi de oluşturabilir. Kendi başına kullanılır. Uygulamaya konulan program ve değişkenler yok olur. Bellek RAMTOP'a kadar temizlenir, böylece RAMTOP'un altında depolanmış olan kullanımların tanımladığı grafik karakterler bundan etkilenmez.

Bilgisi

NEW

## NEXT

Klavyede yeni N

Cümle/Emir

NEXT, bir FOR NEXT döngüsü yaratılmak için her zaman FOR ile bağlantılı kullanılır.

**NEXT'in kullanımı**  
NEXT, bir FOR NEXT döngüsünü tamamlamak için bir program cümlesi oluşturmak üzere kullanılır. NEXT'i, döngüde kontrol değişkeni olan bir harf izler. Örneğin,

90 NEXT A

Sindair BASIC'te kontrol değişiminin mutlaka konulması gerekir.

FOR NEXT döngüsünün ayrıntıları için FOR'a bakın.

Bilgisi

NEXT letter

## NOT

Klavyede yeni SYMBOL SHIFT S

Mantık işlemi/İşlev

NOT, bir koşulun gerçekleşmesini ters yüz etmek, yanlış bir koşulu doğru, doğruyu yanlış hale getirmek için kullanılır.

**NOT'un kullanımı**  
NOT'u bir koşul veya bir sayısal değer izler. Örneğin,

90 IF NOT X=Y+2 THEN PRINT "Wrong"

90 LET correct=X=Y+2; IF NOT correct THEN PRINT "Wrong"

NOT'u bir koşul (yukarıda X=Y+2) iflediği zaman, Spectrum bu koşulu doğru ise bina 1, yanlışsa 0 değeri verir. NOT bunun üzerine işlev gibi hareket eder, önce ortaya çıkan değeri tersine çevirir. Koşulun teminin sonmasını sağlar. Dikkat edin, koşul AND veya OR içine girerse, o zaman bunun paranteze alınması gerekir.

Eğer NOT'u bir sayısal değer izliyorsanız, bu değer sıfır değeri NOT 0. 0 ise 1 sonucunu verir. Bu nedenle yukarıdaki örnekte X<Y+2 ise Spectrum "Wrong" (yanlış), diğer değeri 0 ise correct (doğru) sonucu verir.

Bilgisi

NOT cond

NOT num-expr

## OPEN

Microdrive dosya işleme emri. Microdrive ve Interface 1 el kitabına bakın.

## OR

Klavyede yeni SYMBOL SHIFT U

Mantık işlemi/İşlev

OR, bir koşullar bileşkesinin doğruluğunu sınamak için bir mantık işlemi olarak hareket eder. Eğer koşullardan bir veya birkaçı doğru ise o zaman bütün bileşke doğru demektir. OR ayrıca iki sayısal değer üzerinde iki işlemler yapmak için de fonksiyon olarak görev yapar.

**OR'un kullanımı**  
Bir mantık işlemi olarak OR, tümünün gerçekleşmesi için cumledeki iki koşulu bağlar. Örneğin,

70 IF INKEY\$="N" OR INKEY\$="n" THEN STOP

Eğer bu koşullardan bir veya her ikisi de doğru ise tüm bileşke doğru demektir. Yukarıdaki satırda, CAPS SHIFT veya CAPS LOCK tuşlarına basılısa da, basılımsa da N tuşuna basılır basılmaz koşullardan biri INKEY\$="N" ve INKEY\$="n" doğru hale gelecektir. Bu durumda bütün bileşke doğrudur ve program durur.

**İşlev olarak OR**  
ZX Spectrum'a, doğru koşula 1 sayısal değeri, yanlış koşula ise 0 sayısal değeri verir. Sıfır dışındaki her değeri doğru, 0'ı yanlış kabul eder. Bu yüzden OR'dan önce veya sonra bir sayısal değer gelir. Örneğin,

40 LET X=Y OR Z

Bu durumda, Z sıfır değeri ya da doğru bir koşula X değeri 1 değeri, diğer Z 0 ise veya yanlış bir koşul ise Y'nin değeri verilir. Bu anlamda yararlıdır. Aşağıdaki örnekte, yokunun yanı-

14'ten küçükse bilet ücreti yan-

60 PRINT fare(0.5 OR age > 13)

Eğer yaş (age) 14'ten küçükse, age > 13 koşulu yanlış, ücret (fare) 0.5 ile çarpılır. Eğer age > 13 doğru ise ücret 1 ile çarpılır.

Dikkat edin, Spectrum sayısal değerlerin bileşkesini, standart gerçek tablosu gibi değerlendirmez.

Bilgisi

cond OR cond

num-expr OR num-expr

## OUT

Klavyede yeni EXTEND MODE SYMBOL SHIFT T

Cümle/Emir

OUT, bir çıktı donanımını harekete geçirmek için belli bir gövde/çıkış yüklemesi adresine bir byte gönderir.

**OUT'un kullanımı**  
OUT bir program cümlesi ya da doğrudan emir olarak kullanılabilir. Bunu birbiriinden gövde/çıkış aynı iki sayısal değer izler. Örneğin,

40 OUT 254,3

Her iki değeri de en yakın tam sayıya yuvarlanır. İlk değeri (yukarıda 254) 0 ile 65535 arasında olabilir ve yüklemesi adresini gösterir. İkinci değeri (3) 0 ile 255 arasında olabilir ve bu adrese gönderilecek byte'tir.

254 yüklemesi adresindeki byte'ın 0 ile 2 arasındaki bit'leri çerçeve rengini belirler. Bu nedenle yukarıdaki örnek, çerçeve kenarını mora dönüştürür. Bu adresteki 3. bit MIC girişini, 4. bit optik girişini hareket ettirir. 254, 247 ve 239 numaralı yüklemesi adresleri diğer yan donanımları kullanılır.

Bilgisi

OUT int-num-expr,int-num-expr

## OVER

Klavyede yeni EXTEND MODE SYMBOL SHIFT N

Cümle/Emir

OVER, bir karakterin bir diğerinin üzerine yazılmak için kullanılır. Ayrıca murekkep rengi yerine kağıt renginde noktalar konulmak, doğru ya da eğri çizmek için de kullanılabilir.

**OVER'in kullanımı**  
OVER normal olarak bir program cümlesi olarak kullanılır. OVER'i bir sayısal değer izler. Örneğin,

80 OVER 1

OVER'i izleyen değeri en yakın tam sayıya yuvarlanır ve bu değeri 0 veya 1 olabilir. OVER 0 zaten makinenin normal olarak ayarlanmış olduğu durumdur ve yeni karakter bir-

karakter pozisyonundaki eski karakteri silip yeni alır. OVER 1 ise aynı karakter pozisyonunda iki karakterin üstüste birleşmesini sağlar.

OVER'in, aynı INK gibi, bir PRINT veya INPUT cümlesinin içine yerleştirilerek, sadece bu cümlede yer alan karakterleri etkilemesi sağlanabilir. Örneğin şu cümle, bir karakterin altını çizler:

60 PRINT AT 11,15: "YES"; OVER 1; AT 11,15

Ancak dikkat edin, burada murekkep renklerinin (çıkış) yere kağıt rengi verimlidir.

**Yüksek tanımlı çizimde OVER**  
OVER, PLOT, DRAW ve CIRCLE ile de kullanılabilir. OVER olmadan da çiziler, eğriler çizebilir, ancak murekkep renkleri aynı değeri, kesiklikleri karakter pozisyonunun tümünün rengi değeri. Eğer OVER 1 kullanırsanız, çizgiler ve eğrilerin kesiklikleri ya da diğer karakterlerle çakıştıkları yerler kağıt rengini alır. OVER 1 kullanarak, aynı pozisyona noktalar koyup, çizgiler, eğriler üzerine ekler silinir.

Bilgisi

OVER int-num-expr

## PAPER

Klavyede yeni EXTEND MODE SYMBOL SHIFT C

Cümle/Emir

PAPER, ekrandaki görüntü için kağıt ya da fon rengi seçmede kullanılır. Bu renk ya bütün görüntü alanını, ya da sadece belirli karakterlerin, nokta ya da çizgilerin bulunduğu tek bir karakter pozisyonunu kapsar.

**PAPER'in kullanımı**  
PAPER, bir program cümlesi oluşturmak üzere ya da doğrudan bir emir olarak kullanılır. PAPER'i bir sayısal değer izler. Örneğin,

80 PAPER x

PAPER'i izleyen değeri en yakın tam sayıya yuvarlanır ve 0 ile 9 arasında bulunur. PAPER'in renkleri aynı INK gibidir. Kağıt rengi genel olarak beyazdır, fakat murekkep renkleri gibi, bütün görüntü yaratacak cümlelerin içine yerleştirilerek belli bir yerle sınırlamak mümkündür. Ayrıntılar için INK'e bakın.

PAPER çizimleri ardından bazı karakterlerin yazılması sağlanacaksa, genel de olsa doğru ya da yanlış, bu karakterlerin yazıldığı karakter pozisyonlarının rengi, seçilen renge döner. Bu ayrıca bir genel emir ardından değil de, sadece bir PAPER cümlesini izleyen nokta konulmuş, doğru ya da diğer cümle işlemleri için de geçerlidir.

Bütün bir ekranın fonunu belli bir renkle kaplamak için PAPER cümlesi ardından CLS ile kullanılması gerekir. Bütün görüntü-

alanını bir renk kaplar ve fon rengi olarak kalır.

Bilgisi

PAPER int-num-expr

## PAUSE

Klavyede yeni A

Cümle/Emir

PAUSE bir programı belirli ya da belirsiz bir süre için alıkoymayı sağlar.

**PAUSE'un kullanımı**  
PAUSE normal olarak bir program cümlesi olarak kullanılır. Bunu sayısal bir değeri izler. Örneğin,

130 PAUSE 100

PAUSE'u izleyen değeri, en yakın tam sayıya yuvarlanır ve 0 ile 65535 arasında olabilir. Televizyon ekrandaki görüntüde kaç karelik bir duraklama olacağını belirler. Bu nedenle kare frekansı 50 Hz olan İngiltere ve Avrupa da, 50 değeri 1 saniyelik bir duraklama neden olur.

Ancak, duraklama devam ederken herhangi bir tuşa dokunup bu süreyi kısaltmak mümkündür. PAUSE 0 ise bir tuşa basılana kadar sürecektir süresi duraklamayı sağlar.

Bilgisi

PAUSE int-num-expr

## PEEK

Klavyede yeni EXTEND MODE O

İşlev

PEEK, belleğin belli bir adresinde depolanmış byte'ın değerini verir.

**PEEK'in kullanımı**  
PEEK'i bir sayısal değeri izler. Örneğin,

80 LET x=PEEK(256+y)

Dikkat edin ifade paranteze alınmalıdır. PEEK'i izleyen değeri, genellikle en yakın tam sayıya yuvarlanır ve bellekteki adresi vermesi için 0 ile 65535 arasında olur. PEEK bunun üzerine bu adresteki byte'ın değerini (0 ile 255) arasında bir rakam olarak verir.

## Örnek

Spectrum'un son çıkartımından bu yana meydana gelen televizyon görüntü karesinin numarası 23672 ile 23674 arasındaki adreslerde. Bu kaneller belli bir hızla çalışır için, bu yerlerin PEEK edilmesi bir zaman ölçüm yöntemi verir. Aşağıdaki sabir, Spectrum'un certyana beğlenmesinden bu yana geçen zamanı saniye olarak verir. (Buna sayı üretmekle ve kaselli teyp, yazıcı gibi yan donanımları çalıştırmakla geçen zaman dahil değildir.)

10 PRINT (PEEK 23672+256\*PEEK 23673+65536\*PEEK 23674)/30

Not: Eğer kullandığınız elektrik frekansı İngiltere'deki gibi 50 Hz değil de 60 Hz ise, o zaman 50'yi 60'a çevirin.

Bilgisi

PEEK int-num-const

PEEK int-num-var

PEEK (int-num-expr)

## PI

Klavyede yeni EXTEND MODE M

İşlev

PI, hesaplamalarda kullanılan pi sayısının (pi) değerini verir. PI, bir dairenin çemberinin çapına oranıdır.

**PI'nin kullanımı**  
PI bir cümlede ve bir emir olarak kullanıldığında değerlere ya da değişkenlere gereksinim duymaz. Örneğin,

DRAW 244,0,-PI

Yukarıdaki emrin ekranda büyük bir yarı daire çizmeye için PI 3.1415927 değerini verir.

Bilgisi

PI

## PLOT

Klavyede yeni Q

Cümle/Emir

PLOT, yüksek tanımlı çizimlerde ekrandaki belli bir pozisyona bir pixel ya da renkli bir nokta yerleştirmek için kullanılır.

**PLOT'un kullanımı**  
PLOT, bir program cümlesi veya bir emir olarak kullanılır. Normal olarak bunu, birbiriinden gövde/çıkış aynı iki sayısal değeri izler. Örneğin,

50 PLOT 128,87

Eğer gerekiyorsa, PLOT'u izleyen her iki değeri de en yakın tam sayıya yuvarlanır. İlk değeri 0 ile 255 arasında değeri ve dikey koordinatı verir. Bunun üzerine belli bir pozisyonda (yukarıdaki örnekte ekranın ortasında) mevcut murekkep rengi ile bir nokta yerleştirir.

Renk cümlesi ya da emri PLOT üzerinde aşağıdaki etkileri yapar. OVER 1'den sonra aynı pozisyondaki mevcut noktaları rengi, kağıt rengine döner. INVERSE 1'den sonra mevcut kağıt renginde nokta yerleştirir. BRIGHT 1 veya FLASH 1'den sonra alçak tanımlı ekranda, pixel in yerleştirildiği karakter pozisyonunun tümü parlak veya yanar söner hale gelir. Bu dört komut ve INK, PRINT ile olduğu gibi bir PLOT cümlesine de yerleştirilebilir. Örneğin,



## 160 PLOT INK 2: x, y

İki ayrı ama, bu cümle ile yerleştirilen pixel ve çevresi ile renklendirir. Bir PLOT cümlesine PAPER yerleştirilirse, pixel çevresindeki karakter pozisyonunun tümünü kağıt rengi, verilene renge döndürür.

Dikkat edin, PLOT aynı zamanda bir sonraki DRAW cümlesinin başlangıç pozisyonunu da tanımlar.

**Biçimi**  
PLOT (statement)  
int-num-exp; int-num-exp

## POINT

**Klavyede yeni**  
EXTEND MODE  
SYMBOL, SHAFT B

İşlev

POINT, yüksek tanımlı bir ekranda belirli bir pozisyonun rengini mürekkep rengi mi, kağıt rengi bir olduğunu anlamak için kullanılır. POINT, rengin hangisi olduğunu araştırır.

## POINT'ın kullanımı

POINT, birbiriinden virgüle ayrılmış ve hepsi paranteze alınıp iki sayısal değer alır. Örneğin,

240 IF POINT (x, y) = 1 THEN  
GOSUB 600

POINT'ı izleyen iki değer, genellikle tam sayıya yuvarlanır. 0 ile 255 arasında değişen iki değer ekrandaki pixel'in yatay koordinatını verir. 0 ile 175 arasında değişen ikinci değer ise dikey koordinatı verir. Bunun üzerine POINT, eğer belirlenen pozisyondaki pixel mürekkep rengindeyse 1, kağıt rengindeyse 0 verir.

**Biçimi**  
POINT (int-num-exp; int-num-exp)

## POKE

**Klavyede yeni**  
O

Cümle/Emir

POKE, bellekte belli bir adresdeki byte'in değerini değiştirmek için kullanılır. Bellekteki belli yerlere POKE ile değerler, normal olarak, BASIC komutlarının sağlandığı bir hanelerle yapılmak için verilir.

## POKE'in kullanımı

POKE bir program cümlesi ya da emir olarak kullanılır. Buna bir virgüle ayrılmış iki sayısal değer gelir. Örneğin,

POKE 21609, 255

POKE'ı izleyen iki değer gerekirse en yakın tam sayıya yuvarlanır. İlk değer, 16384 ile 65535 arasıdır ve RAM'daki bir adrestir. İkincisinin değeri ise 0 ile 255 arasında değişir ve belirlenen adrese yazılacak byte'dir.

Yukarıdaki örnekte, 255, 21609 sayılı adrese POKE ile verilecektir. Bu adres, tuşa basıldığında okunan ses kontrol etmektedir. 255 değeri, normal kık sesinin yerine uzun bir bip sesinin çıkmasını sağlar. Diğer değerler daha kısa bir ses çıkarırlar.

**Biçimi**  
POKE int-num-exp,  
int-num-exp

## PRINT

**Klavyede yeni**  
P

Cümle/Emir

PRINT, verileri ekrana yansıtır. Veri bir tek veya bir dizi karakter olabilir. PRINT cümlesi, verinin pozisyonunu ve rengini belirleyen diğer komutları da içerebilir.

## PRINT'in kullanımı

PRINT kendi başına kullanılabilir gibi, bunu veriler de izleyebilir. Bu veriler, herhangi bir sayısal ya da dizi ifade veya bunların bir bileşimi de olabilir.

PRINT'ı veri ile birlikte kullanırken, iki ve daha fazla öge birbirinden virgüller, noktalı virgüller ve kesme işaretleriyle ayrılır.

PRINT ile verilen arasında diğer bazı komutlar da yerleştirilebilir. Ancak bir komuta olacağı her cümle bir noktalı virgüle bitmelidir. Böyle kullanılacak komutlar: CHR\$, TAB, AT, INK, PAPER, FLASH, BRIGHT, INVERSE ve OVER.

## PRINT'le diziler

PRINT kendi başına ya da bir boş dizi (" ") ile kullanıldığında, bir boş satır oluşturur ve göstergeç bir sonraki satırın başına geçer.

PRINT'ı bir değişmez dizi (bina) içeren içinde herhangi bir karakteri izlerse, ekrana bu tek karakteri içeren karakterler yansıtır. Örneğin şu emir

PRINT "3/542/76/21"

ekrana şu görüntüyü yansıtır:

3/542/76/21

PRINT'ı izleyen bir dizi değişken veya ifade bunların bembel ettiği dizi veya dizi ekranı yansıtır.

## PRINT'le sayılar

PRINT'ı herhangi bir sayısal ifade izlerse, ekrana bu ifadenin değeri yansır. Sayılar ondalık sayılar halinde 8 haneli olarak verilir, ondalık işaretinden sonra gelen sıfır yansımaz.

Çok büyük ya da çok küçük rakamlar, birbirinden 8 haneli ile ayrılmış iki sayı halinde, daha kısa bir bilimsel ifade ile yansır. Bu, ilk bölümün (kök) ikinci bölümdeki (uz) kuvveti yüksekliği anlamına gelir. Şu emir, örneğin,

PRINT 1/542/76/21

ekrana şu görüntüyü verir:

## 1.4680798E-6

## PRINT'in noktalama işaretleriyle kullanımı

PRINT'ı izleyen veri ögesi birbiriinden noktalı virgüle ayrılmış, ekranda bunlar aralarında hiç boşluk olmadan yan yana görünürler.

PRINT 1,2,3

emir, şu görüntüyü verir:

123

PRINT'ı izleyen veri ögesi birbiriinden virgüle ayrılmış, ekranda her öge, ilk ögenin pozisyonuna göre, ya bir satırın başında ya da ortasında görünür.

PRINT 1,2,3

emir, şu görüntüyü verir:

1 2

PRINT'ı izleyen veri ögesi birbiriinden kesme işaretleriyle ayrılmış, ekranda her bir veri bir satırın başında görünürler.

PRINT 1,2,3

emir, şu görüntüyü verir:

1 2 3

Bir PRINT cümlesi veya emir noktalı virgül, virgül veya kesme işaretleriyle sona ererse, bir sonraki PRINT cümlesinde yer alan öge de bu durumdan ayrı biçimde etkilenebilir.

## PRINT ve diğer komutlar

PRINT'ı sırasıyla TAB, bir sayısal değer, bir noktalı virgül ve bir de veri ögesi izleyebilir. Örneğin,

60 PRINT TAB x; a\$

TAB'ı izleyen değer (yukarıda x) gerekiyorsa, en yakın tam sayıya yuvarlanır, ardından 32'ye bölünür ve kalan 0 ile 31 arasında bir değer verecek biçimde ekrana yansır. Bundan sonra, aynı satırda veya bunu izleyen satırda bir veri ögesi hep bu satır pozisyonunda yansır.

PRINT'ı AT, virgüle ayrılmış iki sayısal değer, bir noktalı virgül ve bir veri ögesi izleyebilir. Örneğin,

50 PRINT AT 11,16: "Data"

İkinci değer (yukarıda 1) 0 ile 21 arasında ve verinin yansıtılacağı satır ya da satırın göstergeci. İkinci değer (0 ile 31 arasında değişir ve veriden önce karakter veya rakamın yansıtılacağı sütunun numarasını kabul edilir ve en yakın tam sayıya yuvarlanır. PRINT AT 11,16: "Data" emir, ekranın ortasına bir yıldı yansıtır.

PRINT'ı ayrıca bir veya daha fazla CHR\$ fonksiyonu da izleyebilir. Bunun ayrılmış için CHR\$ ile bulun.

PRINT ve renk komutları  
PRINT'in sağlanacağı görsel, INK, PAPER, FLASH, BRIGHT, INVERSE

ve OVER'in verildiği ve uygulanmakta olan renk emirleri ve cümlelerinden etkilenebilir. PRINT'ı, bu alt cümleden bir veya birkaç noktalı virgüle ayrılmış olarak izleyebilir. Veri, bunlardan sonra gelir. Örneğin,

50 PRINT AT 11,16: INK 2: FLASH 1: "

Bunun üzerine veri ögesi, renk komutlarının belirlendiği görsellerle ekrana yansır. Bu görseller emirlerdir ve sadece o ögeye uygulanır. PRINT cümlesinin uygulamaya konmasıyla, en baştaki veya bundan bir önceki genel renklere döner. PRINT ayrıca veriyi yerleştirilmiş yeni renk kontrol kodlarına da uyur. Ötek s. 33).

## Biçimi

PRINT (TAB int-num-exp;)  
[AT int-num-exp;  
int-num-exp;]  
[CHR\$ (int-num-exp;)  
[tabnum;] (int-num-exp;)  
[string-exp;]...]

## RANDOMIZE

**Klavyede yeni**  
T

Cümle/Emir

Klavyede RAND olarak gösterilen RANDOMIZE, rastsal veya belirlenmiş aralıklarla bir dizi sayı elde etmek için RND ile bağlantılı olarak kullanılır.

## RANDOMIZE'in kullanımı

RANDOMIZE ya bir program cümlesi ya da ya da emir olarak kullanılır. İzlenirse bir sayısal değer de izleyebilir. Örneğin,

RANDOMIZE 1

10 RANDOMIZE

RANDOMIZE'ı izleyen değer, gerekirse en yakın tam sayıya yuvarlanır ve 0 ile 65535 arasında bir değer olabilir. Bu değer 0 dan büyüğe buna sistem değişkeni SEED tahsis edilir. SEED'ı izleyen RND daha ayrı sayı sırası verir (sistem değişkeni hakkında bilgi için s. 48'e bakın). Sayı sırasını belirleyen RANDOMIZE'nin değeridir.

RANDOMIZE'ı bir 0'lıya veya bir 1'e değer atlamış ya da SEED'e FRAMES diye adlandırılan bir başka sistem değişkeni tahsis edilir. Buna FRAMES (kareler) denir. Bunun, bunun Spectrum'un geliştirilmesinden bu yana geçen televizyon görüntü karelerini saymasıdır. SEED sayıya 50 ya da 60 kez değiştirildiğinde, RANDOMIZE veya RANDOMIZE'ı izleyen RND'in oluşturacağı sayı sıralanmaya rastlar.

Eğer RANDOMIZE kullanılmazsa, RND, cümlesinin başlanmasından, reset düğmesine ya da NEW tuşuna basılmasından bu yana ürettiği

sayıları sırasıyla aynı yerde sayı üretir.

**Biçimi**  
RANDOMIZE (int-num-exp)

## READ

**Klavyede yeni**  
EXTEND MODE  
A

Cümle/Emir

READ, DATA ile bağlantılı olarak, bir DATA cümlesindeki değerleri kullanan değişkenlere değer vermek üzere kullanılır.

## READ'in kullanımı

READ normal olarak bir program cümlesi olarak kullanılır. Bunu bir veya birkaç sayısal değişken veya dizi değişken izler ve bunlar birbirinden virgüllerle ayrılır. Örneğin,

20 READ a\$, x

READ ilk işleme konduğunda, ilk DATA listesinin başından beri kaç değişken varsa aynı sayıda değer alır. Bu değerler de değişkenlere aynı sırada verir. READ işleme konmasa, bu READ cümlesindeki değişkenlere bir sonraki DATA değerleri verilir ve bu böyle gider.

Ayrıntılı bilgi için DATA ya bak.

## Biçimi

READ num-var (num-var)  
[string-var]  
READ string-var (num-var)  
[string-var]

## REM

REM

**Klavyede yeni**  
E

Cümle

REM, programa giriş ve hatırlatmalar yerleştirmek için kullanılır. Bunlar programın adı, yazan ile bir değişkenin amacı, açıklamak için program satırlarına eklenmiş açıklamalar olabilir. Bu görüş, programın işleyişinde hiç bir etki yapmaz, ancak sıralamada görünürler.

## REM'in kullanımı

REM, program içinde kendi başına bir satır olabilir, gibi bir satırın son cümlesi de olabilir. Bunu istediği gibi tuşlara basılarak yazılmış bir görüş izler.

50 INPUT r\$; REM r\$ = name

Bilgisayar REM'e rastladığında, o satırda onu izleyemeyi dikkate almaz.

## Biçimi

REM istenen karakterler

## RESTORE

**Klavyede yeni**  
EXTEND MODE  
S

Cümle/Emir

RESTORE, READ ve DATA ile birlikte, READ'in programdaki ilk veya bir sonraki DATA cümlesinden değeri belli bir DATA cümlesinden değeri almasını sağlamak için kullanılır.

## RESTORE'in kullanımı

RESTORE normal olarak bir program satırı oluşturur. İzlenirse bunu bir sayısal değer alır. Örneğin,

160 RESTORE 800

RESTORE'ı izleyen değer, gerekiyorsa, en yakın tam sayıya yuvarlanır. Bu değeri, programın DATA cümlesi içeren bir satırın numarası olması gerekir. RESTORE'un ardından, bir sonraki READ cümlesi, bu DATA cümlesindeki değerleri verecektir. Eğer bu numarası takip bir satır yoksa ve bu satırda bir DATA cümlesi bulunmuyorsa, o zaman READ, bundan sonraki ilk DATA cümlesine gider.

Eğer RESTORE'ı bir 0'lıya veya bir 1'e değer atlamış ya da o zaman READ cümlesi programdaki ilk DATA cümlesine gider.

## Biçimi

RESTORE (int-num-exp)

## RETURN

**Klavyede yeni**  
Y

Cümle/Emir

RETURN, bir altprograma son vermek ve bilgisayarı ana programa ya da bir önceki altprograma göndermek için kullanılır.

## RETURN'in kullanımı

RND, bir cümle veya bir emir olarak kendi başına kullanılır. Örneğin,

1000 RETURN

İşleme konduğunda, program, uygulanan son GOSUB cümlesini izleyen cümleye uyar. Ayrıntılar için GOSUB'a bakın.

## Biçimi

RETURN

## RND

RND

**Klavyede yeni**  
EXTEND MODE  
T

İşlev

RND, bir rastsal sayı elde etmek için kullanılır.

## RND'in kullanımı

RND, bir cümle veya bir emir olarak kendi başına kullanılır. Örneğin,

60 LET X=RND

Bunun üzerine RND'ın küçük, 0 dan büyük veya 0'a eşit bir rastsal sayı verir.



Spectrum çağrıldığında, reset değerine veya NEW tuşuna basıldığında, RND'nin vereceği sayılar aynı sıradadır. Şimdi, 75 in kuvvetini alarak (75, 75+75, 75+75+75, vb) ve çıkan sayıyı 65537'ye bölüp sadece kalanı kullanıp, kalandan 1 çıkıp, bu sonucu 65536'ya bölerek elde edilir. Daha rastsal veya daha sabit bir sıra gerekiyorsa, RND den önce RANDOMIZE'yi kullanın.

**Rastsal tam sayılar**  
Spectrum'un INK ve CHG gibi birkok cümlesi ya da işlevi sayılan en yakın tam sayıyı yuvarlarlar. RND bunlarla birlikte doğrudan kullanılabilir. Örneğin, INK RND\*7 rastsal bir mükemmel rengi belirler. Diğer bazı cümle ve işlevler ise tam sayı isteyebilirler. INT (RND\*x)+1, 1'den x'e kadar hep tam sayı verir. 0'dan x'e kadar rastsal bir tam sayı elde etmek için INT (RND\*x+0.5) kullanın.

**Biçimi**  
RND

## RUN

**Klavyede yeri**  
R

**Emir/Cümle**

RUN, programı normal olarak ilk satırdan başlatmak üzere işleme koyar.

**RUN'un kullanımı**  
RUN, doğrudan bir emir olabileceği gibi bir program satır da oluşturabilir. İsterse bunu sayısal bir değer de izleyebilir. Örneğin,

RUN 50

Eğer RUN'u bir değer izlemiyorsa, program ilk satırdan işleme konur. Eğer bir değer eklenmişse, bu gerekiyorsa en yakın tam sayıya yuvarlanır ve program bu satırdan başlar. Eğer böyle bir satır yoksa, program bundan sonra gelen satırdan başlar. Dikkat edin, RUN, programı işleme koymadan önce CLEAR'de uygular, bu yüzden mevcut değişken değerleri silinir. Bunu önlemek için, ardından bir satır numarası gelen GOTO kullanın.

Eğer bir program LINE kullanılarak saklanıyorsa, yüklenir yüklenmez otomatik olarak işleme konur, RUN'a gerek olmaz.

**Biçimi**  
RUN [int-num-exp]

## SAVE

**Klavyede yeri**  
S

**Emir/Cümle**

SAVE, bir programı kasette saklamak için kaseti teypte gönderir.

**SAVE'nin kullanımı**  
SAVE normal olarak doğrudan bir emir olarak kullanılır ama bir program cümlesi de oluşturabilir. Buna, bir dizi değer olan dosya adı izler. Örneğin,

SAVE "dosyaad"

Dosya adı en fazla on karakter olabilir. Uygulamaya konduğunda ekranda

Start tape, then press any key

"Teypi çalıştırın, sonra herhangi bir tuşa basın" mesajı görünür. Herhangi bir tuşa basılınca program kaseti teypte gönderilir ve sonuçta **BOOK, 1** mesajı ekranda belirir. Dikkat edin, Microdrive bağlandığında SAVE farklı kullanılır. Bunun ayrıntısı için Microdrive ve Interface 1 el kitabına bakın.

**Otomatik işleme koyma**  
Eğer depolanmış program, yüklenince otomatik olarak işleme konacaksa, SAVE LINE ile bağlantılı kullanılmalıdır. Program adını LINE ve bir sayısal değer izler. Örneğin,

SAVE "dosyaad" LINE 1

LINE'i izleyen değer, gerekiyorsa en yakın tam sayıya yuvarlanır. Bu durumda 1 veya programdaki bir satırın numarası olmalıdır. Program daha sonra teypte aynı SAVE'de yapıldığı gibi gönderilir. Programı yüklenince, bu vermiş olan satırdan veya böyle bir satır yoksa programda bunu izleyen satırdan başlayarak otomatik olarak işleme konur. Pratikte LINE 1'in kullanılması bütün programın otomatik olarak başlamasına neden olur.

**Biçimi**  
SAVE string-exp [LINE int-num-exp]

## SAVE CODE

**Klavyede yeri**  
S

**EXTEND MODE**  
I

**Emir/Cümle**

SAVE CODE, belirlenmiş bulunan bir bölümün bir kasette saklanması üzere kaseti teypte gönderir. Daha sonra bu bilgi LOAD CODE kullanılarak tekrar belleğe yerleştirilebilir.

**SAVE CODE'nin kullanımı**  
SAVE CODE doğrudan bir emir veya programın bir cümlesi olarak kullanılabilir. SAVE'i, bir dizi değer olan dosya adı ile CODE izler. Örneğin ardından virgüle ayrılmış iki sayısal değer gelir. Örneğin,

SAVE "picture" CODE 16384,6912

SAVE'i izleyen dosya adı en fazla on karakterden oluşur. CODE'u izleyen iki değerdendir bir bir gerekiyorsa en yakın tam sayıya yuvarlanır. İki (yukarıda 16384) bellekteki bölgenin başlangıç adresini, ikinci (6912)

ise, depolanacak byte ların sayısını verir. Bunun üzerine bilgi, tipik SAVE ile programa yapıldığı gibi, kaseti teypte gönderir. Yukarıdaki emirle saklanan bilgi ekrandaki görüntüdür.

**Biçimi**  
SAVE string-exp CODE int-num-exp [int-num-exp]

## SAVE DATA

**Klavyede yeri**  
S

**EXTEND MODE**  
D

**Cümle/Emir**

SAVE DATA, bir düzeni teypte depolar. Bundan sonra da DÜZEN LOAD DATA ile yüklenir.

**SAVE DATA'nın kullanımı**  
SAVE DATA, bir program cümlesi oluşturabileceği gibi, doğrudan emir de olabilir. SAVE'i bir dosya adı, DATA, bir harf veya bir harf \$ ve nihayet bir boş parantez izler. Örneğin,

450 SAVE "numbers" DATA n()

750 SAVE "names" DATA n\$()

Düzenin dosya adı en fazla on karakterden oluşur. DATA'yı izleyen harf veya harf\$ teypte saklanacak olan programdaki düzenin addır. Bunun üzerine düzen, tipik SAVE ile programa yapıldığı gibi, kaseti teypte gönderir.

**Biçimi**  
SAVE string-exp DATA letter [S] O

## SAVE SCREENS

**Klavyede yeri**  
S

**EXTEND MODE**  
SYMBOL SHIFT K

**Emir/Cümle**

SAVE SCREENS ekrandaki görüntüyü teypte depolar. İleride bu LOAD SCREENS kullanılarak yeniden bilgisayara yüklenebilir.

**SAVE SCREENS'in kullanımı**  
SAVE SCREENS doğrudan bir emir veya bir program cümlesi biçiminde kullanılabilir. SAVE'i bir dizi değer olan bir dosya adı ile SCREENS izler. Örneğin,

SAVE "picture" SCREENS

Dosya adı en fazla on karakterden oluşur. Bunun üzerine görüntü, tipik SAVE ile programa yapıldığı gibi, kaseti teypte gönderir.

**Biçimi**  
SAVE string-exp SCREENS

## SCREENS

**Klavyede yeri**  
EXTEND MODE  
SYMBOL SHIFT K

İşlev

SCREENS, ekranda belli bir pozisyonda hangi karakterin bulunduğunu sağlar.

**SCREENS'in kullanımı**  
SCREENS'i bir virgüle ayrılmış ve hepsi paranteze alınmış iki sayısal değer izler. Örneğin,

100 IF SCREENS (0,0) = "X" THEN PRINT "CRASH"

SCREENS'i izleyen değerler, gerekiyorsa, en yakın tam sayıya yuvarlanır. İlk değer (yukarıda 0) 0 ile 255 arasında olabilir ve ekrandaki pozisyonun satır sayısını verir. İkinci değer (yukarıda 0) 1 ile 31 arasında olabilir ve bu pozisyonun sütun sayısını verir. Bunun üzerine SCREENS bu pozisyonundaki karakteri bir dizi de gönderir (yukarıdaki örnekte olduğu gibi) ancak işaretlenmiş bir karakter olarak verir. Eğer bu pozisyonda bir karakter yoksa, SCREENS bir boş dizi ("") verir.

Dikkat edin, SCREENS ayrıca SAVE ve LOAD ile birlikte ekrandaki görüntüyü depolamak ya da yüklemek için de kullanılabilir. Ayrıntısı için SAVE SCREENS ve LOAD SCREENS'e bakın.

**Biçimi**  
SCREENS (int-num-exp, int-num-exp)

## SGN

**Klavyede yeri**  
EXTEND MODE  
F

**İşlev**

SGN, bir rakamın artı, eksi veya sıfır olduğunu işaret eder.

**SGN'nin kullanımı**  
SGN'yi bir sayısal değer izler. Örneğin,

50 LET x=SGN y

İfade paranteze alınmalıdır. SGN, önemsenmiş (yukarıda y) değeri artı ne 1, eksi ne -1, sıfır ise 0 görüntüsü verir.

**Biçimi**  
SGN num-const  
SGN num-var  
SGN (num-exp)

## SIN

**Klavyede yeri**  
EXTEND MODE  
Q

**İşlev**

SIN'in işi  
SIN bir açının sinüsünü verir.

**SIN'in kullanımı**  
SIN'i sayısal bir değer izler. Örneğin,

80 LET x=SIN y

İfade paranteze alınmalıdır. SIN'i izleyen değer radian olarak bir açının değeridir. SIN açının sinüsünü

verir. Dereceye PI/180 ile çarpılarak radianla çevrilir.

Dikkat edin SIN, 0 ile 180 arasındaki açıların değerini artı, 180 ile 360 arasındaki açıların değerini ise eksi olarak verir.

**Örnek**  
Şu emir

PRINT SIN (30 \* PI/180)

0.5 görüntüsünü verir. Bu 30 derecelik bir açının sinüsüdür.

**Biçimi**  
SIN num-const  
SIN num-var  
SIN (num-exp)

## SQR

**Klavyede yeri**  
EXTEND MODE  
H

**İşlev**

SQR bir rakamın kare kökünü verir.

**SQR'in kullanımı**  
SQR'u, bir sayısal değer izler. Örneğin,

70 LET x=SQR y

İfade paranteze alınmalıdır. SQR'ı izleyen değer (yukarıda y) sıfırdan büyük olmalıdır. SQR, bunun kare kökünü verir.

**Biçimi**  
SQR num-const  
SQR num-var  
(num-exp)

## STOP

**Klavyede yeri**  
SYMBOL SHIFT D

**Bak FOR**

## STOP

**Klavyede yeri**  
SYMBOL SHIFT A

**Cümle/Emir**

STOP, programı belli bir noktada durdurur. Altprogramlar ayrı bir bölümde toplanmışsa, programın ana bölümünün sonunda STOP'u kullanmak gerekli olabilir. STOP ayrıca bir programın değişkenlerinin kısırlarını gidermek için de yararlı olabilir.

**STOP'un kullanımı**  
STOP normal olarak bir program cümlesi olarak kullanılır. Kendi başına kullanılır. Örneğin,

650 STOP

Uygulamaya konunca program durur ve ekrana

9 STOP statement

mesajı ile programın durdurulduğu satır ve cümle numarası yansır. Bundan sonra, örneğin değişkenlerin değerlerinin yenilenmesi veya değiştirilmesi gibi

değişiklikler yapılabilir. CONTINUE tuşuna basılması ise programın bir sonraki satırdan yeni değerlerle devam etmesini sağlar.

**Biçimi**  
STOP

## STR\$

**Klavyede yeri**  
EXTEND MODE  
Y

**İşlev**

STR\$ bir rakamı dikeye dönüştürür.

**STR\$'nin kullanımı**  
STR\$'yi bir sayısal değer izler. Örneğin,

90 LET a\$=STR\$ x

İfade paranteze alınmalıdır. Bunun üzerine STR\$ önemsenmiş (yukarıda x) değeri bir dizi değeri olarak verir. Eğer x'e 65 değeri verilirse, o zaman yukarıdaki cümle a\$'ye "65" değeri verir.

**Biçimi**  
STR\$ num-const  
STR\$ num-var  
STR\$ (num-exp)

## TAB

**Klavyede yeri**  
EXTEND MODE  
E

**Bak LPRINT, PRINT**

## TAN

**Klavyede yeri**  
EXTEND MODE  
E

**İşlev**

TAN, bir açının tanjantını verir.

**TAN'in kullanımı**  
TAN'i bir sayısal değer izler. Örneğin,

130 LET x=TAN y

İfade paranteze alınmalıdır. TAN'i izleyen değer, radian olarak bir açının değeridir. TAN açının değeri verir. Derece radianla PI/180 ile çarpılarak çevrilir.

Dikkat edin, TAN, 0 ile 90 dereceleri 180 ile 270 dereceleri arasındaki açıların değerini artı, 90 ile 180 ve 270 ile 360 derece arasındaki açıların değerini ise eksi olarak verir.

**Biçimi**  
TAN num-const  
TAN num-var  
TAN (num-exp)

## THEN

**Klavyede yeri**  
SYMBOL SHIFT G

**Bak IF**

**Klavyede yeni**  
SYMBOL SHIFT F

İşlev

TO'nun Sincir BASIC'te işlevi değişik kullanımı vardır. FOR'a bağlanış olarak FOR NEXT döngüsü oluşturmak için (ayrım) için bak FOR'da ve dizi bölücü (dizinin daha küçük alt dizilerine bölmek için) olarak kullanılır.

**TO'nun dizi bölücü olarak kullanımı**

TO bir ana dizindeki bir alt dizinin ilk ve son karakterlerinin belirlenmek için kullanılır. TO'dan önce bir dizi değeri gelir. Parantez açılır, istenirse bir sayısal değer verilir. TO'dan sonra da istenirse bir başka sayısal değer verilir parantez kapanır. Örneğin:

80 PRINT a\$(4 TO 7)

Dizi ifade paranteze alınmalıdır. Dizi değeri (yukarıda a\$) bölünecek olan dizedir. İki sayısal değer (4 ve 7) bu dizi içindeki alt dizinin ilk ve son karakterleridir. Bunun üzerine TO alt diziyi (a\$'nin 4'ten 7'ye kadar olan karakterlerini) verir. Her bir şey verilmese de sayısal değer 1, son sayısal değer de dizideki son karakterin pozisyonunun değeridir. Eğer alt dizi dizinin ilk karakteri ile başlayacaksa ilk değer verilmeyebilir. Eğer alt dizi dizinin son karakteri ile sona eriyorsa, son değer verilmese de olur.

**Biçimi**  
string:const [(num-exp) TO  
(num-exp)]  
string-var [(num-exp) TO  
(num-exp)]  
(string-exp) [(num-exp) TO  
(num-exp)]

**USR** User SubRoutine

**Klavyede yeni**  
EXTEND MODE

İşlev

USR, bellekte belli bir adrese yerleştirilmiş olan bir makina kodu altprogramını işletmek için kullanılır. USR ayrıca, kullanıcının tanımladığı grafikler için verilen bellegin üst kısmında aynımsi yere yerleştirilmesi için de kullanılır.

**USR ve makina kodu**

Makina kodunu kullanabilmek için USR'yi bir sayısal değer ister.

Örneğin:

80 PRINT USR 65000

100 RANDOMIZE USR 65000

İfade paranteze alınmalıdır. USR'yi isteyen değer bir tam sayıya yuvarlanır ve makina kodunun bellekteki başlangıç adresini oluşturur. USR'çeren herhangi bir cümle bu adresteki altprogram ister ve USR "bc" kutuk çıkışın işlevinin değerini verir. Örneğin,

RANDOMIZE USR veya RESTORE USR sadece altprogram işleme koyarken, PRINT USR buna ek olarak "bc" kutuk değerini de ekrana yansıtır.

**USR ve kullanıcının tanımladığı grafikler**

Kullanıcının tanımladığı grafikler için USR, POKe ile birlikte kullanılır. Bunun bir dizi değeri ya da değişken izler ve bunlar POKe cümlesi için bir adres verir. Örneğin:

50 POKe USR "a", 255

USR'yi isteyen dizi değeri, A ile U veya a ile u arasında bir tek harftir.

Büyük harflerle küçük harfler arasında bir ayırım gözükmür.

Bunun üzerine USR, bellegin kullanıcının tanımladığı grafiklere ayrılmış 21 bölümden birinin başlangıç adresini verir. Her bölüm, bir grafik karakter yaratacak şekilde sekiz byte'in POKe edilebilirliği sekiz adres içerir. Byte'ler ondalık veya iki biçimde (bak BIN) verilebilir.

**Biçimi**

USR int-num-const

USR int-num-var

USR (int-num-exp)

USR string-const

USR string-var

**VAL** Value

**Klavyede yeni**  
EXTEND MODE

İşlev

VAL, sayısal değere sahip bir diziyi sayıya çevirir.

**VAL'in kullanımı**

VAL i bir dizi değeri veya değişken izler. Örneğin,

70 LET x=VAL a\$

Dizi değeri veya değişkenin değeri tmak işaretlerinden çıkarılır ve bu durumda sayısal bir değer olması gerekir. VAL bunu değerlendirir ve bu bir sayısal değeri olarak verir.

**Örnekler**

a\$'nin değeri "435" ise, o zaman yukarıdaki cümle x'e 435 değeri verir. Ancak VAL, ifadeleri de değerlendirir.

10 INPUT a\$,x

20 PRINT VAL a\$

a\$'yi verilen değer x'i kullanan bir ifade olmadıkça, "x=x" gibi. Bunun üzerine x'e örneğin 5 gibi bir sayısal değer verilir. VAL, bu ifadeden tmak işaretlerini çıkıp x=x elde eder ve x'e verilen değeri değerlendirilerek ekrana sonuç olarak 25'i yansıtır.

**Biçimi**

VAL string:const

VAL string-var

**VAL\$** Value (string)

**Klavyede yeni**  
EXTEND MODE

SYMBOL SHIFT J

İşlev

VAL\$ bir diziyi bir dizi ifade olarak değerlendirir.

**VAL\$'in kullanımı**

VAL\$ i bir dizi değeri izler.

Örneğin,

110 PRINT VAL\$a\$

Dizi değeri değeri, tmak işaretlerinden ayrılır. Bu durumda bunun bir dizi ifade olması gerekir.

VAL\$ ifadesi değerlendirir ve değeri bir dizi değeri olarak verir.

**Örnekler**

Bu programı deneyin:

10 INPUT a\$,x\$

20 PRINT VAL\$a\$

a\$'e verilen dizi değeri x\$'yi kullanan bir ifade, örneğin "x\$+x\$" olmalıdır. Bu dizi değeri, bunun üzerine, x\$ "DO" ya verilir örneğin, VAL\$a\$, x\$'nin değerinden tmak işaretlerini ayıklayıp x\$+x\$'i alır ve x\$'ye verilen değeri değerlendirir, örneğin sonuç olan DODO'ya ekrana yansıtır.

**Biçimi**

VAL\$ string-var

**VERIFY**

**Klavyede yeni**  
EXTEND MODE

SYMBOL SHIFT R

Emir/Cümle

VERIFY, SAVE'den sonra bir programın doğru bir biçimde teypte aktarılıp aktarılmadığını denetler.

**VERIFY'in kullanımı**

VERIFY, Normal olarak, tpki LOAD gibi doğrudan bir emir olarak verilir ve bunu program adı izler. Örneğin,

VERIFY "dosyaad"

Teyp çalışırken, bulunan her programın adı ekrana yansır ve bellekteki programla aynı adı taşıyan teypteki program karşılaştırılır. Eğer ikisi de aynı ise

**OK,0:1**

mesaj verilir.

VERIFY bir Microdrive bağlı ise farklı kullanılır. Aynımsi için Microdrive ve Interface 1 el kitabına bakın.

**VERIFY CODE ve VERIFY DATA**

VERIFY CODE, tpki LOAD CODE gibi, bellekteki bir bilgini bölünmüş teypte aktarılmasını doğrulamak için kullanılır. VERIFY DATA da LOAD DATA gibi bir dizinin teypte aktarılmasını denetlemek için kullanılır. Ayrıntılar için LOAD CODE ve LOAD DATA'ya bakın.

**Biçimi**

VERIFY string-exp

VERIFY string-exp CODE

[(int-num-exp) [(int-num-exp)

VERIFY string-exp DATA

letter (0:0)

**VERIFY CODE**

**Klavyede yeni**

EXTEND MODE

SYMBOL SHIFT R

EXTEND MODE

İ

Bak VERIFY

**VERIFY DATA**

EXTEND MODE

SYMBOL SHIFT R

EXTEND MODE

D

Bak VERIFY









## ZX SPECTRUM + EKRAN MESAJLARI

Spectrum, BASIC'i uygularken durursa, ekranın alt kısmında bir mesaj belirir. Bu, ya bir emrin ya da programın tamamlandığını ya da bir hata olduğunu gösterir. Her mesaj, bir kod numarası veya harf ile kısa bir mesaj ve bilgisayarın durduğu satır ya da cümle numarasından oluşur. Emir, 0. satır ve bir satır başında ise 1. cümle, ilk nokta virgülden sonra ise 2. cümle vb olarak gösterilir. CONTINUE, normal olarak programın ekranda gösterilen cümleden yeniden başlamasını sağlar.

### 0 OK

İşlemin başını ile tamamlandığını veya programda bulunan satır numaralarının daha büyük bir satıra atlama girişimini gösterir. CONTINUE, bu mesajı dikkate almaz ve bir önceki mesajda belirtilen cümleden yeniden başlar.

### 1 NEXT without FOR

Kendisini izleyen bir FOR olmayan bir NEXT ile karşılaşılmıştır ve kontrol değişimi ile aynı adı taşıyan bir başka değişim daha vardır.

### 2 Variable not found

Bir değer verilmemiş veya teypten değer yüklenmeden bir basit değişim kullanılmış; veya önce FOR cümlesi ile tanımlanmadan NEXT ile bir kontrol değişimi kullanılmış; veya DIM ile düzenin boyutu tanımlanmadan ya da bir düzen teypten yüklenmeden önce bir altöge değişimi kullanılmış demektir.

### 3 Subscript wrong

Altöge, düzenin boyutları dışında.

### 4 Out of memory

Cümleyi ya da emri tamamlamak için bellekte yeterli yer yok.

### 5 Out of screen

INPUT, ekranın alt bölümü için 23 satırdan fazla satır vermiş ya da PRINT AT ile birlikte 22 ya da daha yüksek bir satır numarası kullanılmış demektir.

### 6 Number too big

Bilgisayar 10<sup>9</sup>'den daha büyük bir rakam elde etmeye çalışmıştır.

### 7 RETURN without GOSUB

RETURN cümlelerinin numarası GOSUB cümlesinin numarasından bir büyüktür.

### 8 End of file

Microdrive dosya işleme emri.

### 9 STOP statement

Programı durdurmak için STOP kullanılmıştır. CONTINUE bir sonraki cümleden başlatacaktır.

### A Invalid argument

İşleve yanlış değer verilmiştir.

### B Integer out of range

Bir değer en yakın tam sayıya yuvarlanmış, ancak boyutları dışında olduğu için kabul edilememiştir.

### C Nonsense in BASIC

Cümle, program çeviricisinde BASIC açısından bir anlam ifade etmiyor.

### D BREAK - CONT repeats

BREAK'e basılmıştır. CONTINUE programı durdurulduğu cümleyi tekrarlayacaktır.

### E Out of DATA

READ, programdaki son DATA cümlesinin ötesinde birşeyler okumaya kalkışmıştır.

### F Invalid file name

SAVE, ondan fazla karakterden oluşan bir adla birlikte kullanılmıştır.

### G No room for line

Yeni program satırının gireceği yeterli bellek yeri yoktur.

### H STOP in INPUT

INPUT'a karşılık olarak veya veri verilirken STOP'a basılmıştır. CONTINUE INPUT cümlesini tekrarlar.

### I FOR without NEXT

Limitli veya STEP değeri yanlış olduğu için (örneğin FORx=5 TO 0 STEP 1 kullanıldığında) ve bunu karşılayacak NEXT bulunmadığından bir FOR NEXT döngüsü uygulanamamaktadır.

### J Invalid I/O device

Microdrive dosya işleme emri.

### K Invalid colour

INK, PAPER, FLASH, BRIGHT, INVERSE veya OVER ya da bunlara ilişkin kontrol karakteri için belirtilen değer, kullanım alanı dışındadır.

### L BREAK into program

BREAK 'e basılmıştır. Bu mesaj uygulanacak son cümleyi gösterir. CONTINUE bir sonraki cümleden başlatacaktır.

### M RAMTOP no good

RAMTOP için belirlenen değer, ya çok büyük ya da çok küçüktür.

### N Statement lost

Artık mevcut olmayan bir cümleyle atlama girişimi olmuştur.

### O Invalid stream

Microdrive dosya işleme emri.

### P FN without DEF

Bunu karşılayan bir DEF FN cümlesi olmadan bir FN cümlesi kullanılmıştır.

### Q Parameter error

FN cümlesi, fonksiyona iletmek üzere yanlış sayıda değerler içermektedir veya değerlerden biri yanlış türdedir (örneğin sayı yerine diz ya da ters).

### R Tape loading error

Yükleme, birleştirme veya doğrulama işlemleri başarısızlığa uğramıştır.

## BASIC'İNÖTESİ

BASIC, birçok uygulamada çok iyi iş gören, çok amaçlı bir bilgisayar dildir. Ancak bu, Spectrum'da kullanılabilecek tek bilgisayar dili değildir. FORTH, micro-PROLOG ve LOGO gibi diğer dillerdeki yazılar da bulunmaktadır. Bu diğer BASIC'ten çok farklı işler ve bilgisayarınıza yeni olanaklar açar.

BASIC çok amaçlı bir dil olduğundan, bazı uygulamalarda hantal kalabilir. Göreceli olarak yavaşır. Diğer diller, daha basit bir programlama ve daha hızlı işleme ile birleşik daha büyük esneklik sağlayabilirler. Örneğin, FORTH, size kendi kelimelerinizi tanımlama ve bunları bilgisayarınızın anlayacağı, BASIC'deki benzer emirlerden on kat hızla uygulayacağı talimatlarınızda kullanma olanakları sağlar. Micro-PROLOG ile bilgisayar basit İngilizce cümleleri anlar, bunları kullanan ile basit bir karşılıklı konuşma için belleğinde tutar. LOGO eğitimi için kullanılmak üzere geliştirilmiş bir bilgisayar dildir. Çok esnek bir biçimde kullanılabilecek çok basit emirler içerir. Ancak ZX Spectrum+ 'unuzu gerçekten hızlı programlar yazmak istiyorsanız, makina kodu ile nasıl programlama yapacağınızı anlamamız gerekir.

### Makina kodu

BASIC sizin kolaylıkla anlayabileceğiniz biçimde bilgisayara talimat vermenizi sağlar. Spectrum'un Merkezi İşlem Birimi (CPU), yani göçlü Z80A yongası, aslında BASIC'i anlamaz. Belleğin bir kesimi, BASIC çevirmeni denen, sizin BASIC ile verdiğiniz talimatları kodlu sinyallere dönüştüren bir kâğıt program içerir. Aslında Z80A'nın sizin talimatlarınızı uygulamasını sağlayan bu kodlardır.

Bu çevirmenin sizin BASIC ile verdiğiniz talimatları Z80A koduna ya da yaygın adıyla makina koduna çevirmesi biraz zaman alır. Ancak isterseniz BASIC çevirmenini atlayıp makina kodlarınızı doğrudan Z80A'ya gönderebilirsiniz. Bunun üzerine programınız hızla işleme konur. Bunun için ödenmesi gereken bedel, makina kodu ile program yazmak için harcanacak fazladan zamandır. BASIC ten farklı olarak bu hiç de "dostça olmayan" bir dildir. Yani bunu öğrenmek epey bir zaman alır. Makina kodu ile programlama bu kitabın boyutları dışındadır. Ancak Spectrum'un makina kodunu çok iyi düzeyde öğrenen kitaplar vardır. Makina kodunun hızının tadına varabilmemiz için şu kısa göster programını yazıp verin.

### ANINDA ÇİZGİLER

```
10 FOR X=0 TO 15
20 REPEAT N: FOR Y=0 TO 15
30 HP Y:
40 DATA 22,255,63,1,1,26,22
50 DATA 22,255,63,1,1,26,22
60 DATA 22,255,63,1,1,26,22
70 DATA 22,255,63,1,1,26,22
80 DATA 22,255,63,1,1,26,22
90 DATA 22,255,63,1,1,26,22
100 DATA 22,255,63,1,1,26,22
110 DATA 22,255,63,1,1,26,22
120 DATA 22,255,63,1,1,26,22
130 DATA 22,255,63,1,1,26,22
140 DATA 22,255,63,1,1,26,22
150 DATA 22,255,63,1,1,26,22
160 DATA 22,255,63,1,1,26,22
170 DATA 22,255,63,1,1,26,22
180 DATA 22,255,63,1,1,26,22
190 DATA 22,255,63,1,1,26,22
200 DATA 22,255,63,1,1,26,22
210 DATA 22,255,63,1,1,26,22
220 DATA 22,255,63,1,1,26,22
230 DATA 22,255,63,1,1,26,22
240 DATA 22,255,63,1,1,26,22
250 DATA 22,255,63,1,1,26,22
260 DATA 22,255,63,1,1,26,22
270 DATA 22,255,63,1,1,26,22
280 DATA 22,255,63,1,1,26,22
290 DATA 22,255,63,1,1,26,22
300 DATA 22,255,63,1,1,26,22
310 DATA 22,255,63,1,1,26,22
320 DATA 22,255,63,1,1,26,22
330 DATA 22,255,63,1,1,26,22
340 DATA 22,255,63,1,1,26,22
350 DATA 22,255,63,1,1,26,22
360 DATA 22,255,63,1,1,26,22
370 DATA 22,255,63,1,1,26,22
380 DATA 22,255,63,1,1,26,22
390 DATA 22,255,63,1,1,26,22
400 DATA 22,255,63,1,1,26,22
410 DATA 22,255,63,1,1,26,22
420 DATA 22,255,63,1,1,26,22
430 DATA 22,255,63,1,1,26,22
440 DATA 22,255,63,1,1,26,22
450 DATA 22,255,63,1,1,26,22
460 DATA 22,255,63,1,1,26,22
470 DATA 22,255,63,1,1,26,22
480 DATA 22,255,63,1,1,26,22
490 DATA 22,255,63,1,1,26,22
500 DATA 22,255,63,1,1,26,22
510 DATA 22,255,63,1,1,26,22
520 DATA 22,255,63,1,1,26,22
530 DATA 22,255,63,1,1,26,22
540 DATA 22,255,63,1,1,26,22
550 DATA 22,255,63,1,1,26,22
560 DATA 22,255,63,1,1,26,22
570 DATA 22,255,63,1,1,26,22
580 DATA 22,255,63,1,1,26,22
590 DATA 22,255,63,1,1,26,22
600 DATA 22,255,63,1,1,26,22
610 DATA 22,255,63,1,1,26,22
620 DATA 22,255,63,1,1,26,22
630 DATA 22,255,63,1,1,26,22
640 DATA 22,255,63,1,1,26,22
650 DATA 22,255,63,1,1,26,22
660 DATA 22,255,63,1,1,26,22
670 DATA 22,255,63,1,1,26,22
680 DATA 22,255,63,1,1,26,22
690 DATA 22,255,63,1,1,26,22
700 DATA 22,255,63,1,1,26,22
710 DATA 22,255,63,1,1,26,22
720 DATA 22,255,63,1,1,26,22
730 DATA 22,255,63,1,1,26,22
740 DATA 22,255,63,1,1,26,22
750 DATA 22,255,63,1,1,26,22
760 DATA 22,255,63,1,1,26,22
770 DATA 22,255,63,1,1,26,22
780 DATA 22,255,63,1,1,26,22
790 DATA 22,255,63,1,1,26,22
800 DATA 22,255,63,1,1,26,22
810 DATA 22,255,63,1,1,26,22
820 DATA 22,255,63,1,1,26,22
830 DATA 22,255,63,1,1,26,22
840 DATA 22,255,63,1,1,26,22
850 DATA 22,255,63,1,1,26,22
860 DATA 22,255,63,1,1,26,22
870 DATA 22,255,63,1,1,26,22
880 DATA 22,255,63,1,1,26,22
890 DATA 22,255,63,1,1,26,22
900 DATA 22,255,63,1,1,26,22
910 DATA 22,255,63,1,1,26,22
920 DATA 22,255,63,1,1,26,22
930 DATA 22,255,63,1,1,26,22
940 DATA 22,255,63,1,1,26,22
950 DATA 22,255,63,1,1,26,22
960 DATA 22,255,63,1,1,26,22
970 DATA 22,255,63,1,1,26,22
980 DATA 22,255,63,1,1,26,22
990 DATA 22,255,63,1,1,26,22
1000 DATA 22,255,63,1,1,26,22
```



50. satırdaki 55'in değerini 1 ile 255 arasındaki başka herhangi bir rakama çevirin ve çubukların değişimini izleyin. Ayrıca başta bir INK emri kullanarak renkli çubuklar da elde edebilirsiniz. Makina kodu kullanınca ne kadar hızla görüntü elde ettiğinize bakın. Bu, adeta anında olmaktadır. BASIC in ise ekranı doldurması iki saniye sürer.

Bu program, 10 ile 30. satırlarla, bellekte 65000 adresinden sonrasına DATA cümlesi ile 16 kod yerleştirildiğinden işlemektedir. 70. satır kodları Z80A'ya göndermekte ve derhal görüntü oluşturmaktadır. Dikkat edin sekizinci kod, çubukların genişliğini kontrol etmektedir.

Spectrum'unuz için mevcut oyunların çoğu, mümkün olan en hızlı sonucu elde etmek için makina kodu ile yazılmıştır. Makina kodu yazabilmemiz için "düzenleyici" (assembler) denen bazı programlar mevcuttur. Makina kodu sadece bazı rakamları isterken, bunları size tuflara basarak vereceğiniz talimatları sağlar. Bu talimatlar, BASIC komutları gibi İngilizce kelimeler değil, bilgisayarın mutlakla yerine getirmesi gereken işlemleri ifade eden bazı kısaltmalar ya da özel deyimlerdir. Bu yüzden, "düzenleme" dilini kullanmadan önce bilgisayarın nasıl çalıştığını adım adım öğrenmeniz gerekir.

## BİLGİSAYAR JARGONU — NE ANLAMA GELİYOR

Bilgisayarda kullanılan kelimelerin çoğu günlük hayatta kullanılan kelimelerdir ama çoğunun anlamı farklıdır. Burada, bu kitapta kullanılan kelimelerden bazıları ile bilgisayarla ilgili bazı deyimlerin açıklanması bulacaksınız. İtaliye yazılmış kelimeler bu sayfalarda ayrıca açıklanmak. Eğer bu kitapta anlamadığınız bir kelime ya da deyim varsa ve burada bulamazsanız, o zaman dizine bakın.

**Address (Adres)** Belleğin bir tek birimi. ZX Spectrum + 'da 65536 adres vardır.

**Argument (Önerme)** Bir sonuç almak için bir işlemin kullandığı değer.

**Array (Düzen)** Belleğin bir kesiminde bir arada tutulan, birbiriyle bağlantılı, bir grup veri.

**Attributes (Gönderme)** Karakterlerin renklerini veren kodlar.

**BASIC** ZX Spectrum + ve diğer ve bilgisayarlarının kullandığı bilgisayar dili.

**Binary code** (İkili kod) Bilgisayarların kullandığı bir tür kod. Akımin açık ya da kapalı olması durumundan oluşan bazı obeklerdir.

**Bit** (İkili kodda açık ya da kapalı olma durumu). İkili rakamın kısaltılması.

**Byte** Değeri 0 ile 255 arasında bir rakamı ifade eden sekiz bit'ten oluşan bir dizi. Bellekteki her adreste bir byte vardır.

**Character** (Karakter) Ekrana yansıtılan veya yazdırılan, tek bir harf, (0 ile 9 arasında) bir rakam, bir işaret veya grafik birimi.

**Character set** (Karakter tablosu) Bilgisayarın kullandığı önceden belirlenmiş karakterlerin ve bazı kontrol kodlarının tam listesi.

**Command** (Emir) Bilgisayarın uyguladığı bir tek talimat ve doğrudan emir.

**Concatenation** (Küme) Birbirine ekleyerek dizilen birleştirme.

**Constant** (Değişmez) Bir sayı veya bir ya da birkaç harf ya da başka karakterden oluşan grup.

**CPU (Central Processing Unit)** (Merkezi İşlem Birimi) Bilgisayarın işlemleri yapan ve

diğer birimleri denetleyen ana parçası. ZX Spectrum + bir Z80 mikro bilgi işlem aygıtı kullanır.

**Cursor** (Göstergeç) Ekranda bir sonraki görüntünün belireceği pozisyon. Bunu bilgisayarın hangi modda olduğunu gösteren yanıp sönen bir işaret belirir.

**Data** (Veri) Bir sonuç elde etmek için bilgisayara beslenen veya bilgisayarın bir programdan elde ettiği bilgi.

**Direct command** (Doğrudan emir) Bilgisayara verilir verilmez derhal uygulanan bir ya da birkaç talimat.

**Edit** (Düzeltilme) Bir programdaki ayrıntıları değiştirme.

**Enter** (Verme) Tamamlanmış bir talimat veya bilgiyi bilgisayara verme.

**Expression** (İfade) Değişmez, değişken komutlardan oluşan bir birleşim

**False** (Yanlış) Bilgisayarın gerçek dışı ya da yanlış olduğuna karar verdiği bir durum veya sonuç. False'ın sayısal değeri 0'dır.

**Function** (Fonksiyon/İşlev) Bilgisayarın bir veya birkaç değer (veya önerme) alıp, bunları kullanarak bir başka değer olan sonucu verdiği işlem.

**Graphics** (Grafik işaretler) Bilgisayarın resim, çizim veya şema gibi görüntüler oluşturmaları.

**Hardware** (Donanım) Bilgisayarın kendisi veya bununla ilişkili yan donanım gibi araç ve makineler.

**Information** (Bilgi) Bilgisayarın işlemesi istenen kelimeler, rakamlar ve işaretlerden oluşan herhangi bir bileşim.

**Input** (Girdi) Bilgisayara verilen program ve veriler.

**Interface** (Araparça) Bilgisayar ve/veya yan donanım birbirine bağlayan ve bunların birbiriyle haberleşmesini sağlayan bir birim.

**K** Bir bilgisayarın bellek kapasitesinin ölçüsü. 1K bir kilobyte'a, yani 1024 byte'a eşittir. K olarak bellek kapasitesi, her bir byte depolayabilecek bellekteki adreslerin toplamına eşittir. ZX Spectrum + 'in 48K RAM ve 16K ROM, toplam 64K kapasitesi vardır.

**Keyword** (Komut) Bilgisayara BASIC'le verilen talimat. İşleme girebilmesi için bazı değerlere gerek olabilir.

**Line** (Satır) Bir programdaki bir talimat veya bir dizi talimat. Bunun diğer satırlarla birlikte

doğru bir noktada işleme girebilmesi için bir numara verilir.

**Listing** (Sıralama) Sırayla dizilmiş program satırları.

**Load** (Yükleme) Bir kartuş veya kaset gibi bir depolama aygıtından bilgisayara program veya veri verme.

**Logic** (Mantık) Bilgisayarın sonuçların doğru veya yanlış, bir durumun geçerli veya geçersiz olup olmadığının karar verdiği süreç.

**Machine code** (Makina kodu) ZX Spectrum + 'un anladığı dil. Bilgisayar BASIC'le verilmiş programları işlerken bunları makina koduna çevirir.

**Memory** (Bellek) Bilgisayarın gerektiğinde program ve verileri, ayrıca kalıcı işletme talimatlarını saklayan kesimi.

**Mode** (Mod) Spectrum'da klavyedeki her tuşla üretililecek komut veya karakterlerin elde edilebilecekleri beş durumdan biri. Program sırasında modun hangisi olduğunu göstergeçte yanıp sönen harf gösterir.

**Nesting** (Yuvalama) Programda bir döngünün ya da birkaçının bir başkasının içinde uygulanmasını sağlayan düzenleme.

**Numeric variable** (Sayısal değişken) Değişken bir sayı içerir. Sayısal değişkenler, bir veya birkaç harften oluşur.

**Operator** (İşlem) Aritmetik ya da mantık işlemleri yapılmasını sağlayan talimat.

**Output** (Çıktı) bilgisayarın sağladığı sonuçlar.

**Peripheral** (Yan donanım) Bilgisayara bağlı herhangi bir araç.

**Pixel** Ekranda görünebilecek en küçük renkli nokta. İngilizce picture cells – resim hücrelerinin kısaltılması.

**Print** (Yaz) Ekranda sonuçları veya grafik işaretleri yansıtan ya da yazıya yazdıran emir.

**Program** (Program) Bilgisayarın uygulayacağı bir sıra talimat.

**RAM** (Random Access Memory) (Rastal Ulaşılan Bellek) Belleğin bir program, veriler veya diğer değişkenler depolayabileceği kesimi. Ayrıca değişken bellek diye de bilinir. Cereyan kesildiği zaman RAM'in içeriği silinir. ZX Spectrum + 'in 48K'lık bir RAM'i vardır.

**Register** (Kütük) Ana bellek dışında bir küçük bellek birimi. CPU içindeki kütükler,

bilgisayar işlemlerinin yapılmasında kullanılır.

**Report** (Mesaj) Bilgisayarın bazı eylemlerini haber veren, ekrana yansıtılan bilgiler.

**Resolution** (Tanım) Bilgisayar çizimlerinde ne ölçüde ayrıntı verilebileceğinin göstergesi.

**ROM** (Read Only Memory) (Sadece Okuyucu Bellek) Belleğin kalıcı bazı programlar ya da bilgisayar için talimatlar içeren kesimi. ZX Spectrum + 'un 16K'lık bir ROM'u vardır.

**Save** (Sakla) Bir program veya verinin kartuş veya kaset gibi bir araçta depolanması.

**Scroll** (Kaydır) Bir ekrandan fazlasını doldurabilecek bilginin ekrana yansımaları sağlayacak hareket.

**Software** (Yazılım) ROM ve kartuşlardaki daimi programlar da dahil herhangi bir program.

**Statement** (Cümle) Bir program satırında bir talimat oluşturmaya üzere kullanılan bir komut veya talimatın kendisi.

**String** (Dizi) Rakamlardan veya sayısal değişkenlerden ayrı tutulmak üzere olmak üzere işaretleri içine alınmış bir veya birkaç karakterden oluşan grup.

**String variable** Bir dizi içeren bir değişken. Dizi değişkenler daima bir harf ve \$ işaretinden oluşur.

**Syntax** (Söz dizini) Geçerli bir BASIC talimatı oluşturmak üzere, doğru bir biçimde sıralanmış komutlar, değişmezler, değişkenler ve ifadeler.

**True** (Doğru) Bilgisayarın doğru veya geçerli saydığı bir durum veya sonuç. Doğrunun sayısal değeri 1'dir.

**Value** (Değer) Bir değişmez, değişken veya ifade tarafından verilen veya temsil edilen bir sayı veya dizi.

**Variable** (Değişken) Bilgisayar tarafından kullanılmak üzere belli bir değişmezi saklayan, belleğin bir veya birkaç birimi. Her birine kolayca tanımlanabilmesi için bir ad veya bir harf verilmiştir. ZX Spectrum + , sayısal değişkenlerle dizi değişkenler arasındaki farkı ayırt eder.



## Dizin

İtali olarak verilen sayfa numaraları, resimler veya başlıklara işaret eder.

Adlar programı 8  
Alçak tanımı çizimler 26-7  
Altprogram 30-1  
Anten girişi ve bağlantısı 4-5  
Ara çubuğu (tuşu) 19  
Araparçalar (interface) 45, 46-7  
Animetik işlemi 22, 22  
ATTR 35

Bağlantılar 5  
Elektrik 5  
Kasetli teyp 13  
Televizyon 4  
Bağlantısı Mantık Düzeni (ULA) 42  
BASIC 18, 49-73  
BEEP 36, 18  
Belek 12, 42, 43, 44-8  
Belek haritası 48  
BIN 33  
BREAK 19  
BRIGHT 31  
Büyük harf modu 21, 20

CAPS LOCK 21, 18  
CAPS SHIFT 8, 21, 18  
Cereyan bağlantısı 4, 5, 43  
CIRCLE 28  
Cümleler 22, 50  
Çarpım Tablosu programı 23  
Çarpma 34-5  
Çerçeve rengi 24-5, 6  
Çiğın mozaik programı 10  
Çözge, alçak tanımı 26, 80  
yüksek tanımı 28, 80  
Çizimler ve hareketli görüntüler 34-5  
alçak tanımı 26-27  
desenler 30-1  
karakter yaratıcısı 32-3  
rastsal etkiler 30  
renk 24-5  
şekillerin doldurulması 29, 29  
yüksek tanımı 26, 28-9  
Çokgen programı 10  
Çubuklu grafikler 25, 25

DATA 33  
Değişkenler 22-3, 50  
DELETE 10  
Depolama 44, 45  
Desenler programı 9  
Dizi 22  
Doğan güneş programı 11  
9VDC girişi 5, 43  
Donanım, tanımı 12  
Dongüler 26-7, 30  
DRAW 28-9  
Düzeltilme, hataların 10, 21  
EAR (kulaklık) girişi 37, 5, 13

EDIT (düzeltme) 18, 21  
program satırı 21  
Ekranındaki mesajlar 74  
Emirler 22, 50  
ENTER 9, 10, 11, 19  
EXTEND MOOE 8, 21, 18  
Genişletilmiş mod 21, 20

FLASH 31  
FOR NEXT 26-7, 29, 30, 31, 34  
FORTH 75  
Girdi-çıkışı yolları 45  
Ginsler 5  
GOTO 23  
Gökkuşuğu programı 26-7  
Görüntü kontrolü 19  
Grafik modu 21, 20  
GRAPH 21, 18, 26

Hareketli görüntüler (animasyon) 34-5  
Harf modu 21, 20  
Hatalar, düzeltilme 10, 21  
ekran mesajları 74  
Hızlı Yanım 12, 13, 13  
Hesaplama 22-3, 22, 23  
IF THEN 29  
INPUT 23, 29  
INV VIDEO 18  
INVERSE 31  
İşaretler, hesaplama 22, 50  
Seçimi 19  
İkili kodlar 44  
İşlevler 50  
İşleme koyma, programların 8-9

Kağıt rengi 24-5  
Karakter tabloları 51  
Karakterler, seçimi 20  
yapımı 32-3  
Kareler programı 30  
Kartuşlar, Microdrive 12, 46, 46  
ROM 12, 47, 47  
Kaset 12, 44, 45  
Bakımı 12  
Etiketlenmesi 14  
Saklanması 12  
Sesi 12  
Kasetli teyp, amplifikatör olarak 37, 37

Bağlanış 5-13, 13  
Program saklama 38-40  
Program yükleme 14-16  
Sayacı 14  
Seçimi 12  
Ses kontrolü 14, 15, 16  
Ton kontrolü 14, 15, 16  
Klavye 18-19  
Grafik karakterleri 26-7  
Modlar 20-1  
Komut modu 20, 20  
Kartuşlar 9, 18-19, 50, 52-73, 20-1  
Seçimi 19-20  
Kullanımın tanımladığı karakterler 80, 32-3  
Kundela kablo 46

LET 23  
LIST 21  
LOAD 14-16  
LOGO 75  
Makina kodu 75  
Mantık yongaları 43  
Merkezi İşlem Birimi (CPU) 43-44, 48, 75, 43, 45  
MIC girişi 37, 5, 13  
Microdrive 46, 5, 46  
Kartuşları 12, 45  
Yüklenmesi 46  
Micro-PROLOG 75  
Modemler 46  
Modlar 20-1  
Mürekkep rengi 24-5  
Müzik 36-7

NEW 11, 12, 18  
Nokta 23, 51  
Noktalama işaretleri 23, 51  
Noktalar virgül 23, 51  
Oparlar 43  
Oyun kolu (joy stick) 45, 47  
Parantez 23  
Parazit 4  
Piramit programı 31  
Pixel 28  
PLOT 28-28  
POKE 48  
PRINT 22

Program sabırları, sıma 21  
Düzeltilme 21  
Program verme 8-9  
Programda değişiklik 9  
Programlama 17-40  
Programlar  
Değiştirme 9  
Doğrulama 39  
Hataları düzeltilme 10  
İşleme koyma 8-9, 44  
Saklama 13, 38-40  
Verme 8-9, 44  
Yeniden başlatma 10  
Yenisine başlatma 11  
Yükleme 12-13, 14-15, 14-15

Programların yeniden başlatılması 10  
Rakam tuşları 19  
Rakamlar 50  
RAM (Rastsal Kullanım Belleği) 42-48, 42, 45  
RAM takımı 4  
RAMTOP 48  
Rastsal etkiler 9  
READ 33  
REM 39  
Reset düğmesi 11, 12, 5  
Resim defteri programı 29  
Resimler, çizimi 30-1  
Yüksek tanımı 26-7  
RND 26, 30  
ROM (Sadece Okuyucu Bellek) 48, 43, 45  
ROM kartuşları 12, 47, 46-7  
RS232 Araparçası 47, 45

Saklama 13, 38-40  
Satırlar 8  
İptal etme 21  
Düzeltilme 21  
Satıran programı 51  
SAVE 38-9  
Scroll (kaydırma) 8  
Seçme, işaretler 20  
Ses düğmesi, kasetli teyp 12, 14, 15  
Ses efektleri 36-7  
Sesi yükseltme 37  
Sıralama 8-21  
Simetrik desenler programı 30  
Sinclair BASIC 49-73  
Sistem değişkenleri 48  
STEP 29  
SYMBOL SHIFT 8, 21, 19  
Şekiller, içini doldurma 29, 29

Televizyon alıcısının ayarlanması 6, 6  
Televizyon, bağlantısı 5  
Uygunluğu 4  
Ayarlanması 6, 6  
Tini, müzikte 36  
Tımak işaretleri 23, 51  
Ton düğmesi, kasetli teyp 12, 14, 15  
TRIPLE VIDEO 18  
Tuşlar 18-19, 18, 19  
Kullanımı 20-1, 20-1  
Tuşları kullanarak yazma 8, 9  
TV kodlayıcısı 42  
Voltaj regülatörü 43  
Yan bağlantısı 5, 43, 47  
Yan donanımlar 45, 46-7  
Yanıp sönen halkalar programı 9

Yazıcılar 45, 47, 45, 47  
Yazılım 12  
kullanıma hazır 12-13, 13  
türleri 12  
uygunluk 12  
yükleme 14-16, 14-16  
Yeni programlar 11  
Yıldız programı 28  
Yıldızlar ve çubuklar programı 11  
Yongalar 42-3  
Yükleme 13, 14-15, 14-16  
Yüksek tanımı çizimler 26, 28-9  
Ziplayan top programı 35  
Z80 mikrobilgisayar 43, 75, 45  
ZX interface 1, 45, 46-7  
ZX Robot programı 27  
ZX 16K RAM 4

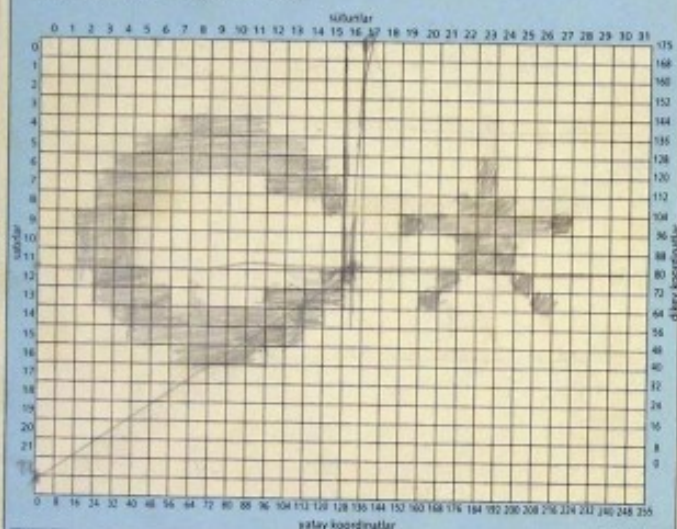
İlk yayımı 1984 – Yayınlayan: Dorling Kindersley Ltd, 9 Henrietta Street, London WC2E 8PS – Sinclair Research Ltd, 25 Willis Road, Cambridge ile işbirliği yapılmıştır.

Copyright © 1984 Sinclair Research Ltd ve Dorling Kindersley Ltd, London'a aittir. Resimler için copyright © 1984 Dorling Kindersley Ltd, London'a aittir.  ZX Spectrum +, ZX Spectrum Microdrive ve ZX Interface Sinclair Research Limited'in tescilli markalarıdır.

Bütün hakları mahfuzdur. Copyright sahiplerinin yazılı izni olmadan elektronik, mekanik, fotokopi, kayıt ve başka herhangi bir yöntemle, bu yayının hiç bir kısmı çoğaltılamaz, bilgisayarda saklanamaz veya yayınlanamaz.

Yayımcı: David Burnie  
Sanat yönetmeni: Peter Luff  
Tasarımcı: Debra Lee  
Fotoğrafçı: Trevor Melton  
Ekran fotoğrafları: Vincent Oliver  
Yönetim sorumlusu: Alan Buckingham  
Dizgi: The Letter Box Company (Woking) Ltd., Woking, Surrey, England  
Reproduksiyon: A. Mondadori, Verona  
Basım ve cilt: A. Mondadori, Verona, İtalya

Yüksek tanımlı çizim çizelgesi sadece ana



Kullananın tanımladığı grafikler çizelgesi

Bu çizelgenin her sırasının ondalık değerini elde etmek için sayıları toplayın ve karakteri programlamak için POKE USR'yi kullanın.

[illegible]



**SPECTRUM YAZILIMLARI**  
Spectrum bilgisayarları için  
kullanılabilen yazılımların tümü yeni  
ZX Spectrum+ 'unuza da uyar.

DORLING KINDERSLEY LTD  
SINCLAIR RESEARCH LTD'nin  
işbirliği ile