

Android Tabanlı Mobil Ödeme ve Araç İçi Kontrol Merkezi

Android Based Mobile Payment and Vehicle Control Device

Ercüment Türk¹, Aydoğan Savran², Sadık Arslan¹

¹Araştırma Geliştirme Bölümü, İzmir, Türkiye
Kentkart Ege Elektronik A.Ş.
ercument.turk@kentkart.com.tr, sadik.arslan@kentkart.com.tr

Elektrik-Elektronik Mühendisliği Bölümü
Ege Üniversitesi
aydogan.savran@ege.edu.tr

Özet

Bu çalışmada, tüm ücret toplama sistemlerinde kullanılabilecek aynı zamanda araç içinde her türlü kontrolü sağlayabilecek akıllı bir gömülü sistem cihaz tasarlanmış ve Android İşletim Sistemi'nin bu cihaz üzerine uyumlandırılması gerçekleştirilmiştir. Mobil ödeme sistemlerinden beklenen özellikler üzerinde durulmuş, Android İşletim Sisteminin genel yapısı incelenip herhangi bir gömülü cihaza uyumlandırma için gerekli bilgiler verilmiştir. Çalışmaya başlanılmadan önce gereksinim analizleri yapılarak sistemin sahip olması gereken donanım özellikleri belirlenmiştir. Daha sonra tasarlanan gömülü cihaz için önyükleme yazılımı, Linux çekirdeği ve kök dosya sistemi hazırlanmıştır. Önyükleyicinin çevresel parametreleri ihtiyaçlar doğrultusunda belirlenmiştir. Android kaynak kodu gerekli düzenlemelerden sonra derlenmiş ve sistem imajı elde edilmiştir. Çalışmanın sonucunda Android İşletim Sistemi tabanlı mobil ödeme ve araç içi kontrol cihazı tasarımı tamamlanmıştır.

Abstract

In this study, an intelligent embedded system which it can be used in fare collection system and all indoor vehicle applications at same time, is designed and Android porting to this device. We discuss expected characteristics of mobile payment and how Android works on any hardware and the concepts that outline the porting of Android onto any hardware. Requirement analysis is performed before starting the study to determine hardware features of the system. Bootloader, Linux kernel and root file system is prepared for this device. Environmental parameters of the bootloader is determined in accordance with the requirements. Android source code is patched and compiled to get system image. In conclusion, Android based mobile payment and vehicle control center design has been completed.

1. Giriş

Günümüzde teknolojik gelişmeler sayesinde güçlü, yüksek performanslı donanım ve daha gelişmiş işletim sistemleri tüketicilerin hizmetine sunulmaktadır. Bu gelişmelerin paralelinde akıllı telefon pazarı her geçen gün daha da büyümekte, yeni ürün ve modeller piyasaya sürülmekte ve her geçen gün gelişen ürün yelpazesi tüketicilerin hayatında önemli bir yer tutmaktadır. Gelişen teknolojiler ile birlikte insanların beklentileri de artmaktadır. Artık tüketiciler sadece cep telefonlarını kullanarak hem özel hem de resmi işlemlerini fazla gayret harcamadan gerçekleştirmek istemektedirler. Bu konular hayatımızın tüm alanlarında önemli bir yer tuttuğu gibi toplu ulaşım da ön sıralarda yer almaktadır. Özellikle mobil ödeme sektöründeki gelişmeler ve bu teknolojinin her geçen gün yaygınlaştığı düşünüldüğünde, mobil uygulamalardaki bilgi güvenliğinin sağlanması ve tüm mobil ödemeler için ortak bir platformun ulusal ve dünya çapında yaratacağı fayda oldukça büyük olacaktır.

Mobil ödeme sisteminin ortaya çıkışı ve böylece evrensel ticaretin gelişmesiyle daha hızlı daha kolay alışveriş imkânı oluşmuştur. Mobil alışverişlerde NFC kullanımı ile ilgili çalışmalar [1] ve [2]’de sunulmuştur. Husni [2] Android uygulamalarıyla iki cep telefonu arasında NFC yardımıyla bir alışveriş uygulamasının nasıl yapılabileceğini detaylı bir biçimde anlatmıştır. [3]’ta bulut mimarisinin de kullanıldığı bir NFC destekli mobil ödeme uygulaması anlatılmaktadır. [4]’de Android İşletim sistemine sahip cihazlarda NFC uygulamalarıyla tren bilet satış uygulaması örneği verilmektedir. [5]’te ise gelecekteki mobil ödeme sistemleri için NFC’nin bir değerlendirilmesi yapılmıştır.

[6]’da şehir içi ulaşımında kullanılan otobüsler için tasarlanan akıllı kontrol sistemi anlatılmaktadır. Son zamanlarda kullanıcı etkileşimli ve görsel arayüzü gelişmiş sistem

istekleri de bir hayli artmıştır. [7]'de Android tabanlı araç içi bilgilendirme ve medya yazılımından bahsedilmiştir. Donanım çeşitliliği sistem uyumluluğu konusunda büyük dezavantaja neden olmakta, sistem entegrasyonu çok büyük maliyetlere sebebiyet vermektedir. Her türlü işletmesel uygulamanın donanımdan bağımsızlığı ve sürdürülebilirliğin yüksek olduğu ortak bir platform arayışı bulunmaktadır.

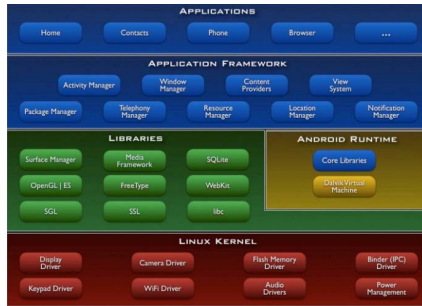
Tüm bu ihtiyaçlar, üzerinde gerekli donanımsal birimlere sahip, mevcut sistemlerdeki donanıma bağımlılık ve geliştirmeye açık olmayan dezavantajları giderecek, kullanıcıların kendi uygulamalarını hazırlayıp kullanabileceği bir platform oluşturmak, sadece ücret düşümü için cihaz tasarlamak yerine mobil ödeme, araç takip, kullanıcı bilgilendirme cihazları, el terminalleri, kişisel asistan bilgisayarlar için aynı ürünün kullanılması gibi ihtiyaçlardan dolayı bir Android İşletim Sistemine sahip araç içi kontrol ve mobil ödeme merkezi geliştirilmiştir.

Tasarlanan Android İşletim Sistemi tabanlı cihazın NFC, Bluetooth, Wireless, 3G, RFID, Denetleyici Alan Ağı (Controller Area Network, CAN) gibi standartları desteklemesi sağlanmıştır ve böylece tek bir cihaz ile tüm işlemler yerine getirilecektir.

2. Android İşletim Sistemi

Android, mobil telefon, tablet ve kişisel bilgisayarlar için Google ve Açık Telefon Birliği (Open Handset Alliance, OHA) [8] tarafından Linux çekirdeği üzerine geliştirilen açık kaynak bir işletim sistemidir.

Android mimarisi, Linux çekirdeği, sistem kütüphaneleri, Dalvik Sanal Makinası, uygulama çatıları ve uygulama katmanından oluşmaktadır. Bu mimari Şekil 1'de gösterilmiştir.



Şekil 1: Android mimarisi. [9]

2.1. Linux Çekirdeği

Android İşletim sistemi her ne kadar Linux çekirdeği üzerine inşa edilmiş olsa da bu aslında alışlagelmiş bir Linux çekirdeği değildir. Linux'un birçok yeteneği mobil cihazların kısıtlı kaynakları için yeniden tasarlanmıştır.

Android Honeycomb'a [10] kadar Linux 2.6 sürümlü çekirdek kullanılırken, Ice Cream Sandwich'ten [11] sonra 3.0 sürümlü Linux çekirdek kullanılmaya başlanmıştır. Sürücü desteği genişliği, hafıza ve süreç yönetimindeki başarısı, ağ yönetimindeki başarısı, güvenlik ve hız konularında kendini kanıtlamış olması ile açık kaynaklı oluşu Android İşletim Sisteminin Linux çekirdeği üzerine kurulmasının en önemli

nedenleridir. Ayrıca Android özellikli cihazların donanım, model ve teknik özellik bakımından birbirlerinden oldukça farklı oluşu uyum problemlerini de beraberinde getirir. Linux çekirdeği farklı mimarilere göre derlenir ve gerekli sürücü yazılımları çekirdeğin içerisine eklenerek yaşanan uyum sorunları çözülür. Çekirdek donanım ile yazılım arasında soyut bir katman görevi görür.

Ancak standart Linux dağıtımları cep telefonları gibi gömülü sistemler için tasarlanmadığı için Google tarafından Android ile birlikte kullanılmadan önce güç tüketimi, hafıza ve çalışma zamanlı sistem kaynakları yönetimi gibi konularda düzenlemeler yapılmıştır.

Android içerisinde çalışan onlarca servis ve uygulamanın birbiriyle iletişime geçmesi gereken durumlarda güvenli ve performanslı bir yönetim sağlayabilmek için Bağlayıcı Sürücüsü (Binder Driver) [12] Android işletim sistemine dahil edilmiştir. Özellikle paylaşımlı hafıza üzerinde yüksek performans sağlamaktadır. Her bir süreç için ayrı iplik havuzu (thread pool) tutarak iplik yönetimini de gerçekleştirir. Ayrıca süreçlerin yaptığı çağrılar sonucu referanslarını takip ederek, kullanılmayan alanları hafızadan silerek yer açılmasını sağlar.

2.2. Kütüphaneler

Doğal kütüphaneler C/C++ dilleri ile yazılmıştır. Genellikle donanım sürücülerinin ve doğal dille yazılmış programların uygulama çatısı üzerinden kullanımı amacıyla yazılırlar.

Bu kısma ait olan Donanım Soyutlama Kütüphaneleri (Hardware Abstraction Layer) donanım katmanı ile diğer katmanlar arasındaki soyutlamayı sağlayan katmandır. Bu katmandaki kütüphaneler Android işletim sisteminin donanımları yönetmesi için ihtiyaç duyduğu arayüzleri tanımlar.

Hemen hemen her uygulamanın kullandığı standart libc kütüphanesinin boyutunu küçültmek amacı ile bu kütüphane Android için özelleştirilmiştir. Bu sayede hafızada daha az yer işgal etmesi sağlanmıştır. Elde edilen bu yeni kütüphane Bionic Libc olarak adlandırılmıştır. Bu kütüphane Android için özel olarak uyumlandırıldığı için standart libc'ye göre daha hızlıdır.

Üç Boyutlu işlemler için gereken kütüphaneler (OpenGL ES), Mpeg4, Mp3, JPG desteği için medya kütüphaneleri, web tarayıcı için Libwebcore, veritabanı işlemleri için de SQLite kütüphaneleri bulunur.

2.3. Android Çalışma Zamanı

Bu bölümde Linux çekirdeği kütüphaneleri ile Java ile birleştiği bölümdür. Dalvik sanal makine ve Java ile yazılmış çekirdek kütüphaneler Android çalışma zamanı bileşenlerini oluşturur [13].

2.3.1. Dalvik Sanal Makinası

Android uygulaması geliştirildiğinde yazılan sınıflar, öncelikle sekizlik kod (bytecode) olarak adlandırılan dosya türlerine derlenir. Bu dosyalar işlemci üzerinde doğrudan çalıştırılabilir kodlar değildir. Öncelikle bu sekizlik kodları yorumlayıp üzerinde çalışılan işlemci komutlarına

dönüştürecek bir katmana ihtiyaç vardır. Java Sanal Makinası (Java Virtual Machine, JVM) bu ara katmanı oluşturarak yazılan herhangi bir programın herhangi bir sistemde çalıştırılmasını sağlar. Java programlarının sistemden bağımsız olmalarının temelinde, derlendiklerinde çalıştırılabilir kodlar üretmek yerine sekizlik kod üretilmesi ve bunların sanal makine tarafından yorumlanması yatar.

Android uygulamalarında da mantık klasik Java uygulamalarındakine benzerlik gösterir. Java kodları, JVM’de olduğu gibi önce sekizlik kod’a derlenir. Dalvik Sanal Makinesi (Dalvik Virtual Machine, DVM) bu derlenmiş kodu yorumlayıp .dex uzantılı çalıştırılabilir dosyalara dönüştürür [13], [14].

2.4. Uygulama Çatısı

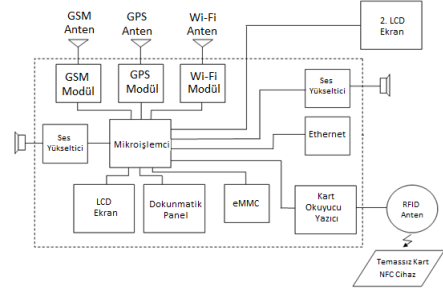
Doğal kütüphanelerin kullanımını sağlamak ve uygulamalar ile donanım arasında bir soyut katman sağlama amacıyla uygulama çatısı geliştirilmiştir. Uygulama geliştiricilere geniş bir platform sunar. Bu katman sayesinde uygulama geliştiricileri kullanacakları donanım hakkında çok fazla detaya sahip olmadan istedikleri uygulamayı geliştirebilirler. Ayrıca bu çatı sayesinde uygulama taşınabilirliği de sağlanmış olur. Bu katmanda yer alan bütün servislerin temel amacı bir uygulamanın yaşam döngüsünde ihtiyaç duyduğu fonksiyonları yönetmektir. Buradaki bütün bileşenler bir uygulama gibidir ve arka planda süreç olarak çalışırlar.

2.5. Uygulama Katmanı

Uygulama katmanı Java programlama diliyle yazılmış uygulamaları içerir. Bu uygulamalar, geliştiricilerin yazdığı uygulamalar olabileceği gibi marketten de indirilebilirler.

3. Mobil Ödeme ve Araç İçi Kontrol Merkezi’nin Donanımı

Çalışma kapsamında geliştirilen Mobil Ödeme ve Araç İçi Kontrol Merkezi cihazının şeması Şekil 2’de verilmiştir. ARM işlemci kullanılarak geliştirilen cihazda veri depolama için Gömülü Multimedya Kartı (Embedded MultiMediaCard, eMMC) ve Çift Veri Hızlı Rasgele Erişimli Tip Üç Bellek (Double Data Rate Type Three Random Access Memory, DDR3 RAM) kullanılmıştır. Veri bağlantısı için Mobil İletişim İçin Küresel Sistem (Global System for Mobile Communications, GSM) modül, konum bilgileri için Küresel Konumlama Sistemi (Global Positioning System, GPS) modülü, kablosuz haberleşmenin sağlanabilmesi için 802.11b/g/n standardında Kablosuz Bağlantı Alanı (Wireless Fidelity, Wifi) modülü, Ethernet, Evrensel Seri Hat (Universal Serial Bus, USB), 24bit Likit Kristal Ekran arayüzü (Liquid Crystal Display LCD), Entegreler Arası Devre (Inter-Integrated Circuit, I2C), Çevresel Seri Arayüz (Serial Peripheral Interface, SPI), Darbe Genişlik Modülasyon içeren uç (Pulse-Width Modulation, PWM), Evrensel Asenkron Alıcı Verici hat (Universal Asynchronous Receiver Transmitter, UART), Abone Kimlik Modülü arayüzü (Subscriber Identification Module, SIM), Güvenli Sayısal Hat (Secure Digital, SD), Analog giriş ve GPIO bağlantılarını içermektedir. Bunların yanında güç tüketimi için geliştirilen tümleşik devreleri içermektedir [15].



Şekil 2: Basitleştirilmiş donanım şeması.

Ödeme terminalinin Yakın Alan İletişimi (Near Field Communication, NFC) uyumlu olması için ise tasarlanan olan anakartta NXP firmasının geliştirdiği CLRC663 temassız okuyucu entegresi kullanılmıştır. Bu entegre, sistemde NFC ile haberleşmede okuma ve yazma görevini üstlenecektir.

4. Android İşletim Sisteminin Entegrasyonu

Bu bölümde daha önce üzerinde Linux çekirdeği koştan gömülü bir sisteme Android entegrasyonu anlatılmıştır.

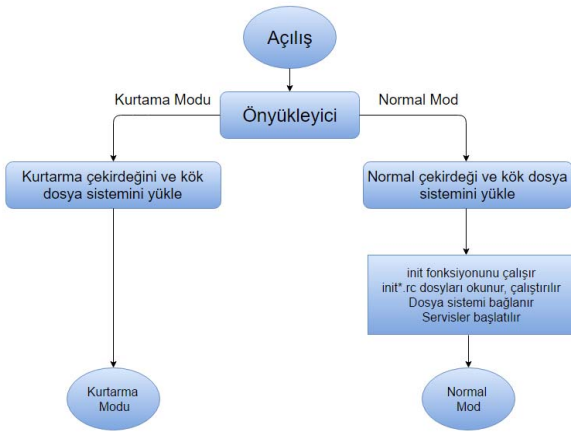
4.1. Önyükleme Yazılımının Uyumlandırılması

İndirgenmiş Komut Takımıyla Hesaplama (Reduced Instruction Set Computing, RISC) tabanlı gömülü sistemlerde, Linux Yükleycisi (Linux Loader, LILO), Büyük Birleşik Önyükleyici (Grand Unified Bootloader, GRUB) gibi önyükleyiciler pek yaygın değildir. Bunun yerine Denx şirketi tarafından geliştirilen GPL lisanslı Evrensel Önyükleyici (Universal Bootloader, U-Boot) sistemi kullanılır.

U-boot gömülü sistem açılır açılmaz çekirdek ve kök dosya sistemini yükleyebilir ya da kullanıcı ile terminal üzerinden etkileşime geçebilir. Gömülü sistemler için çok önemli olan çevre değişkenleri U-boot kaynak kodu derlenirken kodun içerisine gömülebileceği gibi üzerinde gelen kabuk yardımıyla kullanıcı etkileşimli bir şekilde yazılabilir. Bu çevre değişkenleri kullanılarak U-boot’un davranışı değiştirilebilmektedir

U-boot üzerinden sistemin açılışı ve genel karakteristiği ile ilgili pek çok özellik belirlenebilir. Ağ ortamından kök dosya sistemi, Linux çekirdeği ya da herhangi bir dosya yükleyebilir. Farklı önyükleme seçenekleri ile ağ ortamından, USB bellekten, Nand Bellek’ten, SD karttan, seri arayüzden açılabilir. Bu seçim tamamen uygulamanın gereksinimleri doğrultusunda belirlenir.

U-boot normal ve kurtarma modu olmak üzere iki farklı modda açılış sağlar. Normal modda önyükleyici sıra ile önce Linux çekirdeğini daha sonra ise kök dosya sistemini yükler. Böylece kontrolü Linux çekirdeğine ve kök dosya sistemindeki uygulamalara devretmiş olur. Kurtarma modunda ise normal kullanımda erişilemeyen Linux çekirdeği ve kök dosya sistemi açılır. Açılış modlarının akış diyagramı Şekil 3’te verilmiştir.



Şekil 3: Önyükleyici açılış diyagramı

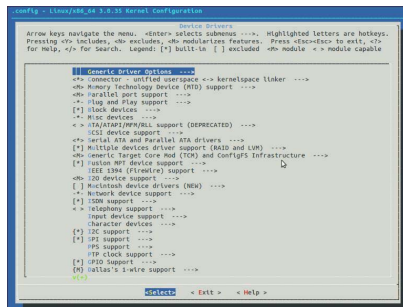
4.2. Linux Çekirdeğinin Uyumlandırılması

Çekirdek işletim sisteminin esasını oluşturur. Gömülü sistem açıldığında belleğe yüklenir ve kapanana kadar orada kalır. Genelde zImage, bzImage, uImage, fImage gibi adlara sahiptir. Tek bir adet dosyada oluşur. Bu dosya Linux çekirdek kodunun derlenmesi ile elde edilir.

Üzerinde bulunduğu makine mimarisi için derleme yapan derleyicilere doğal derleyici (native compiler) denir [16]. Çalışma sırasında x86 mimarili geliştirme ortamında çalıştığımız ve hedef gömülü sistemimiz ARM mimarisine sahip bir işlemci içerdiği için bir çapraz derleyiciye ihtiyacımız vardır. Bu üreticiden temin edilebildiği gibi eğer istenirse alternatif yollardan da elde edilir.

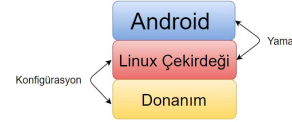
Çapraz derleyici temin edildikten sonra Linux çekirdeğinin kaynak kodunu temin etmek gerekir. Bu aşamada ya üreticiden Android ve kullanılan işlemci ile anakart için tüm yamaları yapılmış çekirdek kodunu temin edilir ya da istenilen çekirdek kodu [17]'den temin edilir. Üreticiden tedarik etme yolu seçilmezse çekirdek koduna gereken tüm yamalar kullanıcı tarafından tek tek yapılmalıdır.

Çekirdek kodu elde edilip, gerekli yamalar uygulandıktan sonra derleme işlemi için uygun yapılandırma ayarlarının yapılması gerekir. Bunun için hedef sistem iyi analiz edilmeli gerekli modüllerin ve özelliklerin desteği yapılandırma ayarlarıyla verilmelidir. Şekil 4'te çekirdek kodunun yapılandırma ekranı görülmektedir.



Şekil 4: Linux çekirdeği yapılandırma ekranı

Linux çekirdeği Android ile donanım arasındadır ve donanımı Android'ten soyutlar. Bu sayede taşınabilirlik, çok çeşitli donanım desteği sağlanmış olur. Çekirdeği Android uyumlu hale getirmek için yama işlemini yapmak gerekirken, donanım uyumu için ise konfigürasyon dosyaları kullanılır. Şekil 5'te bu ilişki gösterilmiştir.



Şekil 5: Linux çekirdeğinin donanım ve Android arasındaki ilişkisi

4.3. Kök Dosya Sisteminin Hazırlanması

Hazırlanan Linux çekirdeği önyükleyici tarafından hafızaya yüklenip çalışmaya başladığında bir dosya sistemi arar. Bu dosya sistemi üzerinde başlangıç betiklerini çalıştırır. Bu betikler de özelleştirilerek sistem ihtiyaçlarına uygun programları, servisleri çalıştırıp gerekli düzenlemeleri yapar.

Kök dosya sistemi ilk açılış sırasında çalışacak betiklerin ve bu betikler tarafından kullanılan dosyaları barındırır. Salt okunur olan bu dosya sistemi RAM Dosya Sistemi (ramfs) şeklindedir ve çalışma sırasında çekirdek ya da önyükleyici tarafından RAM'e yüklenir. Cihaz çalıştığı süre boyunca bu dosya sistemi RAM üzerinde durur. RAM üzerinde açıldığından dosya yapısı ve içeriğinin değiştirilmesi hiçbir etkiye neden olmaz.

Kök dosya sistemi üzerinde sadece gerekli yetkilere sahip kullanıcılar işlem yapabilir. Genellikle son kullanıcı ekstra bir işlem yapmazsa bu yetkiye sahip olamaz. Hem yetkilendirme konusu hem de dosya içeriğinde değişiklik ihtiyacı olduğundan son kullanıcı için depolama biriminde başka bir dosya sistemi oluşturulmuştur. Kullanıcı bu alana dilediği gibi erişebilir ve uygulamalarını bu alanda saklayıp istediği düzenlemeyi yapabilir.

4.4. Android Kaynak Kodunun Derlenmesi

İstenilen Android sürümünü kaynak kodları indirilir. Android kodunun hedef sisteme uygun olarak derlenmesi için gerekli yapılandırma dosyaları üreticiden temin edilebilir ya da istenirse cihaza özel yapılandırma dosyaları hazırlanabilir. Ürüne özgü derleme dosyaları ile ürünün adı ve teknik özellikleri Android kaynak koduna eklenir. Yapılandırma dosyaları ile cihazın Android'in hangi özelliklerine sahip olacağı belirlenebilir.

Derleme işlemi başarı ile bittikten sonra kaynak kodun bulunduğu dizin altında belirtilen klasörde imaj verileri elde edilir.

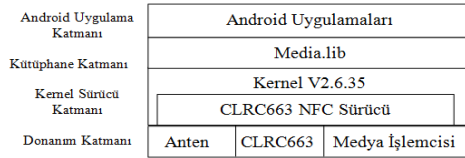
4.5. Hedef Ürün İçin Gerekli Android Modüllerinin Düzenlenmesi

Ödeme terminalinin 3G, Edge bağlantılarını sağlaması, telefon görüşmelerinin gerçekleştirilmesi, SMS özelliğinin kazandırılması, internet bağlantısının sağlanması için terminalde kullanılacak olan modem ile Android Telefon Servisi arasındaki bağlantıyı sağlayan Radyo Arayüz Katmanı

(Radio Interface Layer, RIL) üzerinde işlemler gerçekleştirilecektir. Bu aşamada ilk yapılması gereken, sahip olduğumuz GPRS modem için Linux çekirdeği kısmında çalışacak sürücünün yazılması ve Radyo Servisi'nin, modemimize ait kütüphane dosyalarıyla uyumlu çalışmasını sağlamaktır. Daha sonra ise Uygulama Çatısı ile Uygulama Katmanı Android Yazılım Geliştirme Kiti (Software Development Kit, SDK) sürümüne göre kullanılabilir hale gelir.

Ödeme terminalinin kablosuz bağlantı (802.11n) desteği vermesi için kullanılan kablosuz modem için sürücü yazılımları yazılır. Android'in kablosuz mimarisine uygun sürücü yazılımı tasarlanıp, Java iskeleti üzerinde koşacak olan kablosuz bağlantı uygulamasına hizmet verilecektir. Bu sayede gömülü cihaza her türlü uygulamada kullanabilmesi için kablosuz bağlantı desteği sağlanır.

NFC entegrasi için bir sürücü yazılımı yazılmıştır. Şekil 6'da tasarlanan NFC yapısının sistemsel gösterimi bulunmaktadır. Bu yapı donanım elemanlarından, Linux çekirdeği ve temassız okuyucu entegrasi sürücüsünden, kütüphanelerden ve Android uygulamalarından oluşmaktadır.



Şekil 6: Sistemin NFC Mimarisi

5. Sonuçlar

Bu çalışmada toplu ulaşım sisteminde ücret toplama, araç takip, araç içi bilgilendirme ve kontrol uygulamalarında kullanılan akıllı bir gömülü sistem tasarlanmış ve bu gömülü sistem üzerinde Android entegrasyonu gerçekleştirilmiştir. Özellikle yurtiçindeki literatür taramalarında benzer bir çalışmaya rastlanılmamıştır. Çalışma hem akademik hem de sektörel anlamda çok önemli bir yere sahiptir.

Android İşletim Sistemi Tabanlı Mobil Ödeme ve Araç İçi Kontrol Merkezi sayesinde mobil ödeme uygulamaları ile toplu ulaşım sistemlerinin araç içi yönetiminde kullanılacak bir gömülü sistem ihtiyacı giderilmiştir. Android İşletim Sistemi sayesinde hem üretici kendi uygulamalarını kullanabilir hem de marketten indirebileceği herhangi Android uygulamasını da bu cihazda çalıştırabilir. Android İşletim Sistemi sağladığı SDK sayesinde donanım bağımsız bir platform sağlar. Bu sayede uygulama yazan yazılımcının cihaz hakkında detaylı bilgiye sahip olması gerekmez. SDK'ya uygun kod yazması yeterlidir. Böylece sistem bakım ve entegrasyon maliyetleri azalır.

6. Teşekkür

Yazarlar, bu çalışmaya maddi ve laboratuvar olanakları açısından verdiği destek için Kentkart Ege Elektronik A.Ş.'ne teşekkür ederler.

7. Kaynaklar

- [1] Timalsina, S. K., Bhusal, R. and Moh, S., "NFC and its application to mobile payment: Overview and comparison", *Information Science and Digital Content Technology (ICIDT)*, 2012, 203-206.
- [2] Husni, E. and Purwantoro, S., "Shopping application system with Near Field Communication (NFC) based on Android", *System Engineering and Technology (ICSET)*, 2012, 1-6.
- [3] Pourghomi, P. and Ghinea, G., "Managing NFC payment applications through cloud computing", *Internet Technology And Secured Transactions*, 2012, 772-777.
- [4] Nasution, S. M., Husni, E. M. and Wuryandari, A. I., "Prototype of train ticketing application using Near Field Communication (NFC) technology on Android device", *2012 International Conference on System Engineering and Technology (ICSET)*, 2012, 1-6.
- [5] Ondrus, J. and Pigneur, Y., "An Assessment of NFC for Future Mobile Payment Systems", *Sixth International Conference on the Management of Mobile Business (ICMB 2007)*, 2007, 43-48.
- [6] Jian, H., "Design of City-Bus Intelligent Control System Framework", *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, 2006, 2307 – 2311.
- [7] Macario G., Torchiano M. and Violante M., "An In-Vehicle Infotainment Software Architecture Based on Google Android", *2009 IEEE International Symposium on Industrial Embedded Systems*, 2009, 257-260.
- [8] Open Handset Alliance [Online]. Available: <http://www.openhandsetalliance.com/>, Sep. 12, 2016.
- [9] Android Architecture [Online]. Available: http://elinux.org/Android_Architecture, Sep. 13, 2016.
- [10] Honeycomb Android Operating System [Online]. Available: <https://developer.android.com/about/versions/android-3.0-highlights.html>, Sep. 11, 2016.
- [11] Android Ice Cream [Online]. Available: https://tr.wikipedia.org/wiki/Android_Ice_Cream_Sandwich, Sep. 13, 2016.
- [12] Android Interfaces and Architecture [Online]. Available: www.source.android.com/devices/index.html, Sep. 11, 2016.
- [13] Latifa, E. And Ahmed, E. K. M., "Android: Deep look into Dalvik VM", *2015 5th World Congress on Information and Communication Technologies (WICT)*, 2015, 35-40.
- [14] ART and Dalvik [Online]. Available: <https://source.android.com/devices/tech/dalvik/>, Sep. 02 2016.
- [15] Türk, E., "Düşük Güçlü Bluetooth ile Akıllı Durak Bilgilendirme Sistemi", *EEB 2016 Elektrik-Elektronik ve Bilgisayar Sempozyumu*, 2016, 299-303.
- [16] Koç, N. ve Turan, B. Y., "Gömülü Linux Sistemleri", *Pusula Yayıncılık*, İstanbul, 2011.
- [17] Linux Kernel Archives [Online]. Available: www.kernel.org/, Sep. 14, 2016.
- [18] Radio Layer Interface [Online]. Available: www.wladimir-tm4pda.github.io/porting/telephony.html, Sep. 14, 2016.