# Buildroot Kullanarak Etkin Gömülü Sistem Tasarımı

**Deniz Taşkın[1], Cem Taşkın[2], Eser Sert[3]**

[1] Trakya Üniversitesi, Bilgisayar Mühendisliği Bölümü, Edirne

[2] Trakya Üniversitesi, Tunca Meslek Yüksek Okulu, Edirne

[3] Trakya Üniversitesi, Teknik Bilimler Meslek Yüksek Okulu, Edirne

deniztaskin@trakya.edu.tr, cemtaskin@trakya.edu.tr, esersert@trakya.edu.tr

**Özet:** Gömülü sistemler özel amaçlar için tasarlanmış donanım ve yazılım içeren sistemlerdir. Donanım ve yazılım, bir grup işlevi yerine getirmek üzere etkin biçimde bir araya getirilmektedir. Gömülü sistemlerde kullanılmak üzere geliştirilmiş pek çok ticari ve ticari olmayan işletim sistemi bulunmakla birlikte genellikle Linux işletim sistemi tercih edilmektedir. Gömülü sistemler için Linux işletim sisteminin kullanımı birçok zorluğu da beraberinde getirmektedir. Bu çalışmada gömülü sistemler için işletim sistemi oluşturmayı kolaylaştıran Buildroot adlı yazılım, kullanımı ve sağladığı kolaylıklar anlatılmaktadır.

**Anahtar Kelimeler:** Gömülü sistem tasarımı, Linux, Buildroot

**Effective Embedded System Design Using Buildroot**

**Abstract:** Embedded systems consist of hardware and software components which are designed for special purposes. Hardware and software are effectively combined for performing group of functions. There are plenty of commercial and non-commercial operating systems designed for embedded systems which are mostly Linux based. Using Linux operating system for embedded systems brings lots of problems within. In this study usage and facilities of a tool called Buildroot that makes forming an embedded operating system easier are explained.

**Keywords:** Embedded System Design, Linux, Buildroot

## 1. Giriş

Günümüzde teknolojinin hızlı gelişimi ile birlikte elektronik cihazlar daha akıllı hala gelmiştir. Örnek olarak, birbiri ile iletişim halinde bulunan, birçok özelliği üzerinde barındıran, internet desteği bulunan ev aletlerinin sayısındaki artış verilebilir.

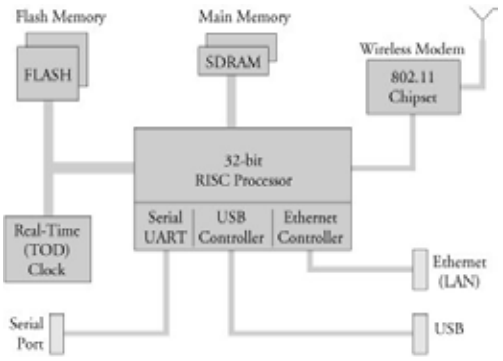
Bir ev aletini bir bilgisayar kadar akıllı kılabilmek için temelde cihaz içine bir kişisel bilgisayar koymak ve işletim sistemi yüklemek gereklidir. Bu, yer ve güç tüketim gereklilikleri ele alındığında imkansızdır. Sistem tasarımcıları bunun yerine gömülü sistemler kullanmayı tercih etmektedir.

Gömülü sistemler donanım açısından bakıldığında bir kişisel bilgisayardan çok daha düşük kaynaklara sahip sistemlerdir. Bununla birlikte güç tüketimleri de düşüktür. Üreticiler, müşteri gereksinimlerine göre birçok farklı sağlayıcıdan aldıkları birimleri ortak bir kart üzerinde toplayarak gömülü sistemlerin donanımlarını özel olarak üretmektedirler.

Gömülü sistemlerin donanım olarak birbirlerinden farklı olmasından dolayı işletim sistemlerinin de farklılıklar içermesi gerekmektedir. Birçok kişi tarafından bu özel donanımlarda göre üretilmiş özel Linux dağıtımları tercih edilmektedir.

## 2. Gömülü Sistemin Anatomisi

Şekil 1'de tipik bir gömülü sistemin anatomisi görülmektedir. Bu sistem kablosuz ve kablolu iletişim, gerçek zaman saati, USB, seri iletişim özelliği barındırmaktadır. Sistemin kalbinde 32 bitlik bir RISC işlemcisi bulunmaktadır. Flash bellek değişken olmayan program ve verileri saklamak için kullanılmaktadır. Ana bellek ise uygulamaya göre kapasitesinde değişkenlik gösteren SDRAM tipinde bellektir. Gerçek zaman saati, tarih ve zamanı tutmaktadır. Bu örnekte ayrıca Ethernet ve USB arabirimleri bulunmaktadır. Konsol işlemleri için RS-232 arabirim ve kablosuz iletişim özelliklerinin gerçekleştirmek için 802.11 yonga kümesi kullanılmaktadır.




**Şekil 1.** Örnek gömülü sistem

## 3. Gömülü İşletim Sistemi

Bir gömülü sistem için özel bir işletim sisteminin yazılması pek çok zorluğu içermektedir. Birçok bileşeni olan işletim sisteminin özel olarak bir donanım için sıfırdan yazılması büyük bir ekip çalışmasının yanı sıra uzunca bir süre gerektirmektedir. Gömülü sistemlerin yaşam döngüsünün ve market zamanının sınırlı olmasından dolayı geliştirme zamanının olabildiğince kısa olması gerekmektedir. Bu da işletim sistemi gibi içinde birçok bileşeni içeren yazılımın hazır alınması ya da Linux gibi açık kaynak kodlu bir işletim sisteminin kullanılması ile mümkündür.

Bir Linux dağıtımı kullanıma hazır (derlenmiş) ikili paketlerin birleşimidir. Günümüzde standart Linux dağıtımları x86 işlemci mimarisinde kullanılmak üzere tasarlanmıştırlar. Bu hazır dağıtımlardan herhangi birisini kullanmak çalışan bir sistem oluşturmanın en kolay gibi gözükse de birçok dağıtım gömülü sistemlerde çalışmak üzere tasarlanmamıştır. Standart x86 mimarisi dışında farklı birçok mimariye göre üretilen gömülü sistemler için işletim sisteminin özel olarak derlenmesi gerekmektedir.

Donanıma özel bir Linux işletim sistemi geliştirmek için depolama ortamının hazırlanması, Linux çekirdeğinin (kernel), sisteme gerekli çalıştırılabilir uygulamaların, sürücülerin, yardımcı yazılımların kaynağından indirilmesi ve ilgili donanıma göre derlenmesi gerekmektedir. Bu işlemlerin gerçekleştirilmesi, sistemi oluşturan birçok bileşen bulunmasından dolayı uzun süreler gerektirmektedir.

## 4. Buildroot

Bir gömülü sistem için Linux dağıtımı oluşturmanın en zor kısmı, sistemde yapılacak en ufak değişikliklerde dahi derleme aşamalarının tekrardan yapılması gerekliliğidir. Bu aşamaların uzun zaman alması, derleme işlemlerini otomatik biçimde gerçekleştiren yardımcı araçların ortaya çıkmasına neden olmuştur. Buildroot, Scratchbox, ELDK, crosstool bunlardan bazılarıdır. Buildroot gömülü sistem geliştiricileri tarafından çoğunlukla tercih edilen ve geliştirici tarafından özelleştirilebilen bir çapraz geliştirme aracıdır. Sisteme ait Linux çekirdeği, busybox, binutils, gcc ve uClibc bileşenlerinin kaynaklarından indirilmesi ve derlenmesi işlemlerini otomatik olarak gerçekleştirir.

### 4.1. Buildroot'un yüklenmesi

Bir gömülü sistem üzerinde işletim sistemi geliştirmek ve derlemek mümkün değildir. Bunun yerine geliştirici ortam kullanılmaktadır. Bu ortam genellikle x86 mimarisinde olan üzerinde Linux işletim sistemi bulunan bir bilgisayardır. İşletim sistemi, geliştirici ortamda derlendikten sonra gömülü sisteme aktarılır.

Geliştirici ortama buildroot çapraz geliştirme aracı farklı şekillerde yüklenebilir. Bunlardan birisi sıkıştırılmış bir dosya olarak indirildikten sonra bir klasör içine açılmasıdır. Fakat bu işlemden önce bu yardımcı araca ait bir takım gerekli paketlerin kurulması gerekmektedir. Apt-get komutu ile bu bileşenler yüklenmelidir.

```
sudo apt-get install build-essential
bison flex gettext libncurses5-dev texinfo
autoconf automake libtool libpng12-dev
libglib2.0-dev libgtk2.0-dev gperf libxt-
dev libxp-dev
```

Diğer bir yöntem ise subversion yardımcı programını kullanmaktır. Öncelikle subversion geliştirici ortamına yüklenir.

```
sudo apt-get install subversion
```

Ardından subversion kullanılarak buildroot sisteme otomatik olarak yüklenebilir.

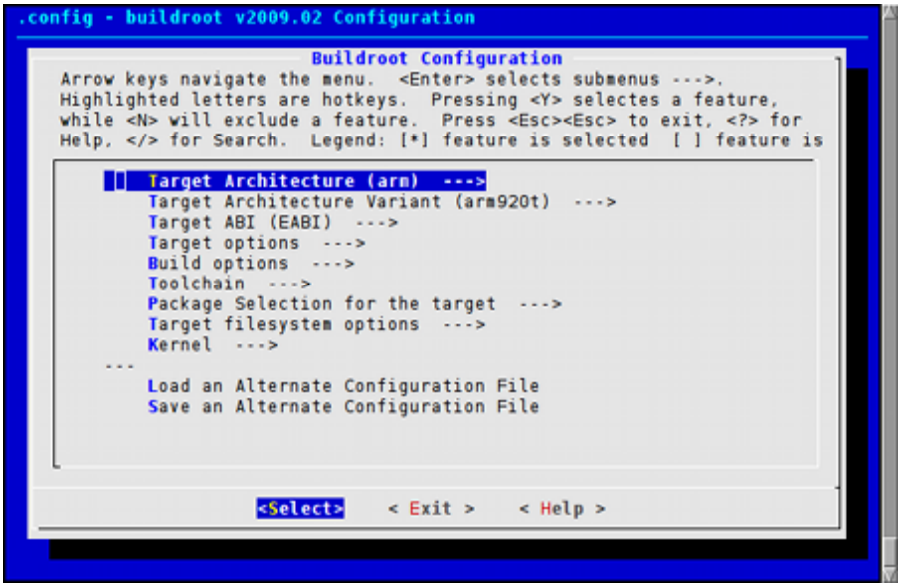
```
svn co svn://uclibc.org/trunk/buildroot
```

### 4.2. Buildroot ile özelleştirme

Buildroot geliştirme platformuna yüklendikten sonra:

```
make menuconfig
```

komutu ile sisteme ait seçenekler ayarlanmaktadır. Şekil 2.'de görüldüğü gibi buildroot yardımcı programı geliştiriciye bu seçenekleri ayarlayabileceği bir ana menü sunmaktadır.




**Şekil 2.** Buildroot ana menüsü

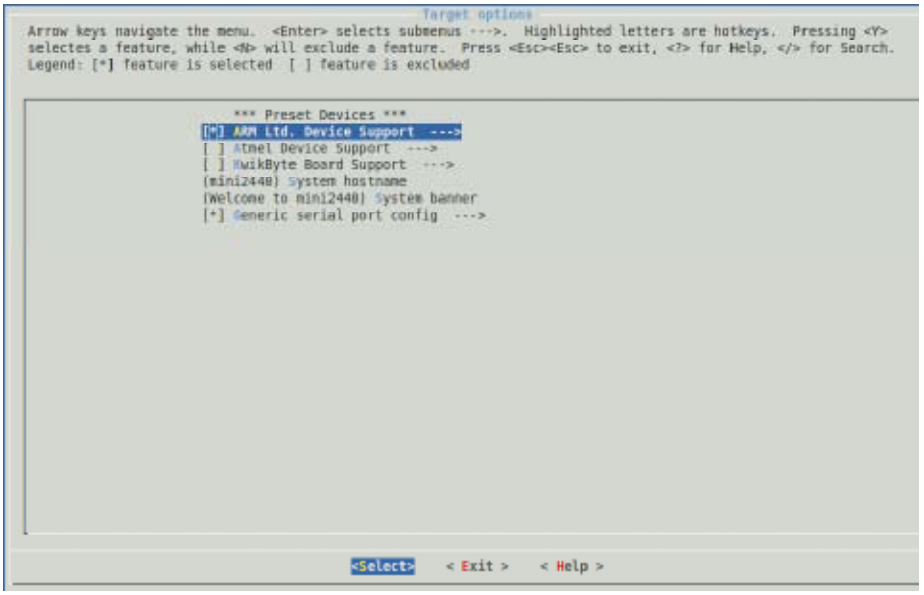
*Target architecture* ile hedef sisteme ait mimari seçeneklerinden birisi seçilmektedir. Buildroot bünyesinde bulunmayan mimariler daha sonra kullanıcı tarafından eklenebilir. Şekil 3'te de görüldüğü gibi *Target options* ile kart parametreleri, hoş geldin mesajı gibi platforma ait gerekli ayarlar yapılmaktadır.

Tüm bunların dışında buildroot ile Hedef mimariyle ilgili detaylı seçenekler, kullanılacak kütüphanelerle ilgili seçenekler, ağ desteği ve ayarları, geliştirme ortamının dosya sistemi gibi ayarlar kolaylıkla yapılabilmektedir.

Gerekli ayarlar yapıldıktan sonra programdan çıkarak ayarlar kaydedilir. Ardından komut isteminde

```
make
```

komutu verilir. Bu komut ile birlikte buildroot sayesinde gerekli tüm paketler kaynağından in-




**Şekil 3.** Hedef seçenekleri

dirilerek hedef sisteme uygun biçimde derlenmektedirler. Derleme işlemlerinden sonra

```
output/images/
```

klasöründe istenen özelliklerde ve istenen dosya sisteminde işletim sistemine ait görüntü dosyaları hazırlanmaktadır.

### 3. Sonuç ve Öneriler

Buildroot gömülü sisteme ait çapraz derleme araçlarının tamamını yönetmenin dışında kök dosya sistemi, çekirdek ve bootloader'ın oluşturulması ve yapılandırmasını da sağlamaktadır. Ayrıca buildroot bu adımlardan sadece birkaçı için özelleştirilebilmektedir. Buildroot yüzlerce paketin yanında X.org, Gtk2, Qt, DirectFB, SDL, GStreamer ve birçok ağ ve sisteme yönelik kütüphaneleri desteklemektedir. Ayrıca hedef sistemin çeşidine göre birçok dosya sistemini destekler. JFFS', UBIFS, tarball, romfs, cramfs, squashfs bunlardan bazılarıdır. Buildroot sayesinde sadece gerekli basit ayarlar yapıldıktan sonra daha kısa bir sürede gömülü bir Linux işletim sisteminin oluşturulması mümkün olmaktadır.

### 4. Kaynaklar

[1] http://buildroot.uclibc.org/

[2] http://www.linuxfromscratch.org/

[3] Christoper Hallinan, "Embedded Linux Primer: A Practical, Real-World Approach", 2007

[4] Joseph A. Fisher, Paolo Faraboschi, Cliff Young, "Embedded Computing", (2005)

[5] Karim Yaghmour, Jon Masters, Gilad Ben-Yossef, Philippe Gerum, "Building Embedded Linux Systems", 2003

[6] Michael Barr, Anthony Massa, "Programming Embedded Systems", 2007

[7] P. Raghavan, Amol Lad, Sriram Neelakandan, "Embedded Linux System Desgin and Devolopment" , 2006

[8] Richard Zurawski, "Networked Embedded Ssytems", 2009