

C64 TÜRKİYE

SAYI : #S01

KASIM 2002

BİLGİ PAYLAŞTIKÇA ARTAR

```
**** COMMODORE 64 BASIC V2 ****  
64K RAM SYSTEM 38911 BASIC BYTES FREE  
READY.
```

BU SAYIDA.....

TURBO ASSEMBLER

ASSEMBLER KURSU – 1

C64 BELLEK HARİTASI – 1

MAKİNA DİLİ EDITÖRÜ

PROGRAM KÖŞESİ

İÇİNDEKİLER

BU SAYFA	2
TURBO ASSEMBLER	3
ASSEMBLER KURSU – 1	6
BELLEK HARİTASI – 1	9
MAKİNA DİLİ EDITÖRÜ	13
PROGRAM KÖŞESİ	16
BEN KİMİM ?	17

- 1- *Dergi için COMMODORE 64 ile ilgili her türlü yazı, resim, program, oyun tanıtımı, ipuçları, hile ve ufak-tefek rutinler gibi malzemeler kabul edilir.*
- 2- *Yukarıdaki malzemelerin çalışır durumda olduğundan emin olunmalıdır.*
- 3- *Gönderilen malzemeler dergide mutlaka yayınlanacak diye herhangi bir şart yoktur.*
- 4- *Malzemeyi gönderenin adı-soyadı-e-mail adresi-oturduğu şehir ve kendisini tanıtan kısa bir not yazması tercih edilir.*
- 5- *Derginin editörü gönderilen malzemelerde değişiklik, düzenleme vs yapabilir.*
- 6- *Gönderilen malzemelerin sorumluluğu gönderene aittir.*
- 7- *Hiç bir şekilde siyasi, dini yazılara yer verilmeyecektir. Ayrıca kişilere ve şirketlere yönelik hakaret, aşağılama, küfür gibi yazılar yayınlanmayacaktır.*
- 8- *Emeğe saygı gereği kaynak göstermek şartıyla herkes dergideki yazılardan alıntı yapabilir.*
- 9- **BU DERGİ KESİNLİKLE TİCARİ AMAÇLI DEĞİLDİR VE PARA İLE SATILMAZ.**
- 10- *Dergiyi okuyan herkes bütün bu maddeleri kabul etmiş sayılır.*

C64 TÜRKİYE

Hazırlayan ve sahibi : İSMAİL “HADES” ŞAHİN
Katkıda bulunanlar : Şimdilik yok.

EDITÖRÜN KLAVYESİ

ÖNSÖZ : Günümüzde PC denilen para tuzağı bilgisayarların hiç durmadan gelişmesine !! rağmen neredeyse 20 sene önce üretilen bir ev bilgisayarı olan COMMODORE 64’ü hala ilk günkü heyecanla kullananlar var. Bende bunlardan biriyim. Ve şu anda ülkemizde belki bir ilki gerçekleştiriyorum. Bir dergi çıkartıyorum. Ama bu dergi sadece benim çabalarımla değil, aynı zamanda sizlerinde katkılarıyla çıkmalı. Yıllardır uyuyan bir deviyandırmak için elimizde her türlü teknoloji var artık. Belki sizlerde böyle bir şey yapmak istediniz. Fakat nereden başlayacağına nasıl olacağına karar veremediniz. Belki de yaptınız ama devam ettirmediniz. Ne olursa olsun artık bir araya gelip bir şeyler üretmenin zamanı geldi. İnternet ortamında bir çok yabancı C64 sitesi var. Ama çoğu emülatör, oyun, demo vs. Sitesi. Birde beni böyle bir dergi çıkarmak için gaza gelmemi sağlayan İngilizce ve enson 21. Sayısı çıkan text formatında bir dergi olan C=HACKING dergisi ile sadece 3 sayılık yine İngilizce DISCOVERY var. Ama İngilizce bilmediğim için sadece program kısımlarına bakıp geçiyorum. Yurt dışında durum böyleyken ülkemizde ise iki elin parmak sayısı kadar site bile yok diyebiliriz. Bu nedenlerle elimden geldiği, dilimin döndüğü kadar bu dergiyi hazırlayacağım. Belki çok az kişi hatırlar, C64’ü Türkiye’ye getiren TELETEKNİK firmasının C64 ilanlarında kullandığı bir slogan vardı.

İÇİNİZDEKİ DAHİYİ UYANDIRIN!

İşte benim yapmak istediğimde bu. Belki bazılarınız “bu saatten sonra artık kim uğraşır antika bir bilgisayarla” diyecektir. Eğer internetteki C64 sitelerine bakarsanız ne kadar çok antikacı olduğunu görürsünüz. Üstelik COMMODORE ONE isimli bir model üzerinde çalışmalar yapılyorsa bu antika basit bir antika değil demektir.

Her neyse, yandaki sütuna göz attıysanız ilk sayının içeriği hakkında bir fikir sahibi olmuştunuzdur. Bunların dışında çeşitli ufak-tefek programlar, ipuçları vs. Dergimizde her zaman olacaktır. Ayrıca sizlerden gelen programlara ve yazılara da yer vereceğim. Unutmadan görüşlerinizi, eleştirilerinizi en önemlisi bilgilerinizi paylaşabilirsiniz. Benimle iletişim kurmak için..

1 – HADES6510@HOTMAIL.COM

2 – HADES6510@YAHOO.COM e-maillerini ve

3 - <http://hades6510.sitemynet.com> ‘daki ziyaretçi defterini kullanabilirsiniz.

Bir sonraki sayıda buluşabilmek dileğiyle....
Herkes bol klavyeli günler....

TURBO ASSEMBLER

TURBOASSEMBLER C64'te assembler program yazan (intro,demo,utility vs.) bir çok kişinin kullandığı bir programdır. Birkaç versiyonu olmakla birlikte genelde aralarında fazla fark yoktur. Benim kullandığım 5.4b versiyonudur. TURBOASSEMBLER'i bundan sonra TASS olarak adlandıracam.

TASS yüklendiğinde \$9000 (36864) adresinden itibaren çalışmaya başlar. Herhangi bir nedenle programdan çıkarsanız RESTORE tuşu ile veya SYS 36864 komutu ile TASS tekrar çalışmaya başlar. Eğer bir aksilik yoksa yazdıklarınız karşınızda duracaktır.

TASS ile bir çok işi yapabilirsiniz. Birazdan TASS komutlarını yazacağım. Ama gözünüzü korkutmasın. Zaten kullanacağınız komutlar fazla olmayacaktır. Başlıyoruz...

Bir program yazmaya başlamadan önce MUTLAKA programın hangi adresten başlayacağını belirtmeniz lazım. Bunun için kullanacağınız komut şöyledir.

*=\$.... (ÇARPI, EŞİTTİR, DOLAR İŞARETİ, HEX ADRES)

Eğer HEX yerine DECİMAL adres kullanmak isterseniz dolar işaretine gerek yoktur.

Program içinde birden fazla olarak bu komutu kullanabilirsiniz.

Ayrıca 8 karaktere kadar etiket kullanabilirsiniz ama bazı kuralları vardır.

- 1 – Etiketler bir ASSEMBLER komutu olmamalıdır.
- 2 – Etiketler bir rakam ile başlayamaz.
- 3 – Etiketlerde BOŞLUK karakteri ve diğer noktalama işaretleri kullanılamaz.
- 4 – Etiketlerde SADECE RAKAM ve HARF kullanabilirsiniz.

TASS ile önceden tanımlanmış Fonksiyon tuşlarını kullanabilirsiniz. Bu tuşların görevleri şunlardır.

F1 : Uzun bir programda 20 satır yukarı gidersiniz.

F2 : Uzun bir programda en üst satıra yani programın en başına gidersiniz.

F3 : .text komutunu ekrana yazar. Komuttan sonra tırnak içersinde satırdan taşmayacak şekilde bir yazı yazabilirsiniz. Eğer yazınız taşıyorsa bir alt satırda .text komutunu tekrar kullanmalısınız.

F4 : Programınızda bir hata yoksa TASS programınızı koda çevirir ve çalıştırır. Eğer hata varsa hatanın hangi satırlarda olduğunu, hatanın cinsini ve kaç tane hata olduğunu ekrana yazar.

F5 : .byte komutunu ekrana yazar. Komuttan sonra 255'e kadar olan sayıları bir satırdan taşmayacak şekilde, \$ kullanarak HEX sayıları ve % işareti ile BİNARY sayıları, işaret kullanmadan onlu sayıları kullanabilirsiniz. Eğer satırdan taşma oluyorsa bir alt satırda .byte komutunu tekrar kullanmalısınız. Sayılar arasında virgül kullanmanız gerekmektedir.

F6 : Kursörün bulunduğu satırı siler. **Silme işleminden hemen sonra eğer başka bir tuşa basmadıysanız RESTORE tuşu ile satırı geri getirebilirsiniz.**

F7 : Uzun bir programda 20 satır aşağı gidersiniz.

F8 : Uzun bir programda en alt satıra yani programın en sonuna gidersiniz.

Diğer komutlara geçmeden önce bir komut daha var ki Fonksiyon tuşlarıyla tanımlanmamıştır.

Bu komut **.word** komutudur. Yaptığı iş komuttan sonra gelen bir sayıyı – en fazla 65535'tir – LO bayt / HI bayt formatında hafızaya saklamaktır.

Mesela .word \$0801 ile bu sayıyı hafızaya \$01 - \$08 şeklinde saklarsınız. Zaten bütün mikroişlemciler iki baytlık sayıları bu şekilde kullanır.

Diğer komutlar : Bu komutlar ise TASS ile çeşitli işlemler yapabilirsiniz. Komutları kullanabilmek için önce “←” (SOL OK) tuşuna sonra diğer tuşa basmanız lazımdır.

← + 1 : TASS'tan çıkış. RESTORE tuşu ile tekrar TASS'a dönebilirsiniz.

← + 2 : Kursörün olduğu satıra yatay çizgi koyarsınız. Ayırıcı çizgi misali.

← + 3 : Yazdığınız programı koda çevirir. Hata varsa bildirir, hata yoksa programı çalıştırmak için “S” tuşuna, editöre geri dönmek için başka bir tuşa basmanızı bekler.

← + 4 : PRINT FILE (Diskete farklı bir kayıt yapıyor henüz çözemedim.)

⬅ + 5 : OBJECT FILE Yazdığınız programı kod olarak diskete kaydeder.
 ⬅ + 6 : \$... 'dan \$... 'a olan bölgedeki bayları, kursörün bulunduğu yere **.byte** komutu şeklinde programınıza ekler. Bir tür DATA GENERATÖR yani.
 ⬅ + 7 : Tanımsız.
 ⬅ + 8 : Tanımsız.
 ⬅ + 9 : Tanımsız.
 ⬅ + 0 : Tanımsız.
 ⬅ + "+" : HEX sistemde toplama yapar.
 ⬅ + "-" : HEX sistemde çıkartma yapar.
 ⬅ + pound işareti : Tanımsız.
 ⬅ + "INST/DEL" : Satır siler.
 ⬅ + "A" : Tanımsız.
 ⬅ + "B" : Önceden ⬅ + "M" ile tanımlanmış bir blok üzerinde işlem yapar.
 "W" tuşu ile seçilen bloku diskete SEQ file olarak kaydeder.
 "K" tuşu ile seçilen bloku siler.
 "C" tuşu ile seçilen blok kursörün olduğu satırdan itibaren kopyalanır.
 ⬅ + "C" : COLD START. "Y" ile her şey silinir ve TASS'ı yeniden başlatır..
 ⬅ + "D" : Disk komutları kullanabilirsiniz. "I" : INIT, "V" : VALIDATE vs..
 ⬅ + "E" : Daha önceden ⬅ + "W" ile diskete SEQ File olarak kaydedilmiş
 Bir programı kursörün olduğu satırdan itibaren araya ekler.
 ⬅ + "F" : Programda bir yazıyı vs. aratabilirsiniz. Kursör bulunan ilk yere gider.
 ⬅ + "G" : Daha önceden ⬅ + "M" ile numara verilerek tanımlanmış blok'a gider. Blok
 Numarası girilir.
 ⬅ + "H" : Programdaki en alt satıra gider.
 ⬅ + "I" : \$.... 'dan \$.... 'a kadar olan bölgeyi \$.. sayısı ile doldurur.
 ⬅ + "J" : \$.... 'dan \$.... 'a kadar olan bölgenin HEX ve ASCII dökümünü verir.
 ⬅ + "K" : F3, F4, F5 ve F6 tuşları tanımlanabilir.
 ⬅ + "L" : Önceden ⬅ + "S" ile diskete PRG file olarak kaydedilmiş programı yükler.
 ⬅ + "N" : Programdaki her hangi bir satıra gidebilirsiniz.
 ⬅ + "O" : Programın renklerini değiştirebilirsiniz.
 ⬅ + "P" : EOR CODE --- Ne işe yaradığını çözemedim.
 ⬅ + "Q" : Tanımsız.
 ⬅ + "R" : REPLACE --- Ne işe yaradığını çözemedim.
 ⬅ + "S" : Yazdığınız programı **-EKRANDA YAZILDIĞI ŞEKLİYLE -** PRG File olarak
 diskete kaydeder. "⬅ + "S" : OBJECT FILE " komutundan farklıdır.
 ⬅ + "T" : En alt satıra gider.
 ⬅ + "U" : Programda kullandığınız etiketleri ", " adıyla diskete kaydeder.
 ⬅ + "V" : --- Ne işe yaradığını çözemedim.
 ⬅ + "W" : Programı SEQ File olarak diskete kaydeder.
 ⬅ + "X" : Tanımsız.
 ⬅ + "Y" : En alt satıra gider.
 ⬅ + "Z" : Tanımsız.
 ⬅ + "*" : Disketin Directory'sini okur.
 ⬅ + ":" : İşaretlenen yerleri listeler.
 ⬅ + ";" : İşaretleri kaldırır.
 ⬅ + "=" : Kursörün bir üstündeki satırı siler.
 ⬅ + "@" : Disket sürücünün hata kanalını okur.
 ⬅ + "F1" : Fonksiyon tuşlarını sıfırlar.
 ⬅ + "F3" : Tanımsız.
 ⬅ + "F5" : Tanımsız.
 ⬅ + "F7" : Tanımsız.
 Son olarak program içinde yapacağınız açıklamalar ";" ile başlamalıdır.

Örnekler için tıklayınız ☺☺

DİKKAT : TASS kullanırken dikkat edeceğiniz önemli bir konu var. Yazacağınız programın kodlarının TASS'ın bulunduğu adreslere yerleşmemesi gerekir. Yoksa TASS sapıtır. Zaten çalışma sırasında ekranın en altında devamlı değişen bir adres göreceksiniz. Bu adres siz bir şeyler yazdıkça devamlı azalacaktır. Kısaca TASS ile çakışmayacak adreslerin hepsini kullanabilirsiniz. Ama benden tavsiye her zaman \$0900'den itibaren yazmaya başlayın. Bazen bilgisayarı resetlemek durumunda kalabilirsiniz. Bu nedenle daha düşük adreslerdeki programlara – aslında \$0800'ün altında fazla bir yer yok – elveda diyebilirsiniz. Eğer resetledikten sonra SYS 36864 komutuyla TASS'ı tekrar çalıştırırsa bir aksilik yoksa programınız karşınızda duruyor olacaktır. Adresleri kontrol etmeniz gerekir.

NOT : Yazacağınız programda kullanacağınız SAYAC, GEÇİCİ SAKLAMA ADRESİ, TABLO, YAZI gibi hafıza kullanan bölümleri programınızın en sonuna yazarsanız daha iyi olur.

ÖRNEK 1 :

```

*= $0900
BASLA   LDA #<YAZI           ; Yazının alçak baytı aküye
        LDY #>YAZI           ; yüksek baytı Y registerine koy.
        JSR YAZ              ; Rom'daki alt rutini çağır.
        RTS                  ; Programdan çık.
YAZ      = $AB1E
YAZI     .TEXT "DENEME"
        .BYTE 0

```

Yukarıdaki örnekte programın başlangıcı olarak \$0900 = 2304 yazıyoruz. Sonra arka arkaya iki komutla yazımızın bulunduğu yerin adreslerini belirliyoruz. Daha sonra ROM'da bulunan ekrana yazı yazdıran alt rutini çağırıyoruz. En sonunda ise RTS komutuyla programımızı bitiriyoruz. İsterseniz JSR YAZI yerine doğrudan JSR \$AB1E yazabilirsiniz. Ayrıca dikkat etmeniz gereken bazı konular vardır. ROM'daki bir alt rutini çağırırken hangi registerler kullanılır, sonuç hangi registerde veya hangi adreslerde tutulur vs. \$AB1E adresine sızmadan önce A registerine yazının bulunduğu adresin LOW BAYT'ı Y registerine ise HIGH BAYT'ı yüklenmelidir. Son olarak ta yazının bitişinden hemen sonraki adrese mutlaka 0 değeri yazılmalıdır. Yoksa alt rutin yazının bittiğini anlayamaz ve ekrana garip yazılar çıkabilir. Örneği inceleyin.

ÖRNEK 2 :

```

*= $0900
LDA #(TAB/64)
STA SPRPNT
RTS
SPRPNT   = $07F8
TAB      = $02C0

```

END
PRG

ÖRNEK 3 :

```

*= $0800
.BYTE 0
.WORD END
.WORD 2002
.BYTE $9E,$20
.TEXT "2068 HADES"
.BYTE 0
.BYTE 0,0
INC $D020
JMP PRG

```

İkinci örneğimizde birazcık matematik yaptık. C64'ün özelliklerinden biri olan SPRITE ile ilgili çok ufak bir deneme olan bu örnekte 1. sprite için Sprite'ın datalarının bulunduğu hafıza adresi olarak \$02C0'ı tanımladık ve sonra bu değeri 64'e bölerek sprite pointerine yazdık. Bunu yapınca VIC çipi artık 1. sprite'ın şeklinin nerede olduğunu bilecektir. Meraklanmayın BELLEK HARİTASI bölümünde sırası gelince SPRITE'larla ilgili bütün bilgileri vereceğim.

3. örnektekileri yazın fakat çalıştırmayın. Sadece ◀ + "3" komutunu kullanın. Sonra ◀ + "1" ile TASS'tan çıkın. LIST komutunu verin. Ekranda ne görüyorsunuz ?

2002 SYS 2068 HADES satırı değil mi ? Şimdi RUN komutu verin. Programın açıklamasını artık yapabilirsiniz. Zaten TASS'ı kullandıkça daha karmaşık programlar yazabileceksiniz.

TURBOASSEMBLER dersi burada bitti. Eğer eksik ve/veya hatalı yerler varsa bana bildirin. Bir sonraki sayıda düzeltmeler bölümü açıp oraya ekleyelim.... Herkese şimdiden kolay gelsin.....

ASSEMBLER KURSU – 1

Yeni yazı dizimize hoş geldiniz. Bir bilgisayarda bir çok programlama dili ile çalışabilirsiniz. İlk zamanlarda PASCAL, COBOL, FORTRAN, LISP, C, ADA, FORTH ve BASIC gibi birçok programlama dilleri bilgisayarlar için kullanılmaktaydı. Bu dediklerimden BASIC standart olarak COMMODORE 64'te ve diğer ev bilgisayarlarında çeşitli versiyonlar şeklinde bulunmaktaydı. Diğer diller ise ayrı bir program olarak bilgisayara yükleniyordu. Hangi dili kullanırsanız kullanın aslında bir bilgisayarın anladığı tek dil makina dilidir. Yazılan her komut ROM bellekte bulunan bir YORUMLAYICI (INTERPRETER) tarafından yorumlanır ve ne yapması gerektiğine karar verilerek komut işletilir. YORUMLAYICI ise tamamen makina dili ile yazılmış bir programdır. Eğer komut yanlış yazılmışsa gerekli hata mesajını yazarak çalışmayı durduracaktır. Aşağıdaki örnek ile ne demek istediğimi daha iyi anlayacaksınız. Çok basit bir örnek vereceğim.

PRINT "DENEME" yazıp RETURN tuşuna basınca neler oluyor inceleyelim.

1 – İşletim sistemi (yorumlayıcıdan ayrıdır ve basit olarak her türlü giriş çıkış işlemleriyle uğraşır- klavye,ekran,disket vs.) basılan her tuşu okuyarak hafızada klavye tamponu denilen özel bir yere kaydeder.

2 – Eğer RETURN tuşuna basıldıysa klavye tamponundaki karakterler tek tek okunur ve bir komut olup olmadığı kontrol edilir. Eğer komut değilse **SYNTAX ERROR** hatası verir. Eğer bir komutsa bu sefer o komuta ait alt programlar çağrılır ve komuttan sonra gelen ifade çözülür. Bu örnekte tırnak işareti kontrol edilir ve tırnak içindeki ifadede yer alan karakterler tek tek kontrol edilir. Çünkü C64 bu karakterleri okuyarak ekranı silme, yazı rengi değiştirme, yukarı, aşağı vs. Kursör pozisyonu değiştirme gibi işlemleri de yapabilir. Eğer biz **PRINT DENEME** yazsaydık komut yine çalışacaktı ama bu sefer DENEME ifadesini bir yazı olarak değil bir değişken olarak kabul edecek ve değişkenin değerini ekrana basacaktı.

3 – Komuttan sonra gelen ifadeyi çözümlemek yorumlayıcının işidir. Buraya kadar herşey normal. Bundan sonra işlem bu yazıyı nereye göndereceğimizdir. Bunun için de o anki aktif çıkış kanalı kontrol edilir ve işlem yapılır. Çıkış kanalı normalde ekrandır ama siz bunu printer, disket olarakta tanımlayabilirsiniz. Bunun için ise işletim sistemini kullanarak bir kaç değişiklik yapmanız gereklidir.

Gördüğünüz gibi basit bir komut bile ne kadar çok işlem gerektiriyor. Bütün bu işlemler ise sadece makina dili ile yazılmış yorumlayıcı ile işletim sistemi tarafından yapılıyor. Tek komut için belki sorun yok ama aynı işi 256 kez yapmaya kalkarsanız çok zaman harcarsınız.

FOR A=0 TO 256 : PRINT "DENEME" : NEXT

Çünkü yukarıdaki işlemler 256 kez tekrarlanacaktır. İşte burada HIZ kavramı işin içine giriyor. Gerçektende assembler ile yazılan programlar BASIC ile yazılanlara göre nerdeyse 1000 kat hızlı çalışır. Ama bu her programı tamamen ASSEMBLER ile yazabiliriz anlamına gelmez. Mesela $(12345 + 67890) * (1/4096) - (\sin 5 * \cos 1)$ gibi bir hesaplamayı assembler ile yapmak için bir program yazmak kafayı sıyırmaya yol açabilir. Ama BASIC kullanarak hem sprite hareket ettirin hem müzik çaldırın hemde ekranı kaydırın dediğimiz zaman orada kalırsınız. Bu nedenle yerine göre BASIC yerine göre ASSEMBLER kullanmanız gerekir. Fakat bu yazı dizimiz bittiğinde assembler ile ufak tefek programlar yapmaya başlayınca assemblerin diğer adıyla makina dilinin zor ama çok güzel olduğunu fark edeceksiniz. Zaman zaman dergide ufak tefek makina dili rutinlerine yer vereceğim.

Konuyu fazla dağıtmadan ilk dersimize başlıyoruz. İlk dersimiz **SAYI SİSTEMLERİ** olacak. Günlük yaşantımızda her ne kadar onlu sayı sistemini kullanıyor olsakta makina diliyle uğraşacaksınız bu sayı sistemini unutun. Herşeyi ikili (BİNARY) ve onaltılı (HEXADECİMAL) sistemde BİT ve BAYT seviyesinde düşüneceksiniz. Korkmanıza gerek yok. Gerçekten çok basittir.

İKİLİ (BİNARY) SAYI SİSTEMİ

İkili sayı sisteminde sadece iki sayı kullanılır “0” ve “1”. İşlemciler bu 0 ve 1’lerin bir araya gelmesiyle bazı işlemler yaparlar. 0 ve 1 durumu alabilen en küçük bilgi birimine **BİT** denir. İşte bu bitlerin 4 tanesi bir araya gelirse **NİBBLE** olur. Sekiz bit ise bir **BAYT** eder. 1024 BAYT = 1KBAYT eder. Commodore 64’ün işlemcisi 8 bitlik olduğu için bir seferde işlem yapacağı sayı 8 bitlik olacaktır. Peki ama 8 bitlik bir sayı kaçtır ? 10, 50, 300 belkide 1000. Acaba gerçekten öylemi ? Kullanacağımız sayılar hep 8 bitlik olacaktır. Bu nedenle önce sayı sistemleri arasındaki hesaplamaları öğrenmek gerekmektedir. Assembler ile bir program yazarken kullanacağınız bazı özel adreslerde BİT düzeyinde işlem yapıldığı için hesaplama yapmak gerekebilir.

Bir baytlık sayı 8 bittten oluşur demiştik. Bu bitler soldan sağa doğru **7 6 5 4 3 2 1 0** nolu bit olarak dizilirler. Her bitin gerçekte karşılık geldiği sayı ise **2[↑] BİT NUMARASI** şeklindedir. Bu sayıya çarpan değeri diyebiliriz. Aşağıdaki tabloyu inceleyin.

BİT NUMARASI	7	6	5	4	3	2	1	0
ÇARPAN DEĞERİ	128	64	32	16	8	4	2	1

İKİLİ SİSTEMDEN ONLU SİSTEME GEÇİŞ :

İkili sistemdeki bir sayıyı onlu sayıya çevirmek için her bitin durumu (0 veya 1) o bitin çarpan değeri le çarpılır ve çıkan sonuçlar toplanır. Örneklere bakın.

İkili sistemdeki sayıların belirtmek için sayının önüne % işaretini koyacağız..

%00000000 = 128*0 + 64*0 ++ 2*0 + 1*0 = 0 En küçük sayımız 0 dır.

%11010011 = 128*1 + 64*1 + 32*0 + 16*1 + 8*0 + 4*0 + 2*1 + 1*1 = 211

%11111111 = 128*1 + 64*1 ++ 2*1 + 1*1 = 255 En büyük sayımız ise 255’tir.

255’ten büyük sayıları ise 2 bayt olarak tanımlayacağız ve hesaplayacağız. Ama önce ONALTILI (HEXADECİMAL – kısaca HEX diyebiliriz.) sayı sistemini göreceğiz.

ONALTILI (HEXADECİMAL) SAYI SİSTEMİ

HEXADECİMAL sayı sisteminde **0 1 2 3 4 5 6 7 8 9 rakamları ile A B C D E F harfleri kullanılır.**

A=10 B=11 C=12 D=13 E=14 F=15 sayılarının yerine kullanılır. Şimdiden söylemekte fayda görüyorum. Assembler ile program yazarken her zaman onaltılı sayı sistemini kullanacağız.

Hatta ikili sistemden daha çok kulanacağız. Bu nedenle iyi öğrenilmesi gerekmektedir.

Onaltılı sistemde bir sayı yazarken sayının önüne \$ işaretini koyacağız. Onlu sistemdeki sayılar için işaret kullanmayacağız Program yazarken kullanacağımız onaltılı sistemdeki bir sayı için 2 veya 4 karakter gereklidir. İki karakter kullanırsak sayı bir baytlık, 4 karakter kullanırsak sayı iki baytlıktır.

KARAKTER	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
SAYI DEĞERİ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

ONALTILI SİSTEMDEN ONLU SİSTEME GEÇİŞ :

Eğer sayı bir baytlık ise sayının soldaki basamağındaki karakterin sayı değeri 16 ile çarpılır sağdaki karakterin sayı değeri ile toplanır. Örneklere bakın.

\$00 = 16*0 + 0 = 0

\$1A = 16*1 + 10 = 26

\$88 = 16*8 + 8 = 136

\$B3 = 16*11 + 3 = 179

\$FF = 16*15 + 15 = 255

Gördüğünüz gibi en küçük sayımız 0 en büyük sayımız ise 255’tir.

Eğer sayı 255’ten büyükse sayımız iki baytlık demektir ve 4 adet karakter kullanılır.

İki baytlık bir HEX sayıyı onlu sayıya çevirmek için iki yol vardır.

1-HEX sayı ikiye ayrılarak iki adet bir baytlık sayı haline getirilir. Sonra her biri onlu sayıya çevrilir. Daha sonra ise soldan itibaren ayrılmış olan kısmın sonucu 256 ile çarpılır ve diğer sonuca eklenir. Örneğe dikkat edin.

\$0123 sayısını \$01 ve \$23 olarak ikiye ayıralım. Sonra bunları tek tek hesaplayalım
 \$01 = 1 ve \$23 = 35 olur. Daha sonra ise \$01'in sonucunu 256 ile çarparak yeni sonucu \$23'ün sonucuyla toplayalım.

$$256 * 1 + 35 = 256 + 35 = 291$$

Demekki \$0123 sayısının onlu karşılığı 291 oluyormuş.

Örnek2 : \$FCE2 => \$FC = 252 ve \$E2 = 226 => $256 * 252 + 226 = 64738$

2-Sayının basamaklarındaki karakterlerin sayı değeri soldan sağa doğru 4096, 256, 16 ve 1 ile çarpılır ve her basamağın sonucu toplanır.

$$\text{\$ABCD} = 4096 * 10 + 256 * 11 + 16 * 12 + 1 * 13 = 40960 + 2816 + 192 + 13 = 43981$$

$$\text{\$FFFF} = 4096 * 15 + 256 * 15 + 16 * 15 + 1 * 15 = 61440 + 3840 + 240 + 15 = 65535$$

Onaltılı bir sistemde elde edebileceğimiz en büyük sayı 65535'tir ve bu sayı COMMODORE 64'ün hafızasındaki en son adresi gösterir.

ONALTILIDAN İKİLİYE VE İKİLİDEN ONALTILIYA GEÇİŞ :

Sayının karakterlerine aşağıdaki tablodan karşılık gelen kısımlar yanyana yazılarak ikili sistemdeki karşılığı bulunmuş olur. Aynı şekilde ikili sistemdeki bir sayı soldan 4 basamağı ayrılarak iki gruba bölünür ve karşılık gelen HEX karakter yazılırsa sayının HEX karşılığı bulunmuş olur.

HEX SAYI	İKİLİ SİSTEMDEKİ KARŞILIĞI
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

Örnek : \$3C = % 0011 1100

$$\%10110011 \Rightarrow 1011 = B \text{ ve } 0011 = 3 \text{ bulunarak } \%10110011 = \$B3 \text{ olur.}$$

Gördüğünüz gibi sayı sistemleri ve çevirme işlemleri aslında çok basittir.

Peki onlu bir sayıyı onaltılı bir sayıya nasıl çevireceğiz.

1 – Sayı 256'dan küçükse : Sayıyı 16'ya bölün, eğer sonuç tam sayı ise HEX karşılığını yazın ve sonuna bir tane 0 ekleyin. Eğer sonuç tam sayı değilse tam kısmının HEX karşılığını yazın, daha sonra tam kısmını 16 ile çarpın ve sonucu asıl sayıdan çıkartın. Çıkarma işleminin sonucunun HEX karşılığını ise daha önce yazdığımız HEX karşılığın yanına ekleyin.

Örnek : 239'un HEX karşılığı

$$239 / 16 = 14.93 \rightarrow 14 = E \rightarrow 14 * 16 = 224 \quad 239 - 224 = 15 = F \quad 239 = \$EF$$

2 – Sayı 256'dan büyük ise : Sayıyı 256'ya bölün. Tam kısmına 1. Maddedeki işlemleri uygulayın. Tam kısmı 256 ile çarpın asıl sayıdan çıkartın. Sonuca 1. Maddedeki işlemleri uygulayın.

Örnek : 45891'in HEX karşılığı

$$45891 / 256 = 179.26 \text{ tam kısmı } 179 = \$B3 \quad 179 * 256 = 45824$$

$$45891 - 45824 = 67 \quad 67 = \$43 \text{ bulunur. Demekki } 45891 = \$B343 \text{ oluyormuş.}$$

Assembler kursumuzun ilk konusu burada bitti. Bir sonraki sayıda komutları alfabetik sırayla görmeye başlayacağız.

BELLEK HARİTASI – 1

Yeni bir bölüme başlıyoruz. Bu bölümde COMMODORE 64'ün bellek haritasını yani hangi adreslerde ne var, ne işe yarar gibi bilgileri bu bölümde göreceğiz. Bu bölümdeki yazılanlar PROGRAMCININ EL KİTABI'ndan alınmıştır.

İSİM	HEX ADRES	ONLU ADRES	AÇIKLAMA
D6510	00	0	6510 üzerindeki veri yön registeri. 1 nolu adres için giriş çıkış yön tanımlama için kullanılır.
R6510	01	1	6510 üzerindeki veri registeri. Çeşitli bellek konfigürasyonları amacıyla kullanılır.
	02	2	Kullanılmıyor. Makina dili programlarınızda kullanabilirsiniz.
ADRAY1	03 – 04	3 - 4	SIÇRAMA VEKTÖRÜ : Ondalık sayı – Tamsayı dönüşümü.
ADRAY2	05 – 06	5 - 6	SIÇRAMA VEKTÖRÜ : Tamsayı – Ondalık sayı dönüşümü.
CHARAC	07	7	Karakter arama.
ENDCHR	08	8	BAYRAK : Yazısal dizinin sonunu gösteren tırnağın aranması.
TRMPOS	09	9	Ekran kolonunun son TAB pozisyonundan uzaklığı.
VERCK	0A	10	BAYRAK : 0 = Yükleme 1 = Verify.
COUNT	0B	11	Giriş tamponu göstergesi / İndis sayısı.
DIMFLAG	0C	12	BAYRAK : Default dizi boyutu.
VALTYP	0D	13	VERİ TİPİ : \$FF = Yazısal dizi \$00 = Sayısal dizi.
INTFLG	0E	14	VERİ TİPİ : \$80 = Tamsayı \$00 = Ondalık sayı.
GARBFL	0F	15	BAYRAK : DATA tarama /LIST için tırnak işareti / hafıza boşaltma.
SUBFLG	10	16	BAYRAK : İndis referansı / Kullanıcı fonksiyonu çağırısı.
INPFLG	11	17	BAYRAK : \$00 = INPUT \$40 = GET \$98 = READ
TANSNG	12	18	BAYRAK : TAN işareti / Karşılaştırma sonucu.
	13	19	BAYRAK : INPUT uyarısı.
LINNUM	14 - 15	20 - 21	Geçici adres. Tamsayı değeri.
TEMPPT	16	22	GÖSTERGEÇ : Geçici yazısal dizi yığını.
LASTPT	17 -18	23 - 24	Son yazısal dizi adresi.
TEMPST	19 - 21	25 - 33	Geçici yazısal diziler için STACK (yığın).
INDEX	22 - 25	34 – 37	Yardımcı göstergeç alanı.
RESHO	26 – 2A	38 - 42	Çarpma işleminin ondalık sayı sonucu.
TXTTAB	2B – 2C	43 - 44	GÖSTERGEÇ : BASIC metninin başlangıcı.
VARTAB	2D – 2E	45 - 46	GÖSTERGEÇ : BASIC değişkenlerinin başlangıcı.
ARYTAB	2F - 30	47 - 48	GÖSTERGEÇ : BASIC dizilerinin başlangıcı.
STREND	31 - 32	49 - 50	GÖSTERGEÇ : BASIC dizilerinin sonu (+1)
FRETOP	33 - 34	51 - 52	GÖSTERGEÇ : Dizi saklanımının sonu.
FRESPC	35 - 36	53 - 54	Yardımcı dizi göstergesi.
MEMSIZ	37 - 38	55 - 56	GÖSTERGEÇ : BASIC'ın kullandığı en yüksek adres.
CURLIN	39 – 3A	57 - 58	Bulunduğunuz BASIC satır numarası.
OLDLIN	3B – 3C	59 – 60	Bir evvelki BASIC satır numarası
OLDTXT	3D – 3E	61 - 62	GÖSTERGEÇ : CONT için BASIC yönergesi.
DATLIN	3F – 40	63 - 64	O anki DATA satırının nosu.
DATPTR	41 - 42	65 - 66	GÖSTERGEÇ : Bulunduğunuz DATA adresi.
INPPTR	43 - 44	67 - 68	VEKTÖR : INPUT rutini.
VARNAM	45 - 46	69 - 70	O andaki BASIC değişkeninin ismi.
VARPNT	47 - 48	71 - 72	GÖSTERGEÇ : O andaki BASIC değişkeni verisi.
FORPNT	49 – 4A	73 - 74	GÖSTERGEÇ : FOR / NEXT için indis değişkeni.
	4B - 60	75 - 96	Geçici göstergeç / Veri alanı.
FACEXP	61	97	Ondalık sayı akümülatörü 1 : ÜS
FACHO	62 - 65	98 - 101	Ondalık sayı akümülatörü 1 : TABAN
FACSGN	66	102	Ondalık sayı akümülatörü 1 : İŞARET
SGNFLG	67	103	GÖSTERGEÇ: Serileri değerlendirme sabiti.
BITS	68	104	ONDALIK AKÜMÜLATÖR 1 : Taşma hanesi.

ARGEXP	69	105	ONDALIK AKÜMÜLATÖR 2 : Üs.
ARGHO	6A – 6D	106 - 109	ONDALIK AKÜMÜLATÖR 2 : Taban.
ARGSGN	6E	110	ONDALIK AKÜMÜLATÖR 2 : İşaret.
ARISGN	6F	111	İşaretlerin karşılaştırılmasının sonucu : Akü 1 ve Akü 2
FACOV	70	112	ONDALIK AKÜMÜLATÖR 1 : Alt basamak (Yuvarlayarak).
FBUFPT	71 - 72	113 - 114	GÖSTERGEÇ : Teyp tamponu.
CHRGET	73 – 7A	115 - 138	ALTPROGRAM: BASIC metninden bir sonra gelen baytın alınması.
CHRGOT	79	121	Metindeki aynı baytın tekrar alınması.
TXTPTR	7A – 7B	122 -123	GÖSTERGEÇ : BASIC metninin o andaki baytı.
RNDX	8B – 8F	139 - 143	Ondalıklı RND fonksiyonunun değeri.
STATUS	90	144	Kernal G / Ç Statü sözcüğü : ST
STKEY	91	145	BAYRAK : STOP tuşu / RVS tuşu
SVXT	92	146	Teyp için zamanlama sabiti.
VERCK	93	147	BAYRAK : 0 = Yükleme / 1 = Verify
C3PO	94	148	BAYRAK : Seri bağlantı – Tamponlanmış çıkış karakteri.
BSOUR	95	149	Seri bağlantı için tamponlanmış karakter.
SYNO	96	150	Kaset eşzaman sayısı.
	97	151	Geçici veri alanı.
LDTND	98	152	Açık dosya sayısı / Dosya tablosu için indeks.
DFLTN	99	153	Default giriş cihazı (0).
DFLTO	9A	154	Default çıkış (CMD) cihazı (3).
PRTY	9B	155	Teyp karakter eşliği (parity).
DPSW	9C	156	BAYRAK : Kasetten bayt alındı.
MSGFLG	9D	157	BAYRAK : \$80 = Doğrudan mod \$00 = Program.
PTR1	9E	158	Teyp geçişi 1 hata günlüğü.
PTR2	9F	159	Teyp geçişi 2 hata günlüğü.
TIME	A0 – A2	160 - 162	Gerçek zaman jiffy saati (yaklaşık) 1/60 saniye.
	A3 – A4	163 - 164	Geçici data alanı.
CNTDN	A5	165	Eşzamanlı kaset geri sayımı.
BUFPT	A6	166	GÖSTERGEÇ : Teyp Giriş /Çıkış tamponu.
INBIT	A7	167	RS-232 Giriş bit'leri / Kaset için geçici bölge.
BITCI	A8	168	RS-232 Giriş bit'i sayımı / Kaset için geçici bölge.
RINONE	A9	169	RS-232 BAYRAK : Başlangıç biti kontrolü.
RIDATA	AA	170	RS-232 Giriş bayt tamponu / Kaset için geçici bölge.
RIPRTY	AB	171	RS-232 Giriş eşliği / Kaset short cnt.
SAL	AC -AD	172 - 173	GÖSTERGEÇ : Teyp tamponu / Ekran kayması.
EAL	AE - AF	174 - 175	Teyp sonu adresi / Program sonu.
CMPO	B0 – B1	176 - 177	Teyp zamanlama sabitleri.
TAPE1	B2 – B3	178 - 179	GÖSTERGEÇ : Teyp tamponunun başlangıcı.
BITTS	B4	180	RS-232 Çıkış bit'i sayımı / Kaset için geçici bölge.
NXTBIT	B5	181	RS-232'de bir sonra gönderilecek bit / Teyp EOF bayrağı.
RODATA	B6	182	RS-232 Çıkış bayt tamponu.
FNLEN	B7	183	İşlemdeki dosya ismi uzunluğu.
LA	B8	184	İşlemdeki dosya numarası.
SA	B9	185	İşlemdeki ikincil adres.
FA	BA	186	İşlemdeki aygıt numarası.
FNADR	BB - BC	187 - 188	GÖSTERGEÇ : İşlemdeki dosya ismi.
ROPRTY	BD	189	RS-232 Çıkış eşliği / Kaset için geçici bölge.
FSBLK	BE	190	Kaset okuması / Yazma bloğu sayma.
MYCH	BF	191	Seri sözcük tamponu.
CAS1	C0	192	Teyp motoru kilitlemesi.
STAL	C1 – C2	193 - 194	G / Ç başlangıç adresi.
MEMUSS	C3 – C4	195 - 196	Teyp yüklemesi için geçici bölge.
LSTX	C5	197	Basılı tuş : CHR\$(n) = 0 Tuşa basılmamış.
NDX	C6	198	Klavye tamponunda bulunan karakter sayısı.
RVS	C7	199	BAYRAK : Negatif gösterim 1 = EVET 0 = HAYIR.
INDX	C8	200	GÖSTERGEÇ : INPUT için mantıksal satır sonu.
LXSP	C9 - CA	201 - 202	INPUT başlangıcında kursörün X – Y konumu.

SFDX	CB	203	BAYRAK : SHIFT ile karakter yazımı.
BLNSW	CC	204	BAYRAK : Kursör yanıp sönebilir 0 = Parlayan kursör.
BLNCNT	CD	205	ZAMANLAYICI : Kursörü yakıp söndürmek için geri sayım.
GDBLN	CE	206	Kursörün üzerinde bulunduğu karakter.
BLNON	CF	207	BAYRAK : Kursörün son yanıp sönmeye hali (On/Off).
CRSW	D0	208	BAYRAK : Klavyeden INPUT veya GET.
PNT	D1 – D2	209 - 210	GÖSTERGEÇ : Anlık ekran satırı.
PNTR	D3	211	İşlemdeki satırda kursörün bulunduğu sütun.
QTSW	D4	212	BAYRAK : Tırnak modunda editör. \$00 = HAYIR
LNMX	D5	213	Ekran satırının fiziksel uzunluğu.
TBLX	D6	214	Kursörün bulunduğu satırın fiziksel satır numarası.
	D7	215	Geçici veri alanı.
INSRT	D8	216	BAYRAK : Araya karakter yerleştirme modu. >0 yerleştirme sayısı.
LDTB1	D9 – F2	217 - 242	Ekran satırı bağlantı tablosu / Editör için geçici bölge.
USER	F3 – F4	243 - 244	GÖSTERGEÇ : Anlık ekran rengi RAM yeri.
KEYTAB	F5 – F6	245 - 246	VEKTÖR : Klavye çözüm tablosu.
RIBUF	F7 – F8	247 - 248	RS-232 Giriş tampon göstergesi.
ROBUF	F9 - FA	249 - 250	RS-232 Çıkış tampon göstergesi.
FREKZP	FB - FE	251 - 254	Kullanıcı programları için boş ZERO PAGE alanı.
BASZPT	FF	255	BASIC geçici veri alanı.
	0100-01FF	256 - 511	Mikroişlemci sistem yığın alanı.
	0100-010A	256 - 266	Ondalıktan dizgiye çalışma alanı.
BAD	0100-013E	256 - 318	Teyp girişi hata kaydı.
BUF	0200-0258	512 - 600	Sistem giriş tamponu.
LAT	0259-0262	601 - 610	KERNAL TABLOSU : Aktif mantıksal dosya sayısı.
FAT	0263-026C	611 - 620	KERNAL TABLOSU : Her dosya için cihaz sayısı.
SAT	026D-0276	621 - 630	KERNAL TABLOSU : Her dosya için ikincil adres.
KEYD	0277-0280	631 - 640	Klavye tampon kuyruğu (FIFO)
MEMSTR	0281-0282	641 - 642	GÖSTERGEÇ : BASIC için belleğin taban adresi.
MEMSIZ	0283-0284	643 - 644	GÖSTERGEÇ : BASIC için belleğin tavan adresi.
TIMOUT	0285	645	BAYRAK : IEEE zaman aşımı için KERNAL değişkeni.
COLOR	0286	646	Anlık karakter renk kodu.
GDCOL	0287	647	Kursörün üzerinde bulunduğu fon rengi.
HIBASE	0288	648	Ekran belleğinin tavanı (sayfa)
XMAX	0289	649	Klavye tamponunun büyüklüğü.
RPTFLG	028A	650	BAYRAK : Tuşlar için tekrarlama fonksiyonu. \$80 = Tekrarla.
KOUNT	028B	651	Hız sayacı tekrarlama.
DELAY	028C	652	Tekrarlamayı geciktirme hız sayacı.
SHFLAG	028D	653	BAYRAK : Klavye SHIFT tuşu / CTRL tuşu / C= tuşu.
LSTSHF	028E	654	Son klavye SHIFT görüntüsü.
KEYLOG	028F-0290	655 - 656	VEKTÖR : Klavye tablosu durumu.
MODE	0291	657	BAYRAK : \$00 = SHIFT tuşu işlem dışı \$80 = SHIFT tuşu devrede.
AUTODN	0292	658	BAYRAK : Otomatik aşağı kayma. 0 = ON
M51CTR	0293	659	RS-232 : Kontrol registeri görüntüsü.
M51CDR	0294	660	RS-232 : Komut registeri görüntüsü.
M51AJB	0295-0296	661 - 662	RS-232 : Standart olmayan BPS (Zaman / 2 - 100) USA
RSSTAT	0297	663	RS-232 : 6551 Durum registeri görüntüsü.
BITNUM	0298	664	RS-232 : Gönderilecek bit sayısı.
BAUDOF	0299-029A	665 - 666	RS-232 : Baud hızı. Tam bit süresi (µs)
RIDBE	029B	667	Giriş tamponunun sonunu gösteren RS-232 indeksi.
RIDBS	029C	668	RS-232 : Giriş tamponu başlangıcı (sayfa)
RODBS	029D	669	RS-232 : Çıkış tamponu başlangıcı.
RODBE	029E	670	Çıkış tamponu sonuna kadar RS-232 indeksi.
IRQTMP	029F-0300	671 - 672	Teyp Giriş / Çıkış süresince IRQ vektörünü saklama yeri.
ENABL	02A1	673	RS-232 çalışır durumda.
	02A2	674	Kaset Giriş / Çıkış sırasında TOD duyumu.
	02A3	675	Kaset okuması için geçici saklanım.
	02A4	676	Kaset okuması için geçici D1 IRQ

	02A5	677	Satır indeksi geçici bölge.
	02A6	678	BAYRAK : 0 = NTSC 1 = PAL
	02A7-02FF	679 - 767	Kullanılmıyor.
IERROR	0300-0301	768 - 769	VEKTÖR : BASIC hata uyarısı yazılır.
IMAIN	0302-0303	770 - 771	VEKTÖR : BASIC Warm Start.
ICRNCH	0304-0305	772 - 773	VEKTÖR : BASIC tekstini kodla.
IQPLOP	0306-0307	774 - 775	VEKTÖR : BASIC tekst listesi.
IGONE	0308-0309	776 - 777	VEKTÖR : BASIC karakter gönderimi.
IEVAL	030A-030B	778 - 779	VEKTÖR : BASIC token değerlendirmesi.
SAREG	030C	780	6502 için A registeri saklama adresi.
SXREG	030D	781	6502 için X registeri saklama adresi.
SYREG	030E	782	6502 için Y registeri saklama adresi.
SPREG	030F	783	6502 için SP (Stack Pointer) saklama adresi.
USRPOK	0310	784	USR işlevi JUMP komutu (\$4C)
USRADD	0311-0312	785 - 786	USR adresi alt-bayt / üst-bayt.
	0313	787	Kullanılmıyor.
CINV	0314-0315	788 - 789	VEKTÖR : Donanım IRQ interrupt.
CBINV	0316-0317	790 - 791	VEKTÖR : BRK komutu interrupt.
NMINV	0318-0319	792 - 793	VEKTÖR : Engellenemez interrupt.
IOPEN	031A-031B	794 - 795	KERNAL OPEN rutini vektörü.
ICLOSE	031C-031D	796 - 797	KERNAL CLOSE rutini vektörü.
ICHKIN	031E-031F	798 - 799	KERNAL CHKIN rutini vektörü.
ICKOUT	0320-0321	800 - 801	KERNAL CHKOUT rutini vektörü.
ICLRCH	0322-0323	802 - 803	KERNAL CLRCHN rutini vektörü.
IBASIN	0324-0325	804 - 805	KERNAL CHRIN rutini vektörü.
IBSOUT	0326-0327	806 - 807	KERNAL CHROUT rutini vektörü.
ISTOP	0328-0329	808 - 809	KERNAL STOP rutini vektörü.
IGETIN	032A-032B	810 - 811	KERNAL GETIN rutini vektörü.
ICLALL	032C-032D	812 - 813	KERNAL CLALL rutini vektörü.
USRCMD	032E-032F	814 - 815	Kullanıcı tanımlı vektör.
ILOAD	0330-0331	816 - 817	KERNAL LOAD rutini vektörü.
ISAVE	0332-0333	818 - 819	KERNAL SAVE rutini vektörü.
	0334-033B	820 - 827	Kullanılmıyor.
TBUFFR	033C-03FB	828 - 1019	Teyp Giriş / Çıkış tamponu.
	03FC-03FF	1020-1023	Kullanılmıyor.
VICSCN	0400-07FF	1024-2047	1024 bayt ekran bellek alanı.
	0400-04E7	1024-2023	Video matrisi : 25 Satır * 40 Kolon.
	07E8-07F7	2023-2039	Kullanılmıyor.
	07F8-07FF	2040-2047	Sprite data göstergeçleri.
	0800-9FFF	2048-40959	Normal BASIC programlama alanı.
	8000-9FFF	32768-40959	Kartuş bölgesi. 8K ROM.
	A000-BFFF	40960-49151	BASIC ROM 8K. (Veya 8K RAM. (Veya 8K KARTUŞ ROM))
	C000-CFFF	49152-53247	RAM 4K.
	D000-DFFF	53248-57343	GİRİŞ / ÇIKIŞ cihazları ve RENK RAM'ı (Veya 4K karakter ROM. (Veya 4K RAM))
	E000-FFFF	57344-65535	KERNAL ROM 8K. (Veya 8K RAM. (Veya 8K KARTUŞ ROM.))

Bellek haritasının birinci bölümü burada sona eriyor.. İkinci bölümde \$D000 - \$DFFF arasındaki I/O (Giriş/Çıkış) bölgesini ve 1. Nolu adresi ayrıntılı olarak göreceğiz.

MAKİNA DİLİ EDITÖRÜ - MDE

Dergide vereceğimiz programları özelliklerini daha sonra açıklayacağımız aşağıdaki MDE isimli programla girmeniz gerekmektedir. Fakat program çok uzun olduğu için DATA satırları çok dikkatli girilmelidir.

Programın listesi Türkçe COMMODORE dergisinin MART 1986 tarihli 1. sayısının PROGRAM DÖKÜM EKİ'nden yazılmıştır. Yazım sırasında küçük "L" harfi ile "1" sayısına dikkat edin.

```
0 dimh (75) : for i=0 to 9
1 h(48+i) : h(65+i)=i+10 : next
2 for i=49152 to 50911 : read a$
3 h=asc(left$(a$,1)) : l=asc(right$(a$,1))
4 d=h(h)*16+h(l) : s=s+d : poke i,d
5 a=a+1:if a<20 then next : a=-1
6 print "satir:"1000+z;
7 read v : z=z+1 : if v=s then 9
8 print "toplama hatası ! satir : "999+z : stop
9 if a<0 then sys 50865
10 s=0 : a=0 : print : next : sys 50865
1000 data 00,0b,08,0a,00,9e,32,30,36,31,00,00,00,a2,08,a9,36,85,a4,a9, 1247
1001 data 08,85,a5,a9,00,85,a6,a9,b0,85,a7,a0,00,b1,a4,91,a6,c8,d0,f9, 2888
1002 data e6,a5,e6,a7,ca,d0,f2,a9,36,85,01,4c,00,b0,20,d1,b1,a9,06,8d, 2787
1003 data 21,d0,a9,03,8d,20,d0,8d,86,02,a0,b3,a9,74,20,ff,b1,a0,b3,a9, 2667
1004 data b9,20,ff,b1,a0,00,20,cf,ff,99,01,02,c8,c8,0d,d0,f5,88,f0,d2, 2912
1005 data c0,0f,90,02,a0,0e,8c,00,02,20,ea,b1,a0,b3,a9,cf,20,ff,b1,20, 2323
1006 data 8e,b4,85,fc,85,62,20,8e,b4,85,fb,85,61,20,a7,b4,d0,20,a0,b3, 2864
1007 data a9,e5,20,ff,b1,20,8e,b4,85,60,20,8e,b4,85,5f,20,a7,b4,d0,0a, 2624
1008 data a5,61,c5,5f,a5,62,e5,60,90,06,20,43,b3,4c,3a,b0,a9,aa,a0,00, 2379
1009 data 91,fb,e6,fb,d0,02,e6,fc,20,3f,b2,90,ef,4c,fb,b4,a2,02,86,58, 3118
1010 data a9,a6,a0,9d,20,f2,b1,20,e4,ff,f0,fb,c9,30,90,0c,c9,47,b0,08, 2970
1011 data c9,3a,90,0b,c9,41,b0,07,c9,14,d0,0f,4c,0b,b1,20,d2,ff,a6,58, 2322
1012 data 95,f7,c6,58,d0,d2,60,ae,8d,02,f0,26,c9,0c,d0,03,4c,0b,b6,c9, 2685
1013 data 13,d0,03,4c,8b,b5,c9,0d,d0,03,4c,ba,b4,c9,10,d0,03,4c,68,b5, 2282
1014 data c9,0e,d0,06,20,5f,b4,4c,64,b1,4c,92,b0,a5,f9,20,02,b1,0a,0a, 2132
1015 data 0a,0a,85,f9,a5,f8,20,02,b1,05,f9,60,c9,3a,90,02,69,08,29,0f, 1950
1016 data 60,a6,59,e0,08,90,1f,a6,58,e0,02,b0,06,20,d2,ff,4c,8e,b0,c6, 2509
1017 data 59,a0,14,a9,92,20,f2,b1,ca,d0,fa,84,57,68,68,4c,8b,b1,a6,d3, 2891
1018 data e0,08,b0,03,4c,92,b0,20,d2,ff,a6,58,e0,02,90,09,c6,59,20,d2, 2468
1019 data ff,c6,58,d0,f9,4c,8e,b0,48,4a,4a,4a,4a,20,59,b1,68,29,0f,c9, 2419
1020 data 0a,90,02,69,06,69,30,4c,d2,ff,a2,fc,9a,20,d1,b1,20,48,b2,20, 2261
1021 data ea,b1,20,9f,b2,a5,fc,20,4e,b1,a5,fb,20,4e,b1,20,ed,b1,a9,3a, 2860
1022 data a0,20,20,f2,b1,a9,00,85,59,20,8e,b0,20,ed,b1,a4,59,20,ef,b0, 2530
1023 data 91,fb,c8,84,59,c0,08,90,ec,20,10,b2,a9,12,20,d2,ff,20,8e,b0, 2657
1024 data 20,ef,b0,c5,ff,f0,0d,20,43,b3,a9,14,a0,14,20,f2,b1,4c,a2,b1, 2665
1025 data a9,92,20,d2,ff,20,33,b2,20,e0,b2,20,3f,b2,90,9f,4c,8b,b5,a9, 2648
1026 data 93,20,d2,ff,a2,00,a9,03,9d,00,d8,9d,00,d9,9d,00,da,9d,00,db, 2476
1027 data e8,d0,ef,60,a9,0d,2c,a9,20,4c,d2,ff,20,d2,ff,98,4c,d2,ff,20, 2965
1028 data e4,ff,f0,fb,60,84,5d,85,5c,a0,00,b1,5c,f0,06,20,d2,ff,c8,d0, 3100
1029 data f6,60,a5,fb,85,5a,a0,00,84,5b,b1,fb,18,65,5a,85,5a,90,02,e6, 2606
1030 data 5b,06,5a,26,5b,c8,c0,08,90,ec,a5,5a,65,5b,85,ff,60,18,a5,fb, 2467
1031 data 69,08,85,fb,90,02,e6,fc,60,a5,fb,c5,5f,a5,fc,e5,60,60,a0,b3, 3106
1032 data a9,fb,20,ff,b1,a0,01,b9,00,02,20,d2,ff,cc,00,02,c8,90,f4,a9, 2692
1033 data 10,ed,00,02,aa,20,ed,b1,ca,d0,fa,a5,62,20,4e,b1,a5,61,20,4e, 2453
1034 data b1,20,ed,b1,a5,60,20,4e,b1,a5,5f,20,4e,b1,a9,9f,20,d2,ff,20, 2575
1035 data ea,b1,24,5e,10,01,60,a9,12,20,d2,ff,a2,28,20,ed,b1,ca,d0,fa, 2646
1036 data a9,92,4c,d2,ff,a5,d6,c9,16,b0,01,60,a9,a0,85,a4,a9,78,85,a6, 2945
1037 data a9,04,85,a5,85,a7,a2,13,a0,27,b1,a4,91,a6,88,10,f9,ca,f0,19, 2671
1038 data 18,a5,a4,69,28,85,a4,90,02,e6,a5,18,a5,a6,69,28,85,a6,90,e0, 2503
```

```

1039 data e6,a7,4c,b6,b2,a9,91,4c,d2,ff,a9,0f,8d,18,d4,a9,00,8d,05,d4, 2776
1040 data a9,f7,8d,06,d4,a9,11,8d,04,d4,a9,32,8d,01,d4,a9,00,8d,00,d4, 2413
1041 data a0,80,20,09,b3,a9,10,8d,04,d4,60,a2,ff,ca,d0,fd,88,d0,f8,60, 2914
1042 data a9,0f,8d,18,d4,a9,2d,8d,05,d4,a9,a5,8d,06,d4,a9,21,8d,04,d4, 2385
1043 data a9,07,8d,01,d4,a9,05,8d,00,d4,a0,ff,20,09,b3,a9,20,8d,04,d4, 2250
1044 data a9,00,8d,01,d4,8d,00,d4,60,38,20,f0,ff,8a,48,98,48,18,a0,06, 2179
1045 data a2,18,20,f0,ff,a0,b4,a9,0a,20,ff,b1,20,12,b3,20,e4,ff,f0,fb, 2931
1046 data a2,1d,a9,14,20,d2,ff,ca,d0,fa,68,a8,68,aa,18,4c,f0,ff,0d,0d, 2704
1047 data 0d,20,20,20,20,20,20,20,20,20,20,4d,41,4b,49,4e,41,20,44,49, 939
1048 data 4c,49,20,45,44,49,54,4f,52,55,20,20,20,20,0d,0d,20,20,20,20, 1003
1049 data 20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20, 640
1050 data 20,20,20,20,20,20,00,0d,0d,0d,20,20,20,50,52,4f,47,52,41,4d, 863
1051 data 20,41,44,49,20,20,3a,20,00,0d,0d,20,20,20,42,41,53,4c,41,4e, 947
1052 data 47,49,43,20,41,44,2e,3a,20,24,00,0d,0d,20,20,20,42,49,54,49, 966
1053 data 53,20,41,44,52,45,53,49,20,3a,20,24,00,92,05,20,50,52,4f,47, 1208
1054 data 52,41,4d,20,20,3a,20,00,12,20,20,2a,2a,2a,20,20,20,59,41,4e, 914
1055 data 4c,49,53,60,56,45,52,49,20,20,20,2a,2a,2a,20,20,92,00,0d,0d, 1096
1056 data 2a,2a,2a,20,53,4f,4e,20,2a,2a,2a,20,00,13,05,20,20,12,44,92, 908
1057 data 49,53,4b,20,56,45,59,41,20,12,54,92,45,59,50,0d,00,13,20,20, 1186
1058 data 49,2f,4f,20,2d,20,48,41,54,41,53,49,00,20,d1,b1,20,48,b2,a0, 1610
1059 data b3,a9,cf,20,ff,b1,20,8e,b4,85,fc,20,8e,b4,85,fb,c5,61,a5,fc, 3207
1060 data e5,62,90,23,a5,fb,c5,5f,a5,fc,e5,60,b0,19,20,a7,b4,d0,14,60, 2860
1061 data 20,a7,b4,f0,0c,85,f9,20,a7,b4,f0,05,85,f8,4c,ef,b0,68,68,20, 2749
1062 data 43,b3,4c,5f,b4,20,cf,ff,c9,4c,d0,09,20,d1,b1,20,48,b2,4c,0b, 2372
1063 data b6,c9,0d,60,a9,00,85,5e,20,5f,b4,20,ea,b1,20,0d,b5,24,5e,30, 2042
1064 data 05,20,e4,ff,f0,fb,20,e1,ff,f0,26,20,9f,b2,24,5e,10,09,20,4e, 2435
1065 data b5,20,0d,b5,20,60,b5,20,33,b2,20,3f,b2,90,d7,a0,b4,a9,28,20, 2190
1066 data ff,b1,20,e4,ff,c9,0d,d0,f9,a9,00,85,5e,a5,61,85,fb,a5,62,85, 3056
1067 data fc,20,e0,b2,4c,64,b1,a5,fc,20,4e,b1,a5,fb,85,ff,20,4e,b1,a9, 3003
1068 data 20,a0,3a,20,f2,b1,a0,00,20,ed,b1,b1,fb,20,4e,b1,c8,c0,08,90, 2566
1069 data f3,20,ed,b1,24,5e,30,03,a9,12,2c,a9,20,20,d2,ff,20,10,b2,a5, 2190
1070 data ff,20,4e,b1,a9,92,20,d2,ff,4c,ea,b1,a9,ff,85,b8,85,b9,a9,04, 3073
1071 data 85,ba,20,c0,ff,a2,ff,4c,c9,ff,20,cc,ff,a9,ff,4c,c3,ff,20,5f, 3315
1072 data b4,a9,80,85,5e,20,4e,b5,20,48,b2,a2,24,a9,2d,20,d2,ff,ca,d0, 2596
1073 data fa,20,ea,b1,20,ea,b1,20,60,b5,4c,c1,b4,20,b8,b5,a6,5f,a4,60, 2812
1074 data a9,61,20,d8,ff,b0,0a,20,b7,ff,29,bf,d0,03,4c,fb,b4,a9,01,20, 2577
1075 data c3,ff,20,68,b6,a0,b4,a9,4f,20,ff,b1,20,f9,b1,4c,fb,b4,20,68, 2921
1076 data b6,a9,37,a0,b4,20,ff,b1,20,f9,b1,a2,08,c9,44,f0,06,a2,01,c9, 2717
1077 data 54,d0,f1,a9,01,a8,20,ba,ff,a0,00,e0,01,f0,1a,a9,40,8d,20,02, 2403
1078 data a9,3a,8d,21,02,b9,01,02,99,22,02,c8,cc,00,02,90,f4,c8,c8,d0, 2182
1079 data 0c,b9,01,02,99,20,02,c8,cc,00,02,d0,f4,98,a2,20,a0,02,4c,bd, 2018
1080 data ff,20,b8,b5,a5,ba,c9,08,90,33,a6,b9,86,57,a9,01,20,c3,ff,a9, 2800
1081 data 60,85,b9,20,c0,ff,b0,28,a5,ba,20,b4,ff,a5,b9,20,96,ff,20,a5, 2911
1082 data ff,85,61,a5,90,4a,4a,b0,13,20,a5,ff,85,62,20,ab,ff,a5,57,85, 2663
1083 data b9,a9,00,20,d5,ff,90,03,4c,a3,b5,86,5f,84,60,a5,ba,c9,01,d0, 2639
1084 data 0a,ad,3d,03,85,61,ad,3e,03,85,62,4c,fb,b4,a9,13,20,d2,ff,a2, 2300
1085 data 1c,20,ed,b1,ca,d0,fa,60,aa,aa,aa,aa,46,ad,00,c0,8d,00,ff,ee, 2732
1086 data b2,c6,ee,b5,c6,d0,f2,ee,b3,c6,ee,b6,c6,ad,b6,c6,c9,0f,d0,e5, 3882
1087 data a9,b0,85,2d,85,2f,85,31,a9,0e,85,2e,95,30,85,32,4c,74,a4,aa, 2137

```

MAKİNE DİLİ EDITÖRÜNÜ YAZMA

Oldukça uzun olan bu programı hatasız olarak yazdıktan sonra mutlaka kasete veya diskete kaydedin. Çünkü programı çalıştırdığınızda, eğer tüm DATA satırlarını doğru olarak yazmışsanız, program kendi kendisini silerek yerine asıl MDE programını oluşturacaktır.

Şimdi RUN komutunu vererek programı çalıştırın. Ekranda hangi data satırının okunduğu gösterilecektir. Eğer hatalı bir satır varsa, satır numarası belirtilecektir. Hatalı satırı düzelttikten sonra yeniden RUN komutu verin. Tüm DATA satırları okunduktan sonra bilgisayar işini bitirecektir. Artık oluşan asıl MDE programını SAVE "MDE" ile kasete veya SAVE "MDE",8 ile diskete kaydedin.

Makina Dili Editörü (kısaca MDE), makina dilinde yazılmış programları daha kolay yazabilmeniz için hazırlanmış bir yardımcı programdır. MDE ile uzun DATA satırları yazmaktan kurtulacaksınız. Üstelik programlarınız kesinlikle çalışacaktır. Bunu mümkün kılan, her satırın sonundaki bulunan bir kontrol toplamı (CHECKSUM) 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 üstünde hazır olduğunu kabul edelim. LOAD "MDE",8 veya LOAD "MDE",1 ile kullandığınız kayıt aracından MDE'yi bilgisayara yükleyin ve RUN komutuyla çalıştırın. Ekran renkleri değişecek ve MDE sizi açılış ekranı ile selemleyecektir. İlk olarak size yazılacak programın adı sorulacaktır. Bu soruya karşılık olarak, dergideki program listesinin ilk satırındaki "PROGRAM ADI" olarak belirtilen adı yazın. Şimdi de programın bilgisayarın iç belleğinde hangi adresten başlamsı gerektiği sorulacaktır. Buna da listenin ilk satırında verilen iki sayıdan birincisini verin. Örneğin 0801 gibi. Bir sonraki soru ise, programın bitiş adresini soracaktır. Buna da ilk satırdaki ikinci sayı ile cevap verin. Örneğin 16f7 gibi.

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 sizden 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 yazacak ve yeni satırı girmenizi bekleyecektir. Ancak bir satırı girerken bir hata yaptıysanız veya kontrol toplamın yanlış yazdıysanız dikkat çeken bir ses çıkacak 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 sesler sadece dergiye ve klavyeye bakarak yazmanızı sağlayacaktır.

Programın en son satırını da yazdığınız zaman MDE programı nereye kaydedeceğinizi soracaktır. Ekranın sol üst köşesinde belirecek bu soruya D (disket için) veya T (teyp için) ile cevap verin. SAVE işlemi bitince bilgisayar yeniden başlangıç adresini vererek başa dönecektir. Artık RUN/STOP RESTORE ile MDE'den çıkarak, yazmış olduğunuz programı yükleyip çalıştırabilirsiniz.

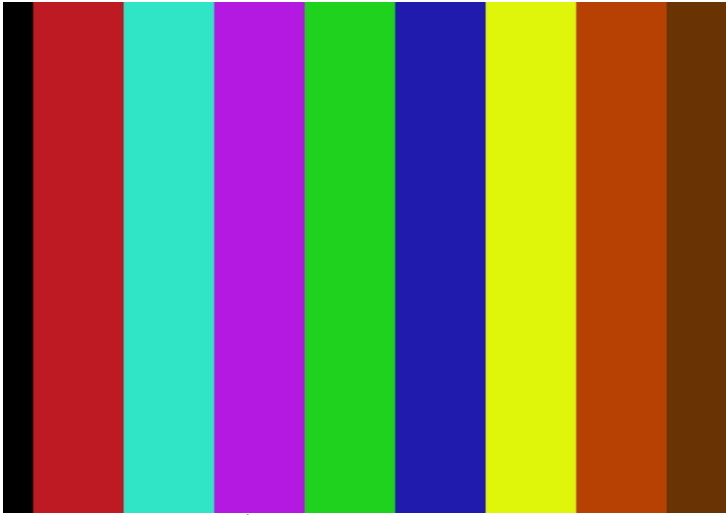

MDE'nin bazı özellikleri de vardır. Şimdi de bunları görelim.

Diyelimki uzun bir program var ve yazarken birden işiniz çıktı gitmeniz gerekiyor. Böyle durumlarda istediğiniz zaman **CTRL** ve **S** tuşlarına basarak SAVE edebilirsiniz. Çıkan soruya D veya T ile cevap vererek diğer işlerinizle ilgilenebilirsiniz. Başka bir zaman yazmaya devam etmek istediğinizde MDE'yi yükleyin ve çalıştırın. Programın adını yazın ve başlangıç adresi olarak "L" harfini verin. Çıkan soruya uygun yanıt vererek yarım kalan programınızı yükleyin. Yükleme bittiğinde başlangıç ve bitiş adresleri hesaplanacaktır.

Şimdi de **CTRL** ve **M** tuşlarına basın. Çıkacak soruya ekranın üstündeki başlangıç adresi ile cevap verin. Daha sonra BOŞLUK tuşu ile programı yazarken bıraktığınız satırı bulun. Yazmayı bıraktığınız satır ve devamı **AA** kodu ile dolu olacaktır (kontrol toplamı hariç). O satırın numarasını not alın. Şimdi RUN/STOP tuşuna basın ve işlemi durdurun. **CTRL** ve **N** tuşlarına basarak yeni başlangıç adresi olarak not aldığınız adresi girin. Yazmaya kaldığınız yerden devam edebilirsiniz. Bir başka komut ise **CTRL P**'dir. Bu komutla herhangi bir programı MDE şeklinde bir printere yazdırabilirsiniz. BuNun için önce MDE'yi çalıştırın ve program izmini yazın. Başlangıç adresi olarak **L** harfini girin. Yükleme bitince **CTRL** ve **P** tuşlarına basarak yazma işlemini başlatın.

MDE'yi yazmak isteyenlere şimdiden kolay gelsin.

PROGRAM KÖŞESİ

VERTİCAL RASTER BAR 1	
 VERTİCAL RASTER BAR 1	<div>TEKRAR</div> <div>*= SEI LDA #S00 STA SD011 LDA #S00 STA SD020 LDA #S02 STA SD020 LDA #S03 STA SD020 LDA #S04 STA SD020 LDA #S05 STA SD020 LDA #S06 STA SD020 LDA #S07 STA SD020 LDA #S08 STA SD020 LDA #S09 STA SD020 LDA #S0A STA SD020 JMP TEKRAR</div>
VERTİCAL RASTER BAR 2	
 VERTİCAL RASTER BAR 2	<div>TEKRAR</div> <div>*= SEI LDA #S00 STA SD011 LDA #S00 LDX #S07 LDY #S02 STA SD020 STX SD020 STY SD020 JMP TEKRAR</div>

Son anda eklediğim bir bölüm. Bu iki ekran görüntüsünü <http://hades6510.sitemynet.com> sitesindeki yazılım bölümünden hatırlayacaksınız. Bu bölümde fazla uzun olmayan programlar ile sizlerden gelecek olan programlara yer vermeyi düşünüyorum. Aslında sayfanın adına tam olarak karar veremedim. Bu nedenle bu sayfada her türlü ipucu, ek bilgi vs. bulunabilir.

Eğer dergiye bu şekilde program göndermek isterseniz yapacağınız iş çok basit. Önce VICE emülatör ile programı yazın ve çalıştırın. İstedığınız ekran çıkınca ALT+C ile ekranın bir kopyasını alın. BMP formatında kaydedilen resmi daha az yer kaplaması için bir grafik programı kullanarak JPG formatında kaydedin. Daha sonra ise gerekli düzenlemeleri yaparak sayfayı hazırlayın.

Bir sonraki sayıda isminizin çıkması için daha ne bekliyorsunuz ?

NOT : Göndereceğiniz her türlü dökümana ad-soyad, şehir, e-mail adresi, varsa web sitenizin adı ve kendinizi tanıtan kısa bir yazı eklemeyi unutmayın. Ayrıca alıp-satmak, takas yapmak istedikleriniz, aradıklarınız içinde bir köşe açmayı düşünüyorum. Hepsinden önemlisi eğer dergi için önerilerinizi, düşüncelerinizi ve eleştirilerinizi bekliyorum.

BEN KİMİM ?

PC'lerin ve pc dergilerinin hüküm sürdüğü bir zamanda COMMODORE 64 ile ilgili bir dergi çıkarmak için zamanını harcayan birinin kim olduğunu merak ediyorsanız işte karşınızda kısaca kişisel bilgilerim ve bilgisayar hayatım.

Adım ve soyadım İSMAİL ŞAHİN.1968 doğumluyum ve elektronik teknikeri olarak özel bir şirkette çalışıyorum. Evliyim ve (bu dergiyi hazırladığım tarihte - Kasım 2002) 16 aylık bir kızım var.

İlk bilgisayarım 1985 yılında SINCLAIR ZX SPECTRUM 48K oldu. O zamanlar Türkiye'ye ev bilgisayarları yeni yeni giriyordu. (Renkli televizyonlarda yeni üretilmeye başlanmıştı.)

Televizyonumuz siyah beyazdı ve UHF bandı olmadığı için bilgisayarı daha çok oyun amaçlı olarak arkadaşların evinde hem de renkli TV'ye bağlayarak kullanırdık. (O zamanki oyunlar ne kadar zevkli gelirdi oynarken.) Sonra UHF bandı olan ufak bir siyah beyaz TV aldım ve odanın ışığını kapatarak geceleri oyun oynardım. İlk sene hem oyun oynamakla hemde o zamanların meşhur programlama dili BASIC ile programlar yazmakla geçti. Durum böyleyken bir elektronik dergisindeki Z-80 assembler köşesi sayesinde SPECTRUM'un mikro işlemcisi olan Z-80 assembler öğrenmeye başladım. Birde baktımki kısa sürede oldukça işe yarar! Programlar yazmaya başladım. Mesela oyunun yükleme sırasında çıkan ekran resmini baş aşağı çevirme rutini, karakter seti değiştirme rutinleri, hatta dahada ileri giderek yine o zamanların bir numaralı kopya programı olan ULTRAFİLE'in kopyalayamayacağı uzunlukta olacak şekilde oyunları tek parça haline getirme rutinleri.....

Yıl 1986, sınıftaki arkadaşta COMMODORE 64 var ve hafiften C64 ve 6502 assembler ile ilgilenmeye başlıyorum. Derken 1987 yılında SPECTRUM'um bozuluyor. Garanti süresi bittiği ve bende de para olmadığı için bir süre bilgisayarsız kalıyorum. Ama içimdeki bilgisayar hevesi geçmek bilmiyor. Bir işe girip Teypli bir C64 sahibi oluyorum. Bir ay sonra da ilk disk driver'ımı (1541-II) alıyorum. İşte o gün bu gündür C64 ve assembler hayatımın vazgeçilmez bir parçası oldu. 1987'den itibaren C64 kullanıyorum. Dile kolay en az 15 sene. Hangi bilgisayar hiç bir değişikliğe uğramadan bu kadar zaman kullanılır. (PC'lere bakın hergün yeni bir parça üretiliyor ve en son aldığınız 1 sene içinde demode oluyor). Bu 15 sene boyunca 2 tane C64 bozdum, şu anda birinde BASIC ROM'u eksik olan 3 tane C64, en sonuncusunu Eylül 2002'de 1.500.000 TL'ye aldığım 3 adet 1541-II, 1 adet MULTI ICE 6 ve 1084 Monitör'üm var. Bunlardan başka yine Eylül 2002'de 500.000 TL'ye aldığım çalışır durumda ZX SPECTRUM 128K +2 bilgisayar var.

Yine bu süre boyunca BRONX ve ASCREAUS gruplarında coder olarak bulundum. Grup içinde kullandığım adlarım ise şunlar.

- 1 - PHANTOMLORD : Türkçe commodore dergisinin program döküm için gönderdiğim ve yayınlanan bazı programlarda bu ismi kullandım.
- 2 - PHANTOM : Bronx grubuna katıldığımda bu ismi kullanıyordum ve uzun süre kullandım.
- 3 - ALTAR : Yurt dışında başka bir PHANTOM olduğunu öğrenince adımı değiştirdim.
- 4 - SOLOMON : ASCREAUS grubunda ilk önce bu ismi kullandım. Arkadaş grubu kurunca bana bu ismi uygun görmüş.
- 5 - CRYPTIC : Solomon ismini beğenmediğim için değiştirdim ve ilk oyunumda kullandım.
- 7 - HADES : 2000 yılında ilk chat yaptığım zamanlarda kullandığım nicki C64 dünyasında kullanmaya karar verdim

Gruptayken bir çok intro yaptım. Yine ASCREAUS grubundayken tamamen makina dilinde 3 TAŞ oyununu yaptım ve yurt dışına dağıtıldı. Bir başka oyunum ise 3 bölümlük bir demo olarak dağıtıldı. Ve aynı grubun çıkardığı bir TOOL DİSC'te bir NOTE MAKER ve FILE LISTER programlarım kullanıldı ve yurt dışına dağıtıldı. Fakat iş-güç-evlilik derken C64'e yeterince vakit ayıramaz oldum. Üstelik bir de PC kullanmak durumunda kalınca C64'ü arada bir hatırlar oldum. Ama hiç bir zaman unutmadım. Son 1 senedir ise zaman buldukça C64'e pc harddiski ve floppy'si bağlama projesiyle uğraşıyorum. Bildiklerimi paylaşmak için bir web sayfası hazırlamaya karar verdim fakat güncellemeye zaman bulamadığım gibi herşeyi html şeklinde yapmak çok zaman alıyordu. Bu nedenle hazırlaması daha kolay olan bu dergiye C64 TÜRKİYE dergisine başladım. Ve gördüğünüz gibi bitti.

Eğer dergi için katkıda bulunmak isterseniz istemediğiniz kadar sayfa var....

BU SEVDA BİTMEZ