

COMMODORE
AYLIK BILGISAYAR DERGİSİ Mart 1986

PROGRAM DÖKÜM EKİ

- Commodore 64'te yazı karakteri tanımlama
- Ekranda dünyamızı çiziyoruz
- Makine dili editörü
- Hyperscreen

Commodore'un Ücretsiz Ekidir

Sevgili okurlar,

Derginiz Commodore her sayısında řu anda elinizde tutmakta olduėunuz program döküm ekini ücretsiz olarak verecektir. Bu bölümde yazılımını kendi kadromuzun yaptığı, ya da yabancı kaynaklardan çevirdiğimiz programlar yer alacak. Yalnız, bu son cümleye bir de “řimdilik” kaydını eklemek istiyoruz.

Neden “řimdilik?” Çünkü, inanıyoruz ki sizler de çok ilginç programlar yazabilirsiniz, çoėunuz halen yazıyorsunuz da... Ve yine inanıyoruz ki, özgün programlarınızın Commodore tarafından yayınlanmasını arzularsınız.

Dergide yer almasını istediėiniz programları print-out halinde adresimize postalayın, yanına bir açıklama metni eklemeyi de unutmayın. Eėer henüz bir printer sahibi olamadıysanız ve kullanımınıza elverişli bir printer bulmakta da güçlük çekiyorsanız, açıklama metninizi hazırlayın, disketinizi yanınıza alın ve dergimize gelin; size yardımcı olalım.

Eėer programınızı yayınlarsak, yayın tarihini izleyen günlerde adresinize 15 000 TL postalayacaėız.

Göndereceėiniz programlar için adresimiz:

Commodore Dergisi
Program/Döküm Bölümü
Silahhane Cad. No:59
Ralli Apt., 80200
Teşvikiye/İstanbul

PROGRAM DÖKÜM



Tahmin edebileceğiniz gibi dergimizin can damarlarından birini her sayıda bu sayfalarda yayınlayacağımız listingler (ya da dökümler) oluşturacak. (Bu fırsattan istifade size dil 'politikamız' hakkında da birkaç şey söyleyelim. Aslında söyleyecek pek o kadar şey de yok; çünkü bu konudaki tavrımız bir cümleyle özetlenebilir: Belli bir dil politikamız yok. Bilgisayar alanındaki terimlerin yerleşmiş, üzerinde herkesin anlaştığı karşılıklarının olmadığı hepimizce malum. Sevmekle birlikte, biz kendimiz bilgisayar olmadığımıza göre, belli bir kavramı anlamak için sabit bir işaretler dizisine ihtiyacımız yok. İletişim sağlandığı, birbirimizi anlayabildiğimiz sürece isteyen, bilgisayar der, isteyen printer, isteyen yazıcı.)

Biz bu sayfalar için, elimizden geldiğince, Commodore'ların çeşitli yönlerinden azami ölçüde yararlanmanızı sağlayacak dökümler hazırlamaya, kendimiz hazırlayamadığımızda çevirmeye çalışacağız. Ama bu konuda asıl kaynağımızın siz okurlarımız olacağını umuyoruz. Eninde sonunda önemli olan, derginin diğer sayfalarında olduğu gibi burada da, sizin birbirlerinizle buluşmanız. Gösterin ustalığınızı!

Ayrıca bu sayfada, sizi kullanım kılavuzlarına ya da bu derginin bu sayısına defalarca başvurmaktan kurtarmak için, dökümlerde sık sık rastlanan özel karakterlerin nasıl girileceğini gösteren tabloyu her ay yayınlayacağız. Aşağıdaki tabloda, soldaki şekiller listinglerde basılı olarak göreceğiniz; sağda ise sizin bu şekli elde etmek için ne yapmanız gerektiği yazılı.

Anahtar	Görüntü
CLR/HOME	S
SHIFT CLR/HOME	♥
↑ CRSR ↓	Q
SHIFT ↑ CRSR ↓	●
← CRSR →	]]
SHIFT ← CRSR →	

Anahtar	Renk	Görüntü
CTRL 1	Siyah	■
CTRL 2	Beyaz	E
CTRL 3	Kırmızı	£
CTRL 4	Çivit	▤
CTRL 5	Pembe	▥
CTRL 6	Yeşil	↑
CTRL 7	Mavi	—
CTRL 8	Sarı	π
⌘ 1	Turuncu	♠
⌘ 2	Kahverengi	▦
⌘ 3	Açık kırmızı	⊗
⌘ 4	Gri 1	○
⌘ 5	Gri 2	♣
⌘ 6	Açık yeşil	▨
⌘ 7	Açık mavi	◆
⌘ 8	Gri 3	⊕

MAKİNE DİLİ EDITÖRÜ

Tuna ERTEMALP

Makine Dili Editörü (kısaca MDE), makine dilinde yazılmış programları daha kolay yazabilmeniz için hazırlanmış bir yardımcı programdır. MDE ile uzun DATA satırları yazmaktan artık kurtulmuş olacaksınız. Öncelikle 3 basamaklı sayılar yerine sadece 2 basamaklı 16'lık sistemde hesaplanmış sayılar gireceksiniz. Üstelik yazdığınız program bitince, programınız kesinlikle (!!) çalışacak; hiçbir hata vermeden...

Bunu mümkün kılan, her satırın sonunda vereceğiniz bir kontrol toplamı (check sum) olacak. Bunu açıklamak için MDE'nin nasıl çalıştığını biraz görelim.

Dergide, MDE ile yazacağınız programların genel görüntüsü aşağıdaki gibi olacaktır:

program: program adı 0801 16f7

```
0801 : 08 01 00 00 9e 32 30 36 33
0809 : 34 3a a2 00 00 00 49 a9 7b
0811 : 8e 20 d2 ff a2 ff e8 bd ad
```

İlk bakışta oldukça garip gelen bu program biçiminin aslında oldukça basit ve hızlı bir yazım sağladığını, MDE'yi kullanmaya başladığınızda hemen anlayacaksınız. Şimdi MDE'nin disket veya kaset üzerinde hazır olduğunu varsayalım. LOAD "MDE", 8 veya LOAD

"MDE", 1 ile kullandığınız kayıt aracından MDE'yi bilgisayarınıza yükleyin ve RUN komutunu verin. Ekranın renkleri değişecek ve MDE sizi açılış ekranı ile selamlayacaktır. İlk olarak size yazılacak programın adı sorulacaktır. Buna karşılık olarak, dergideki program listesinin ilk satırındaki 'program adı' olarak belirtilen adı yazın. Şimdi de programın bilgisayarın iç belleğinde hangi adresten başlaması gerektiği sorulacaktır. Buna da listenin ilk satırında verilen iki sayıdan ilkinin verin. Örneğin 0801 gibi.

Bir sonraki soru ise, programın bitiş adresini isteyecektir. Buna da ilk satırındaki ikinci sayı ile cevap verin. Örneğin 16f7 gibi. Sırası gelmişken daha önce 16'lık sistemle hiç ilişkisi olmamış okuyucularımız için bu sayıların ne anlama geldiğini biraz açıklayalım.

Özellikle makina dili ile yapılan programların bu sistem ile gösterilmesi, programcılar arasında bir alışkanlık haline gelmiştir. Ayrıca bu sistem, normalde kullandığımız onluk sistemde gösterdiğimiz sayıların, daha az basamakla gösterilmesini de sağlar. Bu sistemde kullanılan sayılar, 0-9 ve A-F karakterlerinden oluşur. Genellikle sayının başına \$ işareti konularak, bu sayının 16'lık sistemde yazıldığı belirtilir. \$A ile 10, \$F ile de 15 belirtilmektedir.

16'lık sistemdeki bir sayıyı onluk sisteme çevirmek için şu işlemler yapılır:

$$\begin{aligned} SE8 &= 16 \cdot 14 + 1 \cdot 8 = 232 \\ SF02C &= 4096 \cdot 15 + 256 \cdot 0 + 16 \cdot 2 + 1 \cdot 16 = 61484 \end{aligned}$$

Şimdi tekrar MDE'ye dönelim. Tüm soruları cevapladıktan sonra, ekran değişecektir. Artık ekranın en üst satırında programın adı, başlangıç ve bitiş adresleri yazılacaktır. Daha aşağıda ise başlangıç adresi yazılacak ve iki nokta üstüste konularak sizin sayıları girmeniz beklenecektir. Şimdi sırasıyla (yukarıdaki örneğimiz için) 08 01 00 00 9E 32 30 36 sayılarını girin. Göreceğiniz gibi, bilgisayar her iki basamağı yazdığınızda kendiliğinden bir boşluk yazarak bir karakter sağa kayacaktır. İlk sekiz sayıyı yazdıktan sonra sıra o satırın kontrol toplamına gelecektir. Örneğimizde, 33 sayısını yazınca bilgisayar tatlı bir gong sesi çıkararak o satırı doğru yazdığınızı belirtecek, kendiliğinden bir alt satıra geçip bir sonraki satır numarasını verecek ve yeni satırı girmenizi bekleyecektir. Ancak ilk satırı girerken bir hata yaptıysanız veya kontrol toplamını yanlış yazdıysanız, dikkatinizi çekecek bir ses sinyali verilecek ve o satırdaki hatalı sayıyı düzeltmeniz beklenecektir. Hatalı olan sayıya gelinceye kadar DEL

tuşuna basın ve o sayıyı ve ondan sonra gelen satırın geri kalan kısmını yeniden yazın. Bilgisayar yazdığınız satırı ancak doğru olarak yazdığınızda kabul edecektir. Gördüğünüz gibi, MDE ile program yazarken ekrana bakmanız bile gerekmez. Her satırda çıkan uyarı sinyalleri, sadece dergiye ve klavyeye bakarak yazmanızı sağlayacaktır.

Programın en son satırını da yazdığınız zaman MDE size programı nereye kaydedeceğini soracaktır. Ekranın sol üst köşesinde belirecek bu soruya, D (disket için) veya T (teyp için) cevap verin. SAVE işlemi bitince bilgisayar yeniden başlangıç adresini vererek başa dönecektir. Artık RUN STOP/RESTORE ile çıkıp programınızı disket veya kasetten doldurup, o programın açıklamalarında belirtildiği gibi çalıştırabilirsiniz.

MDE'nin bazı değişik özellikleri de vardır. Şimdi de bunları görelim.

Diyelim ki çok uzun bir program yazmanız gerekiyor ve bu programı bir kerede yazamadınız. Programın yazımı sırasında istediğiniz zaman programınızı CTRL ve S tuşlarına basarak SAVE edebilirsiniz. Ekranın sol üst köşesinde çıkacak soruya D veya T ile cevap verin. Artık bilgisayarınızı kapatıp diğer işlemlerinizi uğraşabilirsiniz.

Başka bir zaman, yazmaya devam etmek istediğinizde MDE'yi yükleyin ve

RUN komutunu verin. Programın adını verdikten sonra başlangıç adresi yerine L harfini verin. Böylece MDE, programı LOAD etmesi gerektiğini anlayacaktır. Ekranın sol üst köşesinde çıkacak soruya D veya T ile cevap verin. Programın yüklenmesi bittiğinde, başlangıç ve bitiş adresleri kendiliğinden hesaplanacaktır.

Şimdi de CTRL ve M tuşlarına basın, çıkacak soruya, ekranın sağ üst köşesinde göreceğiniz başlangıç adresi ile cevap verin. Şimdi programınızın yazımını hangi satırda bıraktığınızı bulmaya çalışacaksınız. Bunun için BOŞLUK tuşuna basın. Daha önce yazdığınız satırlar sırayla LIST edilecektir. Sizin yazmayı bıraktığınız satır ise tamamen AA kodu ile dolu olacaktır (kontrol toplamı hariç). O satırın numarasını aklınızda tutun. Şimdi RUN/STOP tuşuna basarak listeyi durdurun. CTRL ve N tuşlarına basarak yeni başlangıç adresi olarak demin bulduğunuz satır numarasını yazın. Artık yazmaya, kaldığınız yerden devam edebilirsiniz.

Bir diğer emir ise CTRL-P'dir. Bu emir ile, kendi yazdığınız makine dili programlarını MDE formatında bir yazıcıya bastırabilirsiniz. Bunun için MDE'yi çalıştırdıktan sonra, programın ismini yazın ve başlangıç adresi için L harfini verin. Yükleme işlemi bitince de CTRL ve

P tuşlarına aynı anda basarak baskı işlemini başlatın.

Makine dili editörünü yazma

Bu sayımızda MDE'yi oluşturan bir program yayınlıyoruz. Ne yazık ki bu çok büyük kolaylık sağlayan programın kendisi oldukça uzun. Bu nedenle daha az sayı yazmanız için tüm DATA satırlarını 16'lık sistemde hazırladık. Ayrıca her DATA satırının sonuna bir kontrol toplamı yerleştirdik. Bu programı yazdıktan sonra muhakkak (!!) kasete veya diskete kaydedin. Çünkü programı çalıştırdığınızda, eğer tüm DATA satırlarını doğru olarak yazmışsanız, program kendi kendini silerek yerine asıl MDE programını oluşturacaktır.

Şimdi RUN komutunu vererek programı çalıştırın. Hangi DATA satırının okunduğu gösterilecektir. Eğer hatalı bir satır varsa, satır numarası gösterilecektir. O satırı düzelttikten sonra yeniden RUN komutunu verin. Tüm DATA satırları okunduktan sonra bilgisayar işini bitirecektir. Artık oluşan programı SAVE "MDE" veya SAVE "MDE".8 ile kaydedebilirsiniz.

Hepinize bol programlı günler...

```
0 dimh(75):for i=0to9
1 h(48+i)=i:h(65+i)=i+10:next
2 for i=49152 to 50911:reada$
3 h=asc(left$(a$,1)):l=asc(right$(a$,1))
4 d=h(h)*16+h(l):s=s+d:pokei,d
5 a=a+1:ifa<20thennext:a=-1
6 print"satir:"!000+z;
7 readv:z=z+1:ifv=s then9
8 print"toplama hatasi! satir:"999+z:stop
9 ifa<0thensys50865
10 s=0:a=0:print:next:sys50865
1000 data00,0b,08,0a,00,9e,32,30,36,31,00,00,00,a2,08,a9,36,85,a4,a9, 1247
1001 data08,85,a5,a9,00,85,a6,a9,b0,85,a7,a0,00,b1,a4,31,a6,c8,d0,f9, 2888
1002 datae6,a5,e6,a7,ca,d0,f2,a9,36,85,01,4c,00,b0,20,d1,b1,a9,06,8d, 2787
1003 data21,d0,a9,03,8d,20,d0,8d,86,02,a0,b3,a9,74,20,ff,b1,a0,b3,a9, 2667
1004 datab9,20,ff,b1,a0,00,20,cf,ff,99,01,02,c8,c9,0d,d0,f5,88,f0,d2, 2912
1005 datac0,0f,90,02,a0,0e,8c,00,02,20,ea,b1,a0,b3,a9,cf,20,ff,b1,20, 2323
1006 data8e,b4,85,fc,85,62,20,8e,b4,85,fb,85,61,20,a7,b4,d0,20,a0,b3, 2864
1007 dataa9,e5,20,ff,b1,20,8e,b4,85,60,20,8e,b4,85,5f,20,a7,b4,d0,0a, 2624
1008 dataa5,61,c5,5f,a5,62,e5,60,90,06,20,43,b3,4c,3a,b0,a9,aa,a0,00, 2379
1009 data91,fb,e6,fb,d0,02,e6,fc,20,3f,b2,90,ef,4c,fb,b4,a2,02,86,58, 3118
1010 dataa9,a6,a0,9d,20,f2,b1,20,e4,ff,f0,fb,c9,30,90,0c,c9,47,b0,08, 2970
1011 datac9,3a,90,0b,c9,41,b0,07,c9,14,d0,0f,4c,0b,b1,20,d2,ff,a6,58, 2322
1012 data95,f7,c6,58,d0,d2,60,ae,8d,02,f0,26,c9,0c,d0,03,4c,0b,b6,c9, 2685
1013 data13,d0,03,4c,8b,b5,c9,0d,d0,03,4c,ba,b4,c9,10,d0,03,4c,68,b5, 2282
```

```

1014 datac9,0e,d0,06,20,5f,b4,4c,64,b1,4c,92,b0,a5,f9,20,02,b1,0a,0a, 2132
1015 data0a,0a,85,f9,a5,f8,20,02,b1,05,f9,60,c9,3a,90,02,69,08,29,0f, 1950
1016 data60,a6,59,e0,08,90,1f,a6,58,e0,02,b0,06,20,d2,ff,4c,8e,b0,c6, 2509
1017 data59,a0,14,a9,92,20,f2,b1,ca,d0,fa,84,57,68,68,4c,8b,b1,a6,d3, 2891
1018 datae0,08,b0,03,4c,92,b0,20,d2,ff,a6,58,e0,02,90,09,c6,59,20,d2, 2468
1019 dataff,c6,58,d0,f9,4c,8e,b0,48,4a,4a,4a,4a,20,59,b1,68,29,0f,c9, 2419
1020 data0a,90,02,69,06,69,30,4c,d2,ff,a2,fc,9a,20,d1,b1,20,48,b2,20, 2261
1021 dataea,b1,20,9f,b2,a5,fc,20,4e,b1,a5,fb,20,4e,b1,20,ed,b1,a9,3a, 2860
1022 dataa0,20,20,f2,b1,a9,00,85,59,20,8e,b0,20,ed,b1,a4,59,20,ef,b0, 2530
1023 data91,fb,c8,84,59,c0,08,90,ec,20,10,b2,a9,12,20,d2,ff,20,8e,b0, 2657
1024 data20,ef,b0,c5,ff,f0,0d,20,43,b3,a9,14,a0,14,20,f2,b1,4c,a2,b1, 2665
1025 dataa9,92,20,d2,ff,20,33,b2,20,e0,b2,20,3f,b2,90,9f,4c,8b,b5,a9, 2648
1026 data93,20,d2,ff,a2,00,a9,03,9d,00,d8,9d,00,d9,9d,00,da,9d,00,db, 2476
1027 datae8,d0,ef,60,a9,0d,2c,a9,20,4c,d2,ff,20,d2,ff,98,4c,d2,ff,20, 2965
1028 datae4,ff,f0,fb,60,84,5d,85,5c,a0,00,b1,5c,f0,06,20,d2,ff,c8,d0, 3100
1029 dataf6,60,a5,fb,85,5a,a0,00,84,5b,b1,fb,18,65,5a,85,5a,90,02,e6, 2606
1030 data5b,06,5a,26,5b,c8,c0,08,90,ec,a5,5a,65,5b,85,ff,60,18,a5,fb, 2467
1031 data69,08,85,fb,90,02,e6,fc,60,a5,fb,c5,5f,a5,fc,e5,60,60,a0,b3, 3106
1032 dataa9,fb,20,ff,b1,a0,01,b9,00,02,20,d2,ff,cc,00,02,c8,90,f4,a9, 2692
1033 data10,ed,00,02,aa,20,ed,b1,ca,d0,fa,a5,62,20,4e,b1,a5,61,20,4e, 2453
1034 datab1,20,ed,b1,a5,60,20,4e,b1,a5,5f,20,4e,b1,a9,9f,20,d2,ff,20, 2575
1035 dataea,b1,24,5e,10,01,60,a9,12,20,d2,ff,a2,28,20,ed,b1,ca,d0,fa, 2646
1036 dataa9,92,4c,d2,ff,a5,d6,c9,16,b0,01,60,a9,a0,85,a4,a9,78,85,a6, 2945
1037 dataa9,04,85,a5,85,a7,a2,13,a0,27,b1,a4,91,a6,88,10,f9,ca,f0,19, 2671
1038 data18,a5,a4,69,28,85,a4,90,02,e6,a5,18,a5,a6,69,28,85,a6,90,e0, 2503
1039 datae6,a7,4c,b6,b2,a9,91,4c,d2,ff,a9,0f,8d,18,d4,a9,00,8d,05,d4, 2776
1040 dataa9,f7,8d,06,d4,a9,11,8d,04,d4,a9,32,8d,01,d4,a9,00,8d,00,d4, 2413
1041 dataa0,80,20,09,b3,a9,10,8d,04,d4,60,a2,ff,ca,d0,fd,88,d0,f8,60, 2914
1042 dataa9,0f,8d,18,d4,a9,2d,8d,05,d4,a9,a5,8d,06,d4,a9,21,8d,04,d4, 2385
1043 dataa9,07,8d,01,d4,a9,05,8d,00,d4,a0,ff,20,09,b3,a9,20,8d,04,d4, 2250
1044 dataa9,00,8d,01,d4,8d,00,d4,60,38,20,f0,ff,8a,48,98,48,18,a0,06, 2179
1045 dataa2,18,20,f0,ff,a0,b4,a9,0a,20,ff,b1,20,12,b3,20,e4,ff,f0,fb, 2931
1046 dataa2,1d,a9,14,20,d2,ff,ca,d0,fa,68,a8,68,aa,18,4c,f0,ff,0d,0d, 2704
1047 data0d,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20, 939
1048 data4c,49,20,45,44,49,54,4f,52,55,20,20,20,20,20,0d,0d,20,20,20, 1003
1049 data20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20, 640
1050 data20,20,20,20,20,20,20,00,0d,0d,0d,20,20,20,50,52,4f,47,52,41,4d, 863
1051 data20,41,44,49,20,20,3a,20,00,0d,0d,20,20,20,42,41,53,4c,41,4e, 947
1052 data47,49,43,20,41,44,2e,3a,20,24,00,0d,0d,20,20,20,42,49,54,49, 966
1053 data53,20,41,44,52,45,53,49,20,3a,20,24,00,92,05,20,50,52,4f,47, 1208
1054 data52,41,4d,20,20,3a,20,00,12,20,20,2a,2a,2a,20,20,20,59,41,4e, 914
1055 data4c,49,53,60,56,45,52,49,20,20,20,2a,2a,2a,20,20,92,00,0d,0d, 1096
1056 data2a,2a,2a,20,53,4f,4e,20,2a,2a,2a,20,00,13,05,20,20,12,44,92, 908
1057 data49,53,4b,20,56,45,59,41,20,12,54,92,45,59,50,0d,00,13,20,20, 1186
1058 data49,2f,4f,20,2d,20,48,41,54,41,53,49,00,20,d1,b1,20,48,b2,a0, 1610
1059 datab3,a9,cf,20,ff,b1,20,8e,b4,85,fc,20,8e,b4,85,fb,c5,61,a5,fc, 3207
1060 datae5,62,90,23,a5,fb,c5,5f,a5,fc,e5,60,b0,19,20,a7,b4,d0,14,60, 2860
1061 data20,a7,b4,f0,0c,85,f9,20,a7,b4,f0,05,85,f8,4c,ef,b0,68,68,20, 2749
1062 data43,b3,4c,5f,b4,20,cf,ff,c9,4c,d0,09,20,d1,b1,20,48,b2,4c,0b, 2372
1063 datab6,c9,0d,60,a9,00,85,5e,20,5f,b4,20,ea,b1,20,0d,b5,24,5e,30, 2042
1064 data05,20,e4,ff,f0,fb,20,e1,ff,f0,26,20,9f,b2,24,5e,10,09,20,4e, 2435
1065 datab5,20,0d,b5,20,60,b5,20,33,b2,20,3f,b2,90,d7,a0,b4,a9,28,20, 2190
1066 dataff,b1,20,e4,ff,c9,0d,d0,f9,a9,00,85,5e,a5,61,85,fb,a5,62,85, 3056
1067 datafc,20,e0,b2,4c,64,b1,a5,fc,20,4e,b1,a5,fb,85,ff,20,4e,b1,a9, 3003
1068 data20,a0,3a,20,f2,b1,a0,00,20,ed,b1,b1,fb,20,4e,b1,c8,c0,08,90, 2566
1069 dataf3,20,ed,b1,24,5e,30,03,a9,12,2c,a9,20,20,d2,ff,20,10,b2,a5, 2190

```

b9

b9

```

1070 dataff,20,4e,b1,a9,92,20,d2,ff,4c,ea,b1,a9,ff,85,b8,85,b9,a9,04, 3073
1071 data85,ba,20,c0,ff,a2,ff,4c,c9,ff,20,cc,ff,a9,ff,4c,c3,ff,20,5f, 3315
1072 datab4,a9,80,85,5e,20,4e,b5,20,48,b2,a2,24,a9,2d,20,d2,ff,ca,d0, 2596
1073 datafa,20,ea,b1,20,ea,b1,20,60,b5,4c,c1,b4,20,b8,b5,a6,5f,a4,60, 2812
1074 dataa9,61,20,d8,ff,b0,0a,20,b7,ff,29,bf,d0,03,4c,fb,b4,a9,01,20, 2577
1075 datac3,ff,20,68,b6,a0,b4,a9,4f,20,ff,b1,20,f9,b1,4c,fb,b4,20,68, 2921
1076 datab6,a9,37,a0,b4,20,ff,b1,20,f9,b1,a2,08,c9,44,f0,06,a2,01,c9, 2717
1077 data54,d0,f1,a9,01,a8,20,ba,ff,a0,00,e0,01,f0,1a,a9,40,8d,20,02, 2403
1078 dataa9,3a,8d,21,02,b9,01,02,99,22,02,c8,cc,00,02,90,f4,c8,c8,d0, 2182
1079 data0c,b0,01,02,99,20,02,c8,cc,00,02,d0,f4,98,a2,20,a0,02,4c,bd, 2018
1080 dataff,20,b8,b5,a5,ba,c9,08,90,33,a6,b9,86,57,a9,01,20,c3,ff,a9, 2800
1081 data60,85,b8,20,c0,ff,b0,28,a5,ba,20,b4,ff,a5,b9,20,96,ff,20,a5, 2911
1082 dataff,85,51,a5,90,4a,4a,b0,13,20,a5,ff,85,62,20,ab,ff,a5,57,85, 2663
1083 datab9,a9,00,20,d5,ff,30,03,4c,a3,b5,86,5f,84,60,a5,ba,c9,01,d0, 2633
1084 data0a,ad,3d,03,85,61,ad,3e,03,85,62,4c,fb,b4,a9,13,20,d2,ff,a2, 2300
1085 data1c,20,ed,b1,ca,d0,fa,60,aa,aa,aa,aa,46,ad,00,c0,8d,00,08,ee, 2732
1086 datab2,c6,ee,b5,c6,d0,f2,ee,b3,c6,ee,b6,c6,ad,b6,c6,c9,0f,d0,e5, 3882
1087 dataa3,b0,85,2d,85,2f,85,31,a9,0e,85,2e,85,30,85,32,4c,74,a4,aa, 2137
ready.

```

ag

HYPERSCREEN

HYPERSCREEN

İşte Commodore 64 için bulunan en yeni özellik. Bugüne kadar C64'ün yapamayacağı iddia edilen birçok şey, tüm bu iddialara karşılık yapıldı. Tıpkı bu programda olduğu gibi...

Bugüne kadar sprite'ların, ekranın sadece yazı yazılabilen kısımlarında görünebileceği öne sürülüyordu ve bunun aksini ispat etmek kimsenin aklına gelmiyordu. Ancak Holger Gehrmann isimli C64 meraklısı bir Alman çok kısa bir programla bunu yapmayı başardı ve bu özelliğin ilk kullanımları da kendini göstermeye başladı.

Hyperscreen çalışmaya başladığı zaman artık ekranın altında ve üstünde varolan boşlukları unutun! Bu yeni ekran kısımlarında yazı yazamamakla beraber, sprite'lar, bilinen eski kurallara uygun olarak buralarda gösterilebilmektedir. Şimdi isterseniz programımızın nasıl yazıldığını ve çalıştığını görelim...

Önce, hyperscreen adlı programı, bu sayımızda yayınladığımız Makine Dili Editörü (MDE) yardımı ile girin. Bu konu ile ilgili ayrıntılı bilgiyi MDE için verilen yazıda bulabilirsiniz. Programı diskete veya kasete kaydettikten sonra RUN STOP/RESTORE ile MDE'den çıkın. LOAD "HYPERSCREEN",8,1 veya LOAD "HYPERSCREEN" ile disketten veya kasetten

programı hafızaya yükleyin. Yükleme bittikten sonra da NEW komutunu verin.

Artık SYS 49152 komutunu verdiğinizde Hyperscreen çalışmaya başlayacaktır. Aniden alt ve üst boşlukların ortadan kalktığını göreceksiniz. (Ne yazık ki, sağ ve sol tarafların ortadan kaldırılmasının şu anda bilinen bir yöntemi yok). POKE 49209,27 verdiğinizde bu kısımlar yeniden ortaya çıkacak, POKE 49209,19 verdiğinizde ise yeniden kaybolacaklardır. LOAD ve SAVE işlemleri sırasında, kenarlar kısa bir süre için ortaya çıkarlar, ancak işlem bitince yok olurlar.

POKE 53281,sayı ile yazı ekranı ile birlikte yeni ekran kısımları da renk değiştirir. Eğer POKE 16383, sayı yaparsanız, yeni ekran kısımlarında verdiğiniz sayının ikilik sistemde oluşturduğu bit yapısına göre dikey çizgiler ortaya çıkacaktır. Bu çizgiler daima siyah olacaktır, bunu değiştirmek imkansızdır (kimbilir?).

Kimi okuyucularımızın, bu işlemin nasıl başarıldığını soracakları muhtemeldir. Bunu açıklamak için önce Interrupt tekniğini ve ekrandaki görüntünün nasıl oluştuğunu biraz açıklayalım. Bu konuları ve makine dili programcılığını bilen okurlarımız, Hyperscreen'in açıklamalı Assembler listesini okuyarak çalışma biçimini görebilirler.

Interrupt sözcüğü, İngilizce'de devam eden bir şeyi kesmek, rahatsız etmek anlamlarına gelir. Commodore 64 de çalışırken, saniyede 60 kere yaptığı işi bırakarak, belli bir programı işleme sokar.

Daha sonra da eski işine kaldığı yerden devam eder. Bu ara-programın çalışması saniyenin 1/60'undan çok daha kısa sürdüğü için, normal işlemlerde hiç dikkati çekmez. Ancak bilinçli olarak bu ara-program değiştirilirse, programcının isteği doğrultusunda ve normal programdan bağımsız çalışan ikinci bir kısa program çalıştırılabilir. Normal ara-programı gösteren vektör \$0314/\$0315'de (788/789) bulunur. Bu vektörü değiştirerek kendi ara-programımızı göstermesini sağlayabiliriz. Bu, ancak makine dilinde mümkündür.

Ekranın oluşması ise, bilgisayarın ekrana gönderdiği tarama ışınları ile olur. Tüm ekran saniyede 50 kere taranır. Herhangi bir zamanda ekranın hangi satırının tarandığı, C64'ün tüm grafik işlemlerini gerçekleştiren ünitesi VIC tarafından bilinir. Bu satırın numarası \$0012'de bulunur ve kullanıma açıktır. Ayrıca C64, istenirse belli bir satıra gelindiğinde bir Interrupt yaratabilmekte ve böylece programcının, ekranın belli

bir satırından sonra istediği bir işlemi yapmasına olanak tanımaktadır.

İşte bu iki teknik kullanılarak Hyperscreen gerçekleştirildi. VIC, alt ve üst ekran kenarını oluşturmak için iki tarama satırını gözönüne alır. Birincisinde ekran açılır, ikincisinde ise kapatılır. Aşağıdaki kapatılma satırı ile yukarıdaki açılma satırı arası alt ve üst boşlukları yaratır. Bu iki satır numarasını (ki, bunlar normal yazı satırlarından farklıdır), VIC'deki bir bit ile değiştirmek mümkündür. Bu bit'in değeri 1 ise, ekranda bilinen 25 yazı satırı görünür. Eğer bu bit 0 değerine sahipse, ekranda sadece 24 yazı satırı kalır ve ekran kenarları biraz uzar.

Şimdi... Eğer bu bit, ekranın kapatıldığı tarama satırından hemen önce satır sayısı 24 olacak şekilde değiştirilirse, VIC'in içindeki, ekranın nerede kapanacağını gösteren tarama satırının değeri de yukarıya doğru kayacaktır. Ancak bu sırada tarama işi o satırı çoktan geçtiği için ekran kapanmayacaktır. Böylece, ekranın alt ve üst satırları açık kalacaktır.

Tüm bunlar size çok karmaşık görünebilir. Ancak, Hyperscreen bu özelliğin kullanımını çok basite indirgemmiştir.

Hyperscreen'in yanı sıra size bir de kullanım sunuyoruz. Bu program ne yazık ki çok uzun. Ancak kullanılan sprite'ların çokluğu, bunu zorunlu kılıyor. Bu program da MDE ile yazılmak zorunda. Programı yazmayı bitirip yüklediğinizde, kısa bir BASIC program göreceksiniz. Ancak bu sizi aldatmasın. Görünen

BASIC satırlarının içbellekte bittiği yerden sonra, makine dilindeki bölümler başlıyor. Bu nedenle, programın üzerinde **kesinlikle** hiçbir değişiklik yapmayın. Program işlerliğini yitirir.

Kısaca programın neler yaptığını açıklayalım. Öncelikle, Hyperscreen'in kendisi programın kendi içinde var. Bu nedenle önceden Hyperscreen'i yüklemeye gerek yok. RUN komutunu verdiğinizde, program sizden şu anki saati soracaktır. Sonra da alarmin çalmaya başlaması gerektiği saati. Bu sırada veri formatı şu şekilde olacaktır: Eğer saat 16'yı 17 dakika 56 saniye geçiyorsa, vereceğiniz yazı 161756 olacaktır.

Daha sonra program sizin verdiğiniz sayıları, ana programın anlayacağı biçime çevirerek gerekli yerlere yerleştirecek ve programı başlatacaktır. Ekranın alt kısmında dijital bir saat ortaya çıktığını göreceksiniz. Bu saat yazı ekranını rahatsız etmeyecektir ve sizin yapacağınız işlemlerden bağımsız olarak çalışmaya devam edecektir. Alarm saati geldiğinde de bir dakika boyunca gong çalmaya devam edecek, sonra susacaktır.

Disket ve yazıcı işlemleri sırasında, bilgisayarın kilitlememesi için, bu işlemlerden önce (LOAD, SAVE, PRINT#, GET#, INPUT#, OPEN gibi) CTRL ve RESTORE tuşlarına birlikte basın. Kenarların kapatıldığını göreceksiniz. İşleminizi yaptıktan sonra C= tuşu ile RESTORE'a basıp kenarları yeniden ortaya çıkarın.

Eğer RUN STOP/RESTORE ile durdurur-

sanız, yeniden çalıştırmak için SYS 49756 vermeniz gerekir.

Bu programı işletirken kendi sprite'larınızı da kullanabilirsiniz. Bu durumdan saatin sprite'ları etkilenmeyeceklerdir. Tek dikkat etmeniz gereken şey \$C000-\$C207 (49152-49872) arasını kullanmanız gerektiğidir.

Programı RUN etmeden önce saatin arka plan rengini, basamakların rengini, hangi basamakların görüneceğini ve gong sesinin frekansını değiştirebilirsiniz. Aşağıda verilen kalıpları kullanmanız gerekir:

POKE 2974,renk: Arka plan rengi

POKE 2980,sayı : En soldaki basamağın rengi

POKE 2987,sayı : En sağdaki basamağın rengi

POKE 2962, PEEK (2962) AND (255-2il):
I yerine 0-7 arasında (255-2il):
bir sayı koyduğunuzda,
o basamak görünmez.

POKE 3072,INT (F/256)

POKE 3067,F-256*PEEK (3072): F ile belirtilen frekans, gong sesi olarak kullanılır.

Okuyucularımızdan yeni kullanım alanları bekliyoruz...

; HYPERSCREEN ASSEMBLER PROGRAMI

```
;
;
; -----
;
IRQLO      =      $0314      ; INTERRUPT VEKTÖRÜ 1. BYTE
IRQHI      =      $0315      ; INTERRUPT VEKTÖRÜ 2. BYTE
KONTROL    =      $D011      ; KONTROL BYTE'I
TARAMA     =      $D012      ; TARAMA SATIRI BU ADRESTE
IRR        =      $D019      ; INTERRUPT KAYNAĞI
IMR        =      $D01A      ; INTERRUPT MASKESİ
KENAR      =      $D020      ; CERÇEVE RENGİ
EKРАН      =      $D021      ; EKRAN RENGİ
CRA        =      $DC0E      ; CONTROL REGISTER A
IRQORG     =      $EA31      ; NORMALDEKİ INTERRUPT ALTPROGRAMI ADRESİ
YAZMA      =      $FFD2      ; AKÜMÜLATÖRDEKİ KODU EKRANA BASAN PROGRAM
;
*=      $C000      ; PROGRAM $C000'DAN BAŞLIYOR
;
; INTERRUPT VEKTÖRÜ DEĞİŞTİRİLİYOR
;
SEI
LDY      #<IRQ
LDX      #>IRQ
```

```

        STY  IRQLO
        STX  IRQHI
        ;
        ; VIC VE CIA'DEKİ GEREKEN YERLERİ DÜZELTME
        ;
        LDA  #$00
        STA  CRA
        LDA  #$F1
        STA  IMR
        LDA  #$F8
        STA  TARAMA
        LDA  KONTROL
        AND  #$7F
        STA  KONTROL
        ;
        ; EKRAN SİLİNMESİ VE BAŞLAMA
        ;
        LDX  #$0B          ; 11 HARF YAZILACAK
DÖNGÜ   LDA  ISIM,X
        JSR  YAZMA
        DEX
        BPL  DÖNGÜ        ; SON KARAKTER DEĞİLSE YENİDEN YAZ
        LDY  #$07          ; ÇERÇEVE RENGİ
        STY  KENAR
        LDY  #$00          ; EKRAN RENGİ
        STY  EKRAN
        RTS                ; GERİYE DÖNÜŞ
        ;
        ; İSTE GERÇEK BÖLÜM
        ;
        LDX  #$13          ; 24 SATIRA GEÇİŞ
        STX  KONTROL
        LDA  TARAMA
        BNE  BEKLE
        LDX  #$1B          ; 25 SATIRA GEÇİŞ
        STX  KONTROL
        LDA  #$01
        STA  IRR
        LDA  #$F1
        STA  IMR
        JMP  IRQORG        ; ESKİ INTERRUPT ALT-PROGRAMI
        ;
        ISIM  .TEXT "NEERCSREPYH"
        .BYTE 147

```

PROGRAM	: SAAT	0801 0FB8											
0801	: 1A 08 0A 00 97 20 35 33 58	03B9	: B0 D0 02 E6 B1 A5 B1 C9 D1										
0809	: 32 38 30 2C 30 32 3A 97 96	03C1	: C3 D0 EA A9 C6 85 AE A9 83										
0811	: 20 35 33 32 38 31 2C 32 01	03C9	: 0C 85 AF A9 00 85 B0 A9 FB										
0819	: 00 46 08 14 00 99 22 93 3D	03D1	: E0 85 B1 A0 00 B1 AE 91 60										
0821	: 11 11 1D 9E 53 49 4D 44 13	03D9	: 80 E6 AE D0 02 E6 AF E6 A6										
0829	: 49 4B 49 20 53 41 41 54 5B	03E1	: 80 D0 02 EC B1 A5 B1 C9 F3										
0831	: 3A 05 22 3B 3A 9F 31 2C 9C	03E9	: E5 D0 EA 4C 5C C2 78 A9 8C										
0839	: 30 2C 30 3A 84 31 2C 54 FE	03F1	: 7F 8D 0D DC A9 2C 8D 14 70										
0841	: 24 3A A0 31 00 76 08 1E E1	03F9	: 03 A9 C0 8D 15 03 A9 01 C5										
0849	: 00 99 C7 28 31 33 29 20 9E	0A01	: 8D 1A D0 A9 F7 8D 12 D0 DA										
0851	: 22 11 1D 9B 41 4C 41 52 D7	0A09	: A2 2F BD 00 D0 9D 5C C1 A1										
0859	: 4D 20 53 41 41 54 49 20 CF	0A11	: CA 10 F7 A9 00 8D 08 DD 5F										
0861	: 3A 05 22 3B 3A 9F 31 2C CC	0A19	: 58 60 78 AD 19 D0 8D 19 F6										
0869	: 30 2C 30 3A 84 31 2C 57 34	0A21	: D0 AD 12 D0 C9 F5 D0 2F 54										
0871	: 24 3A A0 31 00 9D 08 28 5E	0A29	: A2 18 BD 00 D0 9D 5C C1 36										
0879	: 00 48 24 B2 C8 28 54 24 64	0A31	: BD 8E C1 9D 00 D0 CA 10 2B										
0881	: 2C 32 29 3A 4D 24 B2 CA AE	0A39	: F1 A9 C4 8D 00 DD A9 00 77										
0889	: 28 54 24 2C 33 2C 32 29 1A	0A41	: 8D 12 D0 A2 13 BD 1B D0 8D										
0891	: 3A 53 24 B2 C9 28 54 24 4C	0A49	: 9D 77 C1 BD A9 C1 9D 1B 1F										
0899	: 2C 32 29 00 C7 08 32 00 AE	0A51	: D0 CA 10 F1 4C 9D C0 20 55										
08A1	: 42 B2 30 3A 8B C5 28 48 A8	0A59	: CA C1 A2 18 BD 5C C1 9D B0										
08A9	: 24 29 B1 31 32 A7 42 B2 C3	0A61	: 00 D0 CA 10 F7 A9 F5 8D 3E										
08B1	: 31 3A 48 24 B2 C9 28 C4 3A	0A69	: 12 D0 A2 13 BD 77 C1 9D C8										
08B9	: 28 C5 28 48 24 29 AB 31 74	0A71	: 1B D0 CA 10 F7 A9 97 8D EF										
08C1	: 32 29 2C 32 29 00 D9 08 E3	0A79	: 00 DD 58 4C 31 EA EA A0 5F										
08C9	: 3C 00 8B 20 48 24 B2 22 A1	0A81	: 00 AD 0B DD 29 10 18 4A DE										
08D1	: 30 30 22 A7 42 B2 31 00 15	0A89	: 4A 4A 4A 69 80 48 AD 0B CF										
08D9	: 04 09 46 00 97 20 35 36 AF	0A91	: DD C9 12 D0 09 A9 80 8D EC										
08E1	: 35 38 37 2C 31 36 AC C5 89	0A99	: F3 C7 68 4C C1 C0 C9 92 87										
08E9	: 28 C8 28 48 24 2C 31 29 43	0AA1	: D0 09 A9 81 8D F8 C7 68 21										
08F1	: 29 AA C5 28 C9 28 48 24 2D	0AA9	: 4C C1 C0 68 8D F8 C7 AD 2E										
08F9	: 2C 31 29 29 B0 42 AC 31 60	0AB1	: 0B DD 29 0F 69 80 48 AD EE										
0901	: 32 38 00 29 09 50 00 97 B7	0AB9	: 0B DD C9 12 D0 09 A9 80 64										
0909	: 20 35 36 35 38 36 2C 31 40	0AC1	: 8D F9 C7 68 4C EA C0 C9 FC										
0911	: 36 AC C5 28 C8 28 4D 24 5F	0AC9	: 92 D0 09 A9 82 8D F9 C7 47										
0919	: 2C 31 29 29 AA C5 28 C9 5A	0AD1	: 68 4C EA C0 68 20 32 C1 06										
0921	: 28 4D 24 2C 31 29 29 00 7F	0AD9	: A9 8A 8D FA C7 AD 0A DD 58										
0929	: 4E 09 5A 00 97 20 35 36 4E	0AE1	: 29 F0 4A 4A 4A 4A 69 80 FC										
0931	: 35 38 35 2C 31 36 AC C5 58	0AE9	: 8D FB C7 AD 0A DD 29 0F 6E										
0939	: 28 C8 28 53 24 2C 31 29 F5	0AF1	: 69 80 8D FC C7 A9 8A 8D AC										
0941	: 29 AA C5 28 C9 28 53 24 A9	0AF9	: FD C7 AD 09 DD 29 F0 4A E6										
0949	: 2C 31 29 29 00 7A 09 64 3E	0B01	: 4A 4A 4A 69 80 8D FE C7 30										
0951	: 00 81 20 49 B2 30 20 A4 BA	0B09	: AD 09 DD 29 0F 69 80 8D 31										
0959	: 20 33 3A 97 35 31 31 30 96	0B11	: FF C7 AD 0E DD 09 80 8D 64										
0961	: 30 AA 49 2C C5 28 CA 28 D7	0B19	: 0E DD AD 08 DD 4C 57 C1 A3										
0969	: 57 24 2C 49 AA 31 2C 31 4E	0B21	: 8D F9 C7 AD 0B DD E9 80 9B										
0971	: 29 29 AA 31 32 38 3A 82 D2	0B29	: 10 01 60 EE F8 C7 EE F9 2D										
0979	: 00 8F 09 6E 00 97 20 35 F9	0B31	: C7 EE F9 C7 AD F9 C7 E9 84										
0981	: 36 35 38 34 2C 30 3A 93 47	0B39	: 8A 10 01 60 69 7F 8D F9 D4										
0989	: 22 11 11 11 22 00 9A 09 39	0B41	: C7 EE F9 C7 60 4C DA C1 0E										
0991	: 78 00 9E 20 32 34 36 30 B3	0B49	: 32 60 00 00 00 00 00 00 AC										
0999	: 00 00 00 A9 EF 85 AE A9 08	0B51	: 00 00 00 00 00 00 00 00 52										
09A1	: 09 85 AF A9 00 85 B0 A9 D0	0B59	: 00 00 00 1B FA D1 00 00 FB										
09A9	: C0 85 B1 A0 00 B1 AE 91 18	0B61	: C8 00 15 F9 F1 00 00 00 CD										
09B1	: B0 E6 AE D0 02 E6 AF E6 7E	0B69	: 00 00 FA F2 F1 F2 F3 F4 F7										
		0B71	: F0 F1 F2 F3 F4 F5 F6 F7 60										
		0B79	: FC FF EA EA 32 04 55 04 2D										
		0B81	: 73 04 91 04 B4 04 D2 04 9A										

0B89 :	F0	04	14	04	80	17	59	9A	5C	0D61 :	3F	FF	FD	7F	FF	FE	BF	FF	06
0B91 :	36	FF	C8	00	15	79	F0	FF	DA	0D69 :	FC	C0	00	00	E0	00	00	E0	95
0B99 :	00	00	00	00	FA	0C	F1	F2	57	0D71 :	00	00	E0	00	00	E0	00	00	B1
0BA1 :	F3	F4	F0	01	01	01	01	01	89	0D79 :	EF	FF	F0	DF	FF	F8	BF	FF	67
0BA9 :	01	01	01	02	FF	07	A2	18	9E	0D81 :	FC	00	00	00	00	3F	FF	FD	73
0BB1 :	BD	00	D0	DD	8E	C1	D0	06	A4	0D89 :	1F	FF	FB	0F	FF	F7	00	00	48
0BB9 :	CA	10	F5	60	EA	EA	E0	11	C0	0D91 :	07	00	00	07	00	00	07	00	95
0BC1 :	F0	F6	9D	5C	C1	4C	66	C1	BB	0D99 :	00	07	00	00	07	00	00	03	93
0BC9 :	AD	7C	C1	8D	AE	C1	A2	01	5C	0DA1 :	3F	FF	FD	7F	FF	FE	3F	FF	44
0BD1 :	BD	9C	C7	DD	F8	C7	D0	12	BF	0DA9 :	FD	00	00	03	00	00	07	00	23
0BD9 :	BD	9E	C7	DD	FB	C7	D0	0A	ES	0DB1 :	00	07	00	00	07	00	00	07	B3
0BE1 :	4C	CB	C2	AD	88	C7	C9	0A	7B	0DB9 :	0F	FF	F7	1F	FF	FB	3F	FF	86
0BE9 :	10	03	4C	81	EA	AD	87	C7	88	0DC1 :	FD	00	00	00	00	80	00	01	C5
0BF1 :	C9	D3	F0	2D	A9	0F	8D	18	FF	0DC9 :	C0	00	03	E0	00	07	E0	00	22
0BF9 :	D4	A9	00	8D	00	D4	A9	0C	B9	0DD1 :	07	E0	00	07	E0	00	07	E0	15
0C01 :	8D	01	D4	A9	0C	8D	05	D4	64	0DD9 :	00	07	E0	00	07	C0	00	03	11
0C09 :	A9	00	8D	06	D4	A9	11	8D	D0	0DE1 :	BF	FF	FD	7F	FF	FE	3F	FF	04
0C11 :	04	D4	A9	D3	8D	87	C7	A9	EC	0DE9 :	FD	00	00	03	00	00	07	00	63
0C19 :	00	8D	7E	C7	A9	28	8D	7F	89	0DF1 :	00	07	00	00	07	00	00	07	F3
0C21 :	C7	A9	00	8D	88	C7	EE	89	04	0DF9 :	00	00	07	00	00	03	00	00	D3
0C29 :	C7	AD	89	C7	C9	05	D0	17	58	0E01 :	01	00	00	00	00	BF	FF	FC	FA
0C31 :	A9	00	8D	04	D4	A9	11	8D	B8	0E09 :	DF	FF	F8	EF	FF	F0	E0	00	2F
0C39 :	04	D4	A9	00	8D	89	C7	AD	B1	0E11 :	00	E0	00	00	E0	00	00	E0	51
0C41 :	AF	C1	49	08	8D	AF	C1	4C	1A	0E19 :	00	00	E0	00	00	C0	00	00	59
0C49 :	81	EA	20	00	C0	A9	88	8D	DE	0E21 :	BF	FF	FC	7F	FF	FE	3F	FF	04
0C51 :	18	03	A9	C2	8D	19	03	A9	AE	0E29 :	FD	00	00	03	00	00	07	00	A3
0C59 :	00	8D	FF	3F	8D	FF	FF	8D	FB	0E31 :	00	07	00	00	07	00	00	07	33
0C61 :	87	C7	8D	88	C7	4C	85	C2	BB	0E39 :	0F	FF	F7	1F	FF	FB	3F	FF	06
0C69 :	4B	52	4F	57	20	4B	43	4F	A4	0E41 :	FD	00	00	00	00	BF	FF	FC	36
0C71 :	4C	43	0D	4C	83	A4	08	48	3A	0E49 :	DF	FF	F8	EF	FF	F0	E0	00	6F
0C79 :	8A	48	98	48	AD	8D	02	C9	39	0E51 :	00	E0	00	00	E0	00	00	E0	91
0C81 :	02	D0	0D	A9	2C	8D	14	03	E9	0E59 :	00	00	E0	00	00	C0	00	00	98
0C89 :	A9	C0	8D	15	03	4C	B0	C2	73	0E61 :	BF	FF	FC	7F	FF	FE	BF	FF	46
0C91 :	C9	04	D0	0A	A9	B9	8D	14	98	0E69 :	FD	C0	00	03	E0	00	07	E0	13
0C99 :	03	A9	C2	8D	15	03	68	A8	2F	0E71 :	00	07	E0	00	07	E0	00	07	B2
0CA1 :	68	AA	68	28	4C	47	FE	78	69	0E79 :	EF	FF	F7	DF	FF	FB	BF	FF	40
0CA9 :	AD	19	D0	8D	19	D0	AD	12	BC	0E81 :	FD	00	00	00	00	3F	FF	FD	74
0CB1 :	D0	F0	03	4C	31	EA	4C	81	E2	0E89 :	1F	FF	FB	0F	FF	F7	00	00	48
0CB9 :	EA	CA	10	06	EE	88	C7	4C	B8	0E91 :	07	00	00	07	00	00	07	00	95
0CC1 :	F5	C1	4C	E2	C1	BF	FF	FD	1C	0E99 :	00	07	00	00	07	00	00	03	93
0CC9 :	DF	FF	FB	EF	FF	F7	E0	00	E8	0EA1 :	00	00	01	00	00	00	00	00	E2
0CD1 :	07	E0	00	07	E0	00	07	E0	15	0EAB :	01	00	00	03	00	00	07	00	27
0CD9 :	00	07	E0	00	07	C0	00	03	11	0EB1 :	00	07	00	00	07	00	00	07	B3
0CE1 :	80	00	01	00	00	00	80	00	A4	0EB9 :	00	00	07	00	00	03	00	00	93
0CE9 :	01	C0	00	03	E0	00	07	E0	97	0EC1 :	01	00	00	00	00	BF	FF	FD	BC
0CF1 :	00	07	E0	00	07	E0	00	07	32	0EC9 :	DF	FF	FB	EF	FF	F7	E0	00	E8
0CF9 :	EF	FF	F7	DF	FF	FB	BF	FF	C0	0ED1 :	07	E0	00	07	E0	00	07	E0	15
0D01 :	FD	00	00	00	00	00	00	01	01	0ED9 :	00	07	E0	00	07	C0	00	03	11
0D09 :	00	00	03	00	00	07	00	00	02	0EE1 :	BF	FF	FD	7F	FF	FE	BF	FF	06
0D11 :	07	00	00	07	00	00	07	00	15	0EE9 :	FD	C0	00	03	E0	00	07	E0	93
0D19 :	00	07	00	00	07	00	00	03	13	0EF1 :	00	07	E0	00	07	E0	00	07	32
0D21 :	00	00	01	00	00	00	00	00	62	0EF9 :	EF	FF	F7	DF	FF	FB	BF	FF	C0
0D29 :	01	00	00	03	00	00	07	00	A7	0F01 :	FD	00	00	00	00	BF	FF	FD	F8
0D31 :	00	07	00	00	07	00	00	07	33	0F09 :	DF	FF	FB	EF	FF	F7	E0	00	28
0D39 :	00	00	07	00	00	03	00	00	13	0F11 :	07	E0	00	07	E0	00	07	E0	55
0D41 :	01	00	00	00	00	3F	FF	FD	38	0F19 :	00	07	E0	00	07	C0	00	03	51
0D49 :	1F	FF	FB	0F	FF	F7	00	00	08	0F21 :	BF	FF	FD	7F	FF	FE	3F	FF	44
0D51 :	07	00	00	07	00	00	07	00	55	0F29 :	FD	00	00	03	00	00	07	00	A3
0D59 :	00	07	00	00	07	00	00	03	53	0F31 :	00	07	00	00	07	00	00	07	33

```

0F39 : 0F FF F7 1F FF FB 3F FF 06
0F41 : FD 00 00 00 00 00 00 00 3F
0F49 : 00 00 00 00 00 00 00 00 4A
0F51 : 00 00 00 00 00 00 00 00 52
0F59 : 3C 00 00 3C 00 00 3C 00 0E
0F61 : 00 00 00 00 00 00 00 3C DA
0F69 : 00 00 3C 00 00 3C 00 00 5A
0F71 : 00 00 00 00 00 00 00 00 72
0F79 : 00 00 00 00 00 00 00 00 7A
0F81 : 00 00 00 00 00 00 54 80 D4
0F89 : 06 FA 9F 00 00 57 80 06 BD
0F91 : F4 9F 00 00 48 80 02 F2 CB
0F99 : 9F 00 00 4D 80 02 F0 9F FD
0FA1 : 00 00 53 80 02 EE 9F 00 9C
0FA9 : 00 42 00 00 00 00 00 00 CB
0FB1 : 49 00 83 00 00 00 00 35 46

```

PROGRAM : HYPERSCREEN C000 C060

```

C000 : 78 A0 38 A2 C0 8C 14 03 F1
C008 : 8E 15 03 A9 00 8D 0E DC 75
C010 : A9 F1 8D 1A D0 A9 F8 8D B2
C018 : 12 D0 AD 11 D0 29 7F 8D 8F
C020 : 11 D0 A2 0B BD 54 C0 20 65
C028 : D2 FF CA 10 F7 A0 07 8C 68
C030 : 20 D0 A0 00 8C 21 D0 60 B6
C038 : A2 13 8E 11 D0 AD 12 D0 8E
C040 : D0 FB A2 1B 8E 11 D0 A9 22
C048 : 01 8D 19 D0 A9 F1 8D 1A 05
C050 : D0 4C 31 EA 4E 45 45 52 B9
C058 : 43 53 52 45 50 59 48 93 9A

```

hs-demo , 1

```

10 sys49152:poke49209,27
20 gosub30:goto50
30 print"      hyperscreen      ":return
40 print"          dugmeye basin !"
45 poke198,0:wait198,1:poke198,0:return
50 print"su Komut ile :sys49152"
60 print"hyperscreen calistirilir."
70 forw=1to2000:nextw:poke49209,19:gosub40
80 gosub30
90 print"goruldugu gibi, bu emir ile, ekranin"
100 print"ust ve alt kenarlari ortadan kalkar."
120 print"ekran kenarini acmak icin,"
130 print"poke 49209,27"
140 print"ve kapamak icin de"
150 print"poke49209,19"
160 print"gerikir.":forx=1to5:poke49209,27:forw=1to650:nextw
170 poke49209,19:forw=1to650:nextw,x:gosub40
180 gosub30
190 print"tabii ki bu yeni ekran bolumunde sadece"
200 print"sprite'lar gosterilebilir; normal"
210 print"karakterler degil."
220 poke53269,1
230 poke2040,0
240 poke53248,100:fory=1to3:forx=0to255:poke53249,x:nextx,y
250 gosub40
260 poke53269,0
270 gosub30
290 print"16383"
300 print"numarali hafizaya bir sayi poke"
310 print"edildiginde bu sayi kenar ekranda"
315 print"dik çizgiler olarak görünür."
320 fory=1to3:forx=0to255:poke16383,x:nextx,y:poke16383,0:gosub40
330 gosub30

```


Commodore 64'te yazı karakteri tanımlama

Baytan BİTİRMEZ

Diyelim ki Commodore-64 Türkçeyazar sisteminiz yok (niye yok, o da ayrı bir soru) ve Commodore'unuza **Türkçe** değil **Türkçe** (veya Arapça veya Çince) yazdırmak istiyorsunuz. Yahut Commodore'un standart yazı karakterinden sıkıldınız, farklı bir yazı türünü denemeyi düşünüyorsunuz. İşte size göre bir program. Program, kısmen BASIC, kısmen de makina dilinde. Tamamını BASIC yapmak da mümkün; ama o zaman yaklaşık 500 kat daha yavaş işlediğinden, kısa zamanda sabır taşınızı çatlatması ihtimal dahilinde.

COMMODORE 64'ÜN GÖRÜNTÜ SİSTEMİ

Commodore 64'te tüm ekran kontrol işlemlerinin VIC (Video Interface Chip) isimli bir alet tarafından yapıldığını belki biliyorsunuz. VIC'in asıl görevini, ekranı oluşturan 200*320 ışıklı noktayı yakıp söndürmek diye tanımlayabiliriz. Ekran-da gördüğümüz her harf de, VIC'in gözünde, 8*8 noktadan oluşan soyut bir şekil.

Commodore'un tuşlarından birine bastığımızda, bilgisayarın kendisinin yaptığı iş, basılan tuşun POKE kodu olan rakamı, 1024- 2023 adresleri arasındaki "ekran belleğinde" uygun noktaya yerleştirmekten ibaret. Bu noktada devreye VIC giriyor ve ekran belleğindeki kod numarasına karşılık düşen bazısı yanık, bazısı sönmüş 64 noktalık bir diziyi ekrana gönderiyor. 64 noktalık bu dizileri de, ROM'dan her biri 8'er bitlik 8'er byte, yani 0-255 arası birer rakam olarak okuyor.

64 noktalı dizilerin rakam olarak nasıl tanımlandığını iyice anlamamız şart. Örnek olarak A harfini verelim:

Büyük A harfinin bellekteki tanımı, o halde, 24, 60, 102, 126, 102, 102, 102, 0. Şimdi, ROM'daki bu standart tanımın yerine kendi tanımımızı koymak için yapacağımız, önce RAM belleğin bir böl-

gesinde kendi rakam dizilerimizi oluşturmak, sonra da VIC'i kandırıp, ekrana göndereceği noktaları ROM'dan değil, bizim seçtiğimiz RAM bölgesinden okumasını sağlamak.

Bu işin bir sürü inceliği var. Bunları burada ayrıntılarıyla anlatmaktansa, ilgilenenlerin Reference Guide'a başvuru-

	7	6	5	4	3	2	1	0
0								
1								
2								
3								
4								
5								
6								
7								

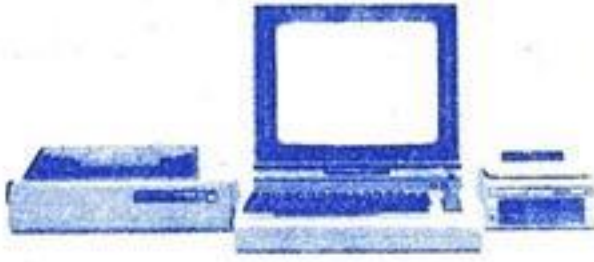
İkili	Onaltılı	Onlu
00011000	18	24
00111100	3C	60
01100110	66	102
01111110	7E	126
01100110	66	102
01100110	66	102
01100110	66	102
00000000	00	0

masını söylemekle yetineceğiz. Şu kadarını belirtmek yeter:

1. Karakter tanımlarını içeren ROM bellek, normal olarak bilgisayara gözükmez; sadece VIC tarafından görülür. Bu ROM'daki bilgileri RAM'a taşımak için, önce 01 adresindeki 2 numaralı bit'i sıfırlayıp, karakter ROM'unu bilgisayarcaya görülebilir hale getirmemiz gerekir.

2. Kendi karakter tanımlarımızı koymak için RAM'daki en uygun bölge olarak, serbest belleğin sonuna yakın bir yer, yani 32768-34815 (onaltılı 8000-87FF) bölgesini seçeceğiz. Önce ROM'daki bilgileri buraya taşıyıp, sonra bunları keyfimizce değiştirebileceğiz.

3. VIC'in ulaşabileceği toplam bellek alanı sadece 16 Kbyte olduğundan, sistem ekran belleğini de 1024-2023'teki normal yerinden, yukarıdakine yakın bir yere, örneğin 34816-35815'e (onaltılı 8800-8BE7) taşımamız gerekecek. Bu taşıma işlemi, VIC'i kontrol eden 53272 no.lu bellek adresine bir-iki POKE'la halolacak.



PROGRAMIN İŞLEYİŞİ

Program RUN edildiğinde, ilk önce sırasıyla iki karakter vermenizi isteyerek, bunların büyütülmüş şekillerini ekrandaki kutulara getirecek. Bu noktada, sol kutudaki birinci karakteri yeniden çizip değiştirmeniz mümkün.

Değiştirme sırasında, + tuşunu bir nokta eklemek, aralık çubuğu veya [DEL] tuşunu bir nokta silmek, [CRSR] tuşlarını da yer değiştirmek için kullanabilirsiniz.

↑ tuşu, ikinci kutudaki şekli birinci kutuya taşıyacaktır. Bu işlem, örneğin c yerine ç şeklini oluştururken kolaylık sağlamak için düşünülmüştür.

Çizim sırasında beğenmediğiniz bir şekli ilk durumuna geri döndürmek için [CLR/HOME], yeni bir karakteri seçmek için ise [SHIFT] [CLR/HOME] tuşlarını kullanabilirsiniz.

[RETURN] tuşu, sol kutuda o anda bulunan şekli, sol kutunun yanında görülen karakterin yerine belleğe geçirecektir.

Yarattığınız yeni karakter setlerini saklamak veya kasetten tekrar yüklemek için s ve l tuşlarını kullanın. Tabii disketten çalışıyorsanız, 270 ve 300'üncü satırlardaki OPEN 2,1,2'yi OPEN 2,8,2 olarak değiştirmeniz gerekiyor.

q tuşu ile programdan çıkmak istediğinizde, normal karakter setine dönmek isteyip istemediğiniz sorulacaktır. Ceva-

biniz hayır ise, programı silseniz bile, yeni karakter setiniz geçerliliğini koruyacak, başka programlarda da kullanılabilecektir.

PROGRAMIN AÇIKLAMASI

Basic Bölümü

Satır 10; 800-814 satırlarda DATA olarak verilen üç makine dili alt-programı, 49152 (\$C000) adresinden başlayarak belleğe POKE ET.

20: SYS komutu, makine dilindeki alt-programlar için GOSUB komutunun eşidir. 49152'deki ilk alt-programın açıklaması için aşağıya bakın. Ondan sonraki iki POKE, ekran rengini bej, yazı rengini siyah yapıyor.

30-50: Ekranı sil, 8*8 iki kutu çiz.

60-70: IN değişkeni bir 'flag' olarak kullanılmıştır. IN=0 ise, yeniden tanımlama için iki yeni karakter istenmekte, IN=1 ise doğrudan doğruya menüye geçilmektedir.

80-99: Bir döngü içinde, c\$(1) ve c\$(2) adında iki karakter alınıyor. ASCII kodları 0-31 ve 128-160 arasında olan karakterler, RETURN, DELETE gibi kontrol karakterleri olduğundan kaale alınmıyor.

95-96: Basılan karakterin POKE kodunu buluyor. CHR\$(147), 'CURSOR geri' karakteri. Sonra ç önemli bellek adresi: 648, ekranın bellekteki 256 byte'lık sayfaların kaçıncısından başladığını (normali 4, şimdi ise 136), 214, cursor'un bulunduğu satırı, 211 ise cursor'un bulunduğu sütunu içeriyor. Her sayfa 256, her satır da 40 byte olduğundan, cursor'un yeri = sayfa * 256 + satır * 40 + sütun. Bu adrese PEEK ettiğimizde, son bastığımız karakterin POKE değerini buluyoruz.

100-105: Bulunan POKE değerlerini 253 (\$FD) adresinde saklayıp, 49269 (\$C075)'teki, karakterlerin büyütülmüş şekillerini çizen ikinci makine dili alt-programa git; karakterlerin normal boy şekillerini kutuların sol yanına POKE et.

49279'a (\$C07F) POKE'lar, doğrudan makine dili programın içine müdahale ederek, büyütülmüş şekillerin ekranda nereye çıkacağını belirliyor (ikinci si ilkinden 13 hane sonra).

Burada POKE'un üç ayrı kullanımını görüyoruz. 253 adresi, bellekte boş bir hane. Buraya, makine dili programın kullanması için bir veri yerleştiriyoruz. 49152'den başlayan 257 byte'ı makine dili programımız için kullandık. Buraya bir POKE, doğrudan doğruya m.d. programı değiştirmek demek. 34816'dan başlayan 1000 byte ise ekran belleği; buraya POKE edilenler doğrudan ekrana çıkıyor.

110-180: Menü. İlk başta [HOME] ve 10 tane CURSOR aşağı var.

182-188 biraz karışık. GET komutuyla bir tuşa basılmasını beklerken normal

cursor kaybolduğundan, burada sah-te bir cursor yaratıyoruz.

Önce pozisyon: [HOME], bir aşağı, üç sola.

185'te, 96'dakine benzer bir yöntemle bu pozisyonun adresini hesaplayıp CC değişkenine veriyoruz: 209-210'da, bulunulan satırın başının adresi, 211'de ise cursor sütunu var. Sonra CC'de o anda bulunan karakteri PP'de saklayıp, yerine 42 numaralı karakter olan yıldız koyuyoruz. Herhangi bir tuşa basıldığında, yıldızın yerine tekrar eskiden o hanede bulunan PP karakteri geliyor.

190-220: Basılan tuş CURSOR aşağı, yukarı, geri, ileri veya space ise aynen bas. [DEL] (karakter no. 20) ise, bir geri, space, bir geri bas. + ise, [RVS ON], space, [RVS OFF], yani siyah bir kutu bas.

230: Basılan CHR\$(13), yani [RETURN] ise, ilk verdiğimiz karakterin POKE kodunu 253'te sakla, sol kutudaki büyütülmüş şekli yeni karakter tanımı olarak belleğe geçiren 49330'daki (\$C0B2) üçüncü makine dili programa git, yeni baştan iki karakter almak üzere başa dön.

240-250: T'ye basılmışsa, [HOME], onüç kez CURSOR ileri, onüç kez [DELETE] ve bir CURSOR aşağı bas. Bu işlemi 10 kez tekrarla. Böylece ilk on satırı onüçer hane sola kaydırıp, ikinci kutuyu birincisinin yerine taşı.

260-265: [HOME]'a basılmışsa (karakter 19) verilen karakterleri saklı tutarak, [CLR]'a basılmışsa (karakter 147) yeni iki karakter almak üzere başa dön.

270-320: SAVE ve LOAD. OPEN komutundaki ikinci rakam, teyp için 1, disk için 8 olacaktır. Yapılan 32768-34815 (\$8000-87FF) bölgesindeki karakter tanımlarını sırasal bir dosyaya saklamak veya sırasal dosyadan okumaktır.

330: Basılan tuş, buraya kadar sayılan seçeneklerden hiçbirisi değil ve Q da değilse, 185'e dönüp tekrar başka bir tuşa basılmasını bekle. Eğer Q ise, evet/hayır sorusunu sor, cevap evetse 49152'deki ilk m.d. programa gidip, normal karakter setini tekrar ROM'dan okut.



MAKİNE DİLİ BÖLÜMLER

Birinci alt-program: C000-C052

Bu rutin, ROM bellekteki karakter tanımlarını RAM'a taşıyor ve VIC'e, görüntü için gerekli bilgileri ROM'dan değil RAM'dan almasını bildiriyor.

Birkaç önemli sistem adresi:

\$34 ve \$38: BASIC program metni ile değişkenlerinin üst sınırı. Buraya 80 (yani 8000) koyarak BASIC serbest belleğinden 8 Kilobyte eksiltiyoruz.

\$01, ikinci bit: Character ROM Enable/Disable. Bu bit 1 iken bilgisayar D800-DFFF bölgesinde renk belleği ile Input/Output sicillerini, sıfırken karakter ROM'unu görür.

Input/Output sicillerini devreden çıkarmadan önce, bilgisayara SEI komutuyla "input/output'a boş ver" demek şart. Yoksa makine saçmalar. CLI komutu ise, "tekrar input/output'larla ilgilen" demek.

\$DD02: Data Port A yön sicili. Son iki bit'e 1 yazılırsa \$DD00'daki Data Port'una yazmak, 0 yazılırsa Data Port'u okumak mümkün olur.

\$DD00, 0 ve 1'inci bit: Buraya verilen değer, VIC'in 16'şar Kilobyte'lık 4 bellek 'bankasından' hangisini göreceğini belirler. 01, \$8000-BFFF arasındaki üçüncü bankayı seçer.

\$DO18: VIC kontrol sicili. Sol 4 bit ekran belleğinin, sağ 4 bit karakter tanımlarının VIC'in gördüğü bellek bankası içindeki konumunu belirler. Buraya yazılan 20 değeri, ekranı \$8000+0800'e, karakter tanımlarını da \$8000+0000'a yerleştirir.

\$0288: Bilgisayara ekran belleğinin ilk sayfasını bildirir. \$88, yani \$8800, yani 34816.

\$D800-DFFF arasındaki ROM'da, POKE kodlarındaki sırayla, 128 normal ve 128 reverse karakter için, 8'er byte'dan 8*256 byte tanım bulunur.

Karakter tanımlarını \$D800'den \$8000'e taşıma işlemi COOF-CO24 arasında gerçekleşiyor. Dikkat edilirse, ilk sayfa (\$0100 yani 256 parça) taşındık-

tan sonra, program kendi kendini değiştirerek \$C010 ve \$C013'teki adresleri sırasıyla D900 ve 8100, DA00 ve 8200 vb. yapıyor; en sonda da (\$C048-C051) programa tekrar başlangıç değerlerini veriyor.

İkinci alt-program

(\$C075-C0B1):

Hem bu, hem üçüncü program tarafından kullanılan \$C061 alt-rutini, BASIC ana program tarafından \$FD (253) adresine konulan karakter kodunu alıp, karakteri tanımlayan 8 byte'lık dizinin başladığı yeri hesaplıyor.

Başlangıç adresi = \$8000 + n * 8

Adresi oluşturan iki byte'ı hesaplamak için, n * 8 256'ya bölünür; çıkan tamsayı + \$80 büyük haneyi, kalan + 00 da küçük haneyi oluşturur.

Örneğin karakter kodumuz 5 (küçük e harfi) olsun.

4 * 8 = 40\$28
80 00
+ 00 28

ADRES = 80 28 (\$8028 = 32808)

Onatılı aritmetikte bu işlemi şöyle yapıyoruz:

LSR işlemi, bir byte'ın tüm bitlerini birer sağa kaydırır; böylece o sayıyı ikiye böler. *8/256 için beş LSR gerekir.

5 = 00000101 = \$05
LSR 00000010 = \$02
LSR 00000001 = \$01
LSR 00000000 = \$00
LSR 00000000 = \$00
LSR 00000000 = \$00

Sonuç \$00, \$80 + 00 = 80.

ASL işlemi, bir byte'ın tüm bitlerini birer sola kaydırır; böylece o sayıyı ikiye çarpar. Üç ASL = *8.

00000101 = \$05 = 5
ASL 00001010 = \$0A = 10
ASL 00010100 = \$14 = 20
ASL 00101000 = \$28 = 40

Doğrudan doğruya onaltılı sistemde düşünmenin kolaylıklarını görmek için şimdi de büyük E (kodu 69) harfini alalım.

69 = 01000101 = \$45 = 69
LSR 00100010 = \$22 = 34
LSR 00010001 = \$11 = 17
LSR 00001000 = \$08 = 8
LSR 00000100 = \$04 = 4
LSR 00000010 = \$02 = 2

Sonuç \$02, \$80 + 02 = \$82

Üç ASL işleminin sonucu: \$28 (40)

Büyük E'nin adresi: \$8228 (33320)

C061 alt-rutini, \$FD'deki karakterin tanımının bulunduğu adresin küçük hanesini \$FB, büyük hanesini de \$FC'ye koyup dönüyor.

\$C078-C0A5'da olanlar şunlar:

\$C07A: \$FE adresine \$80, yani ikili 1000 0000 rakamını koy. Bu rakamı 8 kez LSR'ledikçe, sırasıyla ikinci, üçüncü... biti 1 olan birer rakam elde edeceğiz.

\$C07E: X = 13 veya X = 0. BASIC programın 100-105 satırlarına bakın.

\$C080: \$FB-FC'de gösterilen adresten Y ötedeki byte'ı al. Y'nin değeri 0'dan 7'ye kadar sırayla artacaktır.

\$C082: Esas işlem: Aldığın byte'ı, \$FE'de o an bulunan rakamla AND'le. AND işlemi, birbirine karşılık düşen iki bit de 1 ise 1, yoksa 0 verir.

01000101
AND 10000000

00000000

01000101
AND 01000000

01000000

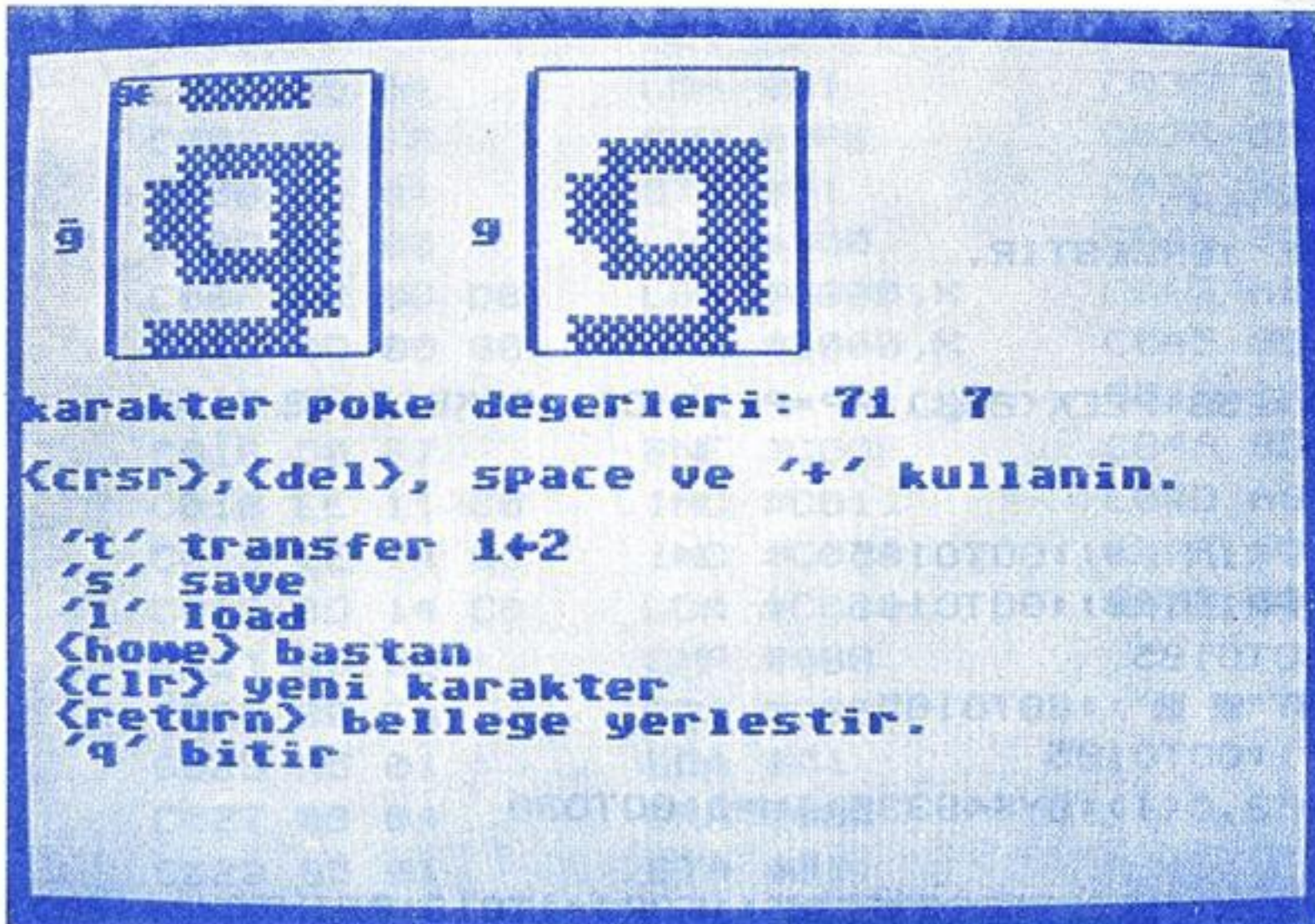
\$FE'deki tek biti 1 olan rakamı teker teker sağa kaydırıp, karakteri tanımlayan byte ile AND'layarak, sırasıyla tüm bitlerin 1 mi, 0 mı olduğunu anlıyoruz.

\$C084: Sonuç sıfırsa, space (kodu \$20 = 32)

\$C086: Sonuç 0 değilse, bir grafik şekil (kodu \$66 = 102) al.

\$C08C: \$882B'den X öteye (ekranın ikinci satırının 4'üncü hanesinden 0 veya 13 öteye) koy.

\$C08F: X'i bir artır; \$FE'deki rakamı sağa kaydır. Rakam henüz 8 kez sağa kayıp da sıfırlanmamışsa



SC080'e dönüp, aynı byte'ın bir sonraki bitini dene.
 SC094: \$882B'ye \$28 ekle, yani ekranda bir satır (40 hane) ileri.
 SC0A2: Tüm işlemi 8 kez tekrarla.
 SC0A7: Ekrandaki başlangıç konumunu ilk baştaki duruma getir.

Üçüncü alt-program (\$C0B2)

Ekrandaki kutuda bizim oluşturduğumuz şekli tarayıp 8'er bitli 8 byte oluşturan bu rutin, bir-iki küçük farkla, bir örneğinin tam tersi.

SC0B5: SFB-FC'ya konan karakter tanım adresinin \$0400 fazlasını SF9-FA'ya koy. Bu adrese, yeni tanımladığımız karakterin 'reverse'ini (negatifini) koyacağız.
 SC0BF: SFF adresini sıfırla. Burada, ekrandaki şekilden bit-bit yeni karakter tanımımızı hesaplayacağız.
 SC0C3: Yukarıdakinin aksine, bu kez hesabımızı sol uçtan değil, sağ uçtaki bittten başlayarak yapıyoruz. Keyif meselesi.
 SC0C9: Ekrandaki kutunun başından 7 sonraki haneyi oku.

SC0CC: Boşsa geç.
 SC0D0: Değilse, şu anda kaçınca bitteysen, SFF'teki sayının o bitini 1 yap.
 SC0D6: X'i azalt; SFE'deki tek bitli rakamı bir sola kaydır. İşlemi 8 kez tekrarla.
 SC0DB: SFF'te oluşmuş olan rakamı SFB-FC'de yazılı karakter adresine, tersini de SF9-FA'daki reverse karakter adresine koy.
 SC0E3: Alt satıra geç; 8 kez yukarıdaki işlemleri tekrarla.

```

10 FOR N=0 TO 256:READ A:POKE 49152+N,A:KNTROL=KNTROL+A:NEXT
15 IFKNTROL<>32248THEN PRINT"DATA YANLIŞ GIRILMIŞ!":END
20 SYS 49152:POKE53281,12:POKE646,0
30 PRINT"  "
40 FORI=1TO8:PRINT"  |  |  |  |  |  |  |  | ":NEXT
50 PRINT"  "
60 IFIN=1THEN100
70 IN=1:PRINT:PRINT
80 FORI=1 TO 2:PRINT:PRINT" KARAKTER";I;" ";
85 GETC$(I):IFC$(I)=""THEN85
90 N=ASC(C$(I)):IF N<32 OR N>127 AND N<161 THEN85
95 PRINT CHR$(N)CHR$(157);
96 C(I)=PEEK(PEEK(648)*256+40*PEEK(214)+PEEK(211))
99 NEXT
100 POKE253,C(1):POKE49279,0:SYS49269:POKE35017,C(1)
105 POKE253,C(2):POKE49279,13:SYS49269:POKE35030,C(2)
110 PRINT"KARAKTER POKE DEĞERLERİ:";C(1);C(2)
115 PRINT:PRINT"<CRSR>,<DEL>, SPACE VE '+' KULLANIN.
120 PRINT"
130 PRINT" 'T' TRANSFER 1+2
140 PRINT" 'S' SAVE
150 PRINT" 'L' LOAD
160 PRINT" <HOME> BASTAN
170 PRINT" <CLR> YENİ KARAKTER
175 PRINT" <RETURN> BELLEGE YERLESTİR.
180 PRINT" 'Q' BITİR
182 PRINT" ";
185 CC=PEEK(211)+PEEK(210)*256+PEEK(209):PP=PEEK(CC):POKECC,42
187 GET A$:IFA$=""THEN 187
188 POKECC,PP
190 IFA$=" " ORA$="□"THEN PRINTA$;:GOTO185
200 IFA$="■" ORA$="■"THEN PRINTA$;:GOTO185
210 IFA$=" " THENPRINTA$;:GOTO185
215 IFA$=CHR$(20)THEN PRINT"■ ■";:GOTO185
220 IFA$="+"THENPRINT"■ ■";:GOTO185
230 IFA$=CHR$(13)THENPOKE253,C(1):SYS49330:IN=0:GOTO30
240 IFA$<>"T"THEN260
245 PRINT" ";:FORI=1TO10:PRINT" ";:FORJ=1TO13:PRINTCHR$(20);

```

```

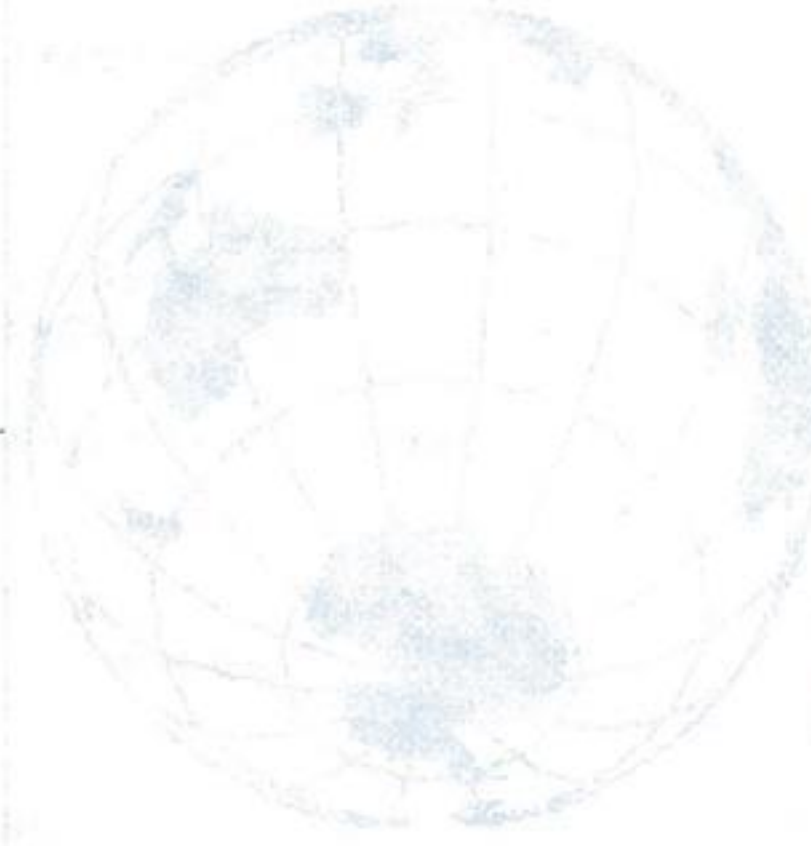
250 NEXTJ:PRINT"J";:NEXTI:GOTO182
260 IFA$=CHR$(19)THEN30
265 IFA$=CHR$(147)THEN IN=0:GOTO30
270 IFA$<>"S"THEN 300
280 INPUT"NORMAL SET ADI";L$:OPEN2,8,2,L$+"S,W"
290 FORP=32768TO34815:PRINT#2,CHR$(PEEK(P));:NEXT:CLOSE2:GOTO30
300 IFA$<>"L"THEN330
310 INPUT"NORMAL SET ADI";L$:OPEN2,8,2,L$+"S,R"
320 FORP=32768TO34815:GET#2,A$:POKEP,ASC(A$+CHR$(0)):NEXT:CLOSE2:GOTO30
330 IFA$<>"Q"THENGOTO185
340 INPUT"NORMAL KARAKTER SETI MI(E/H)";B$
350 IFB$="E"THENSYS49152

800 DATA 169,128,133,52,133,56,120,165,1,41,251,133,1,162,0,189,0
801 DATA 216,157,0,128,232,208,247,238,17,192,238,20,192,173,20
802 DATA 192,201,136,144,234,165,1,9,4,133,1,88,173,2,221,9,3,141
803 DATA 2,221,173,0,221,41,252,9,1,141,0,221,169,136,141,136,2,169
804 DATA 32,141,24,208,169,216,141,17,192,169,128,141,20,192,96,0,0,0
805 DATA 0,0,0,0,0,0,0,0,0,0,0,165,253,74,74,74,74,74,24,105,128,133
806 DATA 252,165,253,10,10,10,133,251,96,32,97,192,160,0,169,128,133
807 DATA 254,162,0,177,251,37,254,240,4,169,102,208,2,169,32,157,43
808 DATA 136,232,70,254,208,236,24,173,141,192,105,40,141,141,192,144
809 DATA 3,238,142,192,200,192,8,144,211,169,43,141,141,192,169,136
810 DATA 141,142,192,96,32,97,192,133,249,165,252,9,4,133,250,160
811 DATA 0,169,0,133,255,169,1,133,254,162,7,189,43,136,201,32,240,6
812 DATA 165,255,5,254,133,255,6,254,202,16,238,165,255,145,251,73
813 DATA 255,145,249,173,202,192,24,105,40,141,202,192,144,3,238,203
814 DATA 192,200,192,8,144,201,169,43,141,202,192,169,136,141,203,192,96
READY.

```

C000 A9 80	LDA #\$80	C02C AD 02 DD	LDA \$DD02
C002 85 34	STA \$34	C02F 09 03	ORA #\$03
C004 85 38	STA \$38	C031 8D 02 DD	STA \$DD02
C006 78	SEI	C034 AD 00 DD	LDA \$DD00
C007 A5 01	LDA \$01	C037 29 FC	AND #\$FC
C009 29 FB	AND #\$FB	C039 09 01	ORA #\$01
C00B 85 01	STA \$01	C03B 8D 00 DD	STA \$DD00
C00D A2 00	LDX #\$00	C03E A9 88	LDA #\$88
C00F BD 00 D8	LDA \$D800,X	C040 8D 88 02	STA \$0288
C012 9D 00 80	STA \$8000,X	C043 A9 20	LDA #\$20
C015 E8	INX	C045 8D 18 D0	STA \$D018
C016 D0 F7	BNE \$C00F	C048 A9 D8	LDA #\$D8
C018 EE 11 C0	INC \$C011	C04A 8D 11 C0	STA \$C011
C01B EE 14 C0	INC \$C014	C04D A9 80	LDA #\$80
C01E AD 14 C0	LDA \$C014	C04F 8D 14 C0	STA \$C014
C021 C9 88	CMP #\$88	C052 60	RTS
C023 90 EA	BCC \$C00F		
C025 A5 01	LDA \$01		
C027 09 04	ORA #\$04		
C029 85 01	STA \$01		
C02B 58	CLI		

C061 A5 FD	LDA \$FD	C0B2 20 61 C0	JSR \$C061
C063 4A	LSR	C0B5 85 F9	STA \$F9
C064 4A	LSR	C0B7 A5 FC	LDA \$FC
C065 4A	LSR	C0B9 09 04	ORA #\$04
C066 4A	LSR	C0BB 85 FA	STA \$FA
C067 4A	LSR	C0BD A0 00	LDY #\$00
C068 18	CLC	C0BF A9 00	LDA #\$00
C069 69 80	ADC #\$80	C0C1 85 FF	STA \$FF
C06B 85 FC	STA \$FC	C0C3 A9 01	LDA #\$01
C06D A5 FD	LDA \$FD	C0C5 85 FE	STA \$FE
C06F 0A	ASL	C0C7 A2 07	LDX #\$07
C070 0A	ASL	C0C9 BD 2B 88	LDA \$882B,X
C071 0A	ASL	C0CC C9 20	CMP #\$20
C072 85 FB	STA \$FB	C0CE F0 06	BEQ \$C0D6
C074 60	RTS	C0D0 A5 FF	LDA \$FF
C075 20 61 C0	JSR \$C061	C0D2 05 FE	ORA \$FE
C078 A0 00	LDY #\$00	C0D4 85 FF	STA \$FF
C07A A9 80	LDA #\$80	C0D6 06 FE	ASL \$FE
C07C 85 FE	STA \$FE	C0D8 CA	DEX
C07E A2 0D	LDX #\$0D	C0D9 10 EE	BPL \$C0C9
C080 B1 FB	LDA (\$FB),Y	C0DB A5 FF	LDA \$FF
C082 25 FE	AND \$FE	C0DD 91 FB	STA (\$FB),Y
C084 F0 04	BEQ \$C08A	C0DF 49 FF	EOR #\$FF
C086 A9 66	LDA #\$66	C0E1 91 F9	STA (\$F9),Y
C088 D0 02	BNE \$C08C	C0E3 AD CA C0	LDA \$C0CA
C08A A9 20	LDA #\$20	C0E6 18	CLC
C08C 9D 2B 88	STA \$882B,X	C0E7 69 28	ADC #\$28
C08F E8	INX	C0E9 8D CA C0	STA \$C0CA
C090 46 FE	LSR \$FE	C0EC 90 03	BCC \$C0F1
C092 D0 EC	BNE \$C080	C0EE EE CB C0	INC \$C0CB
C094 18	CLC	C0F1 C8	INY
C095 AD 8D C0	LDA \$C08D	C0F2 C0 08	CPY #\$08
C098 69 28	ADC #\$28	C0F4 90 C9	BCC \$C0BF
C09A 8D 8D C0	STA \$C08D	C0F6 A9 2B	LDA #\$2B
C09D 90 03	BCC \$C0A2	C0F8 8D CA C0	STA \$C0CA
C09F EE 8E C0	INC \$C08E	C0FB A9 88	LDA #\$88
C0A2 C8	INY	C0FD 8D CB C0	STA \$C0CB
C0A3 C0 08	CPY #\$08	C100 60	RTS
C0A5 90 D3	BCC \$C07A		
C0A7 A9 2B	LDA #\$2B		
C0A9 8D 8D C0	STA \$C08D		
C0AC A9 88	LDA #\$88		
C0AE 8D 8E C0	STA \$C08E		
C0B1 60	RTS		



DÜNYA

Tuna ERTEMALP

DÜNYA programı, SIMON'S BASIC ile yazılmış bir programdır. Bu programı SIMON'S BASIC'e sahip olan C64 kullanıcıları kadar, C128 ve PLUS/4 sahipleri de grafik komutlarını biraz değiştirerek kullanabilirler.

Programın konusu, Dünya'nın çeşitli açılardan küresel görünümünü çizmektir. Grafik kullanımı sırasında yeterli hıza ulaşabilmek için, SIMON'S BASIC kullanılmıştır.

Figürlerde de göreceğiniz gibi, yerküreyi hem kendi eksenini etrafında çevirmek hem de eksenin kendisini uzay içinde değişik açılara göre döndürmek mümkün olmaktadır. İsterseniz programın nasıl işlediğini biraz görelim.

SATIR 500-510: Bu satırlarda grafiğin hangi grafik biçiminde çizilmesi gerektiği sorulmaktadır. Eğer normal grafik seçilirse sadece tek renkle çizilen, 320 * 200 oranında bir çözünürlüğe (resolution) sahip grafik ekranı kullanılır. Multi-color halinde ise, 4 renkli ancak 160 * 200'lük bir çözünürlük kullanılır.

SATIR 600-700: Programın bu kısmında ise, tüm programda kullanılan sayılar değişkenlerin içine yerleştirilmekte ve normal veya multi-color kullanımına göre değişen sayılar saptanmaktadır. Sayıları değişkenler içine yerleştirmekteki amacımız, programın işlemini mümkün olduğu kadar çabuklaştırmaktır. Aynı nedenle, programın içinde



sıfır kullanılması gereken yerlerde bir nokta kullanılmıştır. Gerçekten de bilgisayarı bir noktayı sıfır olarak algılaması, sıfır rakamını tanımasından daha çabuk olmaktadır.

SATIR 710: Bu satırda, çizimi yapılacak yerküreye bakış açısı istenmektedir. Bu iki açının anlamları şöyledir: Dünyaya tam kuzey kutbundan baktığımızı varsayalım. Dönme açısı, bu durumda dünyanın saat ibresi yönünde kaç derece dönmesi gerektiğidir. Eksi bir açı, dünyayı saat ibresinin ters yönünde döndürecek. Şimdi de kuzey kutbu

yukarıda olmak üzere dünyaya karşıdan baktığımızı düşünelim. Eğiklik açısı da bu durumda, eksenin bize göre kaç derece eğik olması gerektiğini verecektir. Pozitif bir sayı kuzey kutbunu bize yaklaştıracak ve güney kutbunu uzaklaştıracak; eksi bir sayı ise kuzey kutbunu bizden uzaklaştırarak güney kutbunu yakınlaştıracaktır.

SATIR 731-732: Eğer belirtilmişse, multi-color grafiğe geçilir ve yerküreyi çevreleyen hafif basık çember çizilir.

SATIR 800-860: Bu satırlarda, ekrana çizilen çemberin doldurulması gerçekleşir. Toplam 22 değişik kara parçası çizilmektedir. Tabii ki bunların sadece o sırada görünen kısımları çizilecektir. İşte bunun için 3000. satırdan itibaren başlayan bir alt-program kullanılır. Bu alt-program, DATA satırlarından okunan enlem ve boylamları ekran koordinatları haline getirir. Satır 830'da da bu ekran koordinatları, eğer çemberin içinde kalıyorlarsa, çizilirler. Bir kara parçasının sınırlarının çizimi bittikten sonra sıra içinin boyanmasına gelir. Her bir kara parçası için yeterli sayıda enlem ve boylam noktası verilmiştir. Bu noktalardan o anda görünen bir tanesi seçilerek o noktadan itibaren boyama işlemi gerçekleşir.

SATIR 1000-1160: Artık sıra enlem ve boylamların çizimine geldi. Önce güney kutbundan kuzey kutbuna doğru

paraleller, sonra da Greenwich'den başlayarak meridyenler çizilir.

SATIR 1160-2999: Eğer multi-color kullanılıyorsa, daha güzel bir görünüm vermek için yerkürenin çevresi koyu mavi ile boyanır. Sonra da kullanıcının bir düğmeye basması beklenir. Bir tuşa basıldığında ise program yeniden çalışmaya başlar. MPS801, VC1525, MPS803 veya bunlar gibi grafik basabilen yazıcısı olanlarınız, bu satırlarda yapacakları ufak bir değişiklik ile istedikleri zaman grafiklerini kâğıda aktarabilirler. Bunun için 2999. satır aşağıdaki gibi değiştirmek gerekir:

2998 geta\$:ifa\$="" then 2998

2999 ifa\$="p" then copy:else:run

Ayrıca 1160. satırdaki 2999 da 2998 olarak değiştirilmeli. Artık grafik bittiğin-



de "p" tuşuna basmak dünya resmini kâğıda geçirecektir.

SATIR 3000-3070: Bu alt-program kendisine gönderilen enlem ve boylamları ekran koordinatlarına çevirir ve z değişkeni içerisinde de bu koordinatın çizilip çizilmeyeceği bulunur. Eğer bu değişken pozitif bir sayı ise çizim yapılır.

SATIR 4000-4188: Toplam 22 kara parçasına ait enlem ve boylamlar. Her biri için önce kaç tane enlem-boylam çifti olduğu yazmaktadır. İlk iki sayı, o kara parçasının başlangıç koordinatlarıdır. O parçayı çevreleyen sınırların koordinatlarından sonra, o kara parçasının içinde bulunan bazı noktaların koordinatları yer alır. Bu koordinatlar da, kara parçasının içinin boyanmasında kullanılır.

SATIR 4000-4007: Güney Amerika
SATIR 4100-4106: Afrika
SATIR 4107-4108: Madagaskar
SATIR 4109-4116: Antarktika
SATIR 4117-4121: Avustralya
SATIR 4122-4124: Yeni Zelanda
SATIR 4125-4127: Yeni Gine
SATIR 4128-4129: Borneo
SATIR 4130-4131: Sumatra
SATIR 4132-4143: Kuzey Amerika
SATIR 4144-4145: Newfoundland
SATIR 4146-4147: Küba

SATIR 4148-4150: Baffin
SATIR 4151-4152: Victoria
SATIR 4153-4154: Ellismere
SATIR 4155-4157: Grönland

SATIR 4158-4177: Avrasya
SATIR 4178-4179: İngiltere
SATIR 4180-4181: İrlanda
SATIR 4182-4183: İzlanda
SATIR 4184-4186: Japonya
SATIR 4187-4188: Sri Lanka

(SHIFT-↑)

```
500 cset0:print"5001) normal":print"5002) multi-color"
510 getrp$:ifrp$<>"1"andrp$<>"2"then510
600 a=1:c=22:d=80:e=57:f=100:g=-90:h=67.5:k=22.5:m=350:o=10:p=337.5:q=22.5
601 r=360/180:s=360:u=3:il=2:se=80:fc=1
700 ifrp$="1"thend=160:fc=2
701 :
710 input"5003)donme acisi 00000";dr:input"5004)egiklik
730 hires6,15:poke53281,14
731 ifrp$="2"thenmulti7,6,2
732 arcd,f,,s,3,fc*58,101,a
800 forj=atoc:readn,1,b
810 gosub3000:x1=x:y1=y
820 fori=aton:readl,b:gosub3000:x2=x:y2=y
825 ifz<.thenifza<.goto840
830 lined+fc*e*x1,f-f*y1,d+fc*e*x2,f-f*y2,a
840 x1=x2:y1=y2:next
845 readn:fornn=aton
850 readl,b:gosub3000
852 ifz>.thenpaintd+fc*e*x,f-f*y,a
855 next
860 next
900 ifrp$="1"thenu=2
1000 forbb=gtohstepk
1010 forll=.tomstepo
1020 b=bb:l=ll:gosub3000:x1=x:y1=y
1030 b=bb:l=ll+o:gosub3000:x2=x:y2=y
```



```

1035 ifz<.goto1050
1040 lined+fc*e*x1,f-f*y1,d+fc*e*x2,f-f*y2,u
1050 next:next
1100 forll=.topstepq
1110 forbb=gtosestepo
1120 b=bb:l=ll:gosub3000:x1=x:y1=y
1130 b=bb+o:l=ll:gosub3000:x2=x:y2=y
1135 ifz<.then1150
1140 lined+fc*e*x1,f-f*y1,d+fc*e*x2,f-f*y2,u
1150 next:next
1160 ifrp$="1"then2999
1200 paint,,,il:paint159,,,il
2999 geta$:ifa$=""then2999:else:run
3000 b=b*r
3001 l=l-dr
3002 ifl>s thenl=l-s:goto3002
3003 ifl<. thenl=l+s:goto3003
3004 l=l*r
3010 x=sin(l)*cos(b)
3020 v=cos(l)*cos(b)
3030 t=sin(b)
3040 y=t*cos(wi)-v*sin(wi)
3050 za=z:z=t*sin(wi)+v*cos(wi)
3070 return
4000 data31
4001 data-70,12,-61,10,-56,6,-52,5,-50,0,-40,-4,-36,-6
4002 data-35,-9,-39,-15,-41,-22,-48,-26,-49,-29
4003 data-55,-35,-57,-35,-58,-39,-61,-39,-68,-47
4004 data-66,-48,-70,-51,-63,-55,-70,-56,-75,-53
4005 data-76,-49,-74,-37,-72,-30,-70,-18,-76,-14
4006 data-81,-6,-80,0,-77,7,-75,11,-70,12
4007 data4,-65,8,-45,-10,-65,-35,-75,-5
4100 data37
4101 data-6,35,2,36,10,37,10,33,20,31,21,33,32,32
4102 data36,20,44,11,51,12,50,8,47,3,39,-4,41,-14
4103 data35,-19,35,-24,33,-26,31,-30,26,-34,19,-34
4104 data17,-28,15,-23,12,-18,14,-11,13,-5,9,-1,10,4
4105 data6,4,4,6,-2,5,-8,5,-12,8,-17,15,-16,21,-12,28
4106 data-10,30,-9,32,-6,35,3,0,20,30,10,25,-15
4107 data5
4108 data49,-12,51,-15,47,-25,43,-25,44,-15,49,-12,1,46,-18
4109 data34
4110 data-180,-89,-180,-85,-155,-86,-159,-77,-122,-72
4111 data-100,-74,-100,-72,-75,-73,-70,-66,-55,-63
4112 data-65,-68,-60,-73,-70,-80,-35,-80,-35,-77,-20,-72
4113 data0,-70,20,-70,40,-69,53,-66,70,-67,70,-70
4114 data80,-68,85,-66,100,-66,115,-65,120,-66,135,-65
4115 data150,-68,166,-70,170,-72,162,-75,160,-82
4116 data180,-85,180,-89,3,120,-70,40,-70,-100,-80
4117 data20
4118 data142,-10,148,-20,153,-25,153,-32,150,-38,140,-38
4119 data134,-32,126,-32,119,-34,116,-33,115,-30,114,-25
4120 data115,-22,121,-20,127,-14,130,-15,130,-11,136,-11
4121 data135,-15,141,-18,142,-10,2,130,-25,140,-27
4122 data5
4123 data173,-35,178,-38,170,-47,167,-46
4124 data175,-37,173,-35,1,176.5,-38.5
4125 data11
4126 data130,0,134,0,136,-2,138,-1,146,-5,150,-10,145,-8
4127 data143,-10,138,-8,138,-4,132,-3,130,0,1,142,-5
4128 data5

```



```

4129 data117,7,119,5,116,-4,110,-4,109,2,117,7,1,113,0
4130 data4
4131 data98,5,105,-3,104,-6,96,5,98,5,1,101,0
4132 data56
4133 data-80,25,-81,25,-83,30,-95,30,-98,27,-98,22
4134 data-95,19,-90,21,-87,21,-89,15,-84,15,-84,10
4135 data-75,10,-85,10,-95,17,-105,20,-112,32,-117,32
4136 data-110,23,-117,34,-120,34,-125,40,-125,48
4137 data-130,54,-140,60,-150,60,-165,55,-158,58
4138 data-165,62,-160,64,-166,64,-166,66,-160,66
4139 data-166,68,-156,71,-135,69,-127,70,-110,68
4140 data-81,70,-81,66,-95,62,-92,57,-83,55,-83,52
4141 data-79,52,-79,55,-77,57,-78,63,-70,58,-65,60
4142 data-56,52,-65,50,-62,45,-75,42,-76,36
4143 data-81,32,-80,25,4,-150,65,-70,54,-110,40,-90,35
4144 data3
4145 data-53,47,-59,48,-56,52,-53,47,1,-57,49
4146 data2
4147 data-85,22,-82,23,-75,21,1,-85,22
4148 data7
4149 data-66,63,-77,65,-72,68,-89,70,-90,73,-79,73
4150 data-62,66,-66,63,1,-69,64
4151 data4
4152 data-101,70,-115,70,-120,73,-105,74,-101,70,1,-110,72
4153 data4
4154 data-80,76,-90,76,-95,82,-70,83,-80,76,1,-85,77
4155 data10
4156 data-45,60,-50,62,-55,70,-60,76,-60,76,-70,78,-40,85
4157 data-20,80,-22,70,-40,65,-45,60,1,-40,70
4158 data117
4159 data76,8,72,21,70,20,66,25,55,26,50,30,48,28
4160 data52,24,56,26,60,22,57,19,52,16,44,12,41,20
4161 data33,32,34,32,36,37,27,36,27,42,42,42,38,47
4162 data30,46,29,42,22,40,25,38,22,36,20,40,20,42
4163 data13,46,12,44,17,40,15,37,12,38,17,39,10,44
4164 data7,43,3,43,3,41,0,40,0,38,-2,37,-6,36,-7,37
4165 data-9,37,-9,43,-2,43,-2,46,-4,48,9,54,8,57
4166 data10,58,14,54,21,55,25,66,21,65,18,62,16,56
4167 data12,55,10,60,5,57,5,63,20,70,25,71,43,67
4168 data65,70,70,73,80,72,80,74,105,78,112,76,113,74
4169 data130,73,130,71,150,72,160,70,175,70,170,66
4170 data180,65,180,63,170,60,162,60,162,56,157,52
4171 data156,58,160,60,157,62,155,59,141,59,136,55
4172 data141,54,140,47,134,43,128,39,129,35,126,35
4173 data125,40,118,39,121,31,120,26,116,23,109,21
4174 data105,19,109,16,109,11,105,10,100,14,100,8
4175 data104,2,100,4,98,8,97,17,94,16,94,20,92,22
4176 data88,22,80,16,80,10,76,8
4177 data5,20,50,45,25,80,25,110,30,100,70
4178 data4
4179 data-5,58,-3,58,1,52,-5,50,-5,58,1,-2,53
4180 data4
4181 data-7,55,-7,52,-10,52,-10,54,-5,55,1,-8,54
4182 data4
4183 data-22,67,-15,67,-13,65,-20,64,-22,67,1,-18,66
4184 data6
4185 data141,45,145,44,140,35,130,31,130,34,139,38
4186 data141,45,1,139,37
4187 data3
4188 data80,10,82,8,80,7,80,10,1,81,3
ready.

```