

KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ DALI
YÜKSEK LİSANS TEZİ

ANDROID VE WEB TABANLI UZAKTAN ARAÇ DENETLEYİCİ TASARIMI

Muhammed Ali ERBİR

Haziran 2015

TEZ KABUL TUTANAĞI
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalında Muhammed Ali ERBİR tarafından hazırlana “ANDROID VE WEB TABANLI UZAKTAN ARAÇ DENETLEYİCİ TASARIMI” adlı Yüksek Lisans Tezinin, Anabilim Dalı standartlarına uygun olduğunu onaylarım.

Prof. Dr. Hasan ERBAY
Anabilim Dalı Başkanı

Bu tezi okuduğumu ve tezin **Yüksek Lisans Tezi** olarak bütün gereklilikleri yerine getirdiğini onaylarım.

Yrd. Doç. Dr. Halil Murat ÜNVER
Danışman

Jüri Üyeleri

Başkan : (Unvanı, Adı ve Soyadı, İmzası) _____
Üye (Danışman) : (Unvanı, Adı ve Soyadı, İmzası) _____
Üye : (Unvanı, Adı ve Soyadı, İmzası) _____

...../...../2015

Bu tez ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onaylamıştır.

Prof. Dr. Mustafa YİĞİTOĞLU
Fen Bilimleri Enstitüsü Müdürü

SEVGİLİ ANNEME ve BABAMA

ÖZET

ANDROID VE WEB TABANLI UZAKTAN ARAÇ DENETLEYİCİ TASARIMI

ERBİR, Muhammed Ali

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Dalı, Yüksek Lisans Tezi

Danışman: Yard. Doç. Dr. Halil Murat ÜNVER

Haziran 2015, 72 sayfa

Bu çalışmada, IP protokolle haberleşen bir kontrol sistemi tasarlanmıştır. Kontrol edilmek üzere bir otomobil uygulama platformu olarak seçilmiştir. Aracın donanımı ile haberleşmek ve onları kontrol etmek üzere Arduino denetleyici, otomatik marş modülü ve röleler kullanılmıştır.

Hem kablolu hem kablosuz kullanılabilmesi, pratik ve ucuz olması, günlük hayatta çokça kullanılan cihazlarımızla rahatlıkla bağlanabilmesi gibi avantajları sebebiyle IP protokol, haberleşme teknolojisi olarak kullanılmıştır.

Hemen her bilgisayar ve telefon cihazı aracılığıyla rahatlıkla kontrol edebilmesi amacı ile HTML ve Android olmak üzere iki farklı kontrol ara yüz hazırlanmıştır. Sistem gerçek hayatta sıradan bir araca tatbik edilmiş, denenmiş ve başarılı sonuçlar alınmıştır.

Sistemin test edildiği aracın, kapı kilitleri, farları, kornası, dörtlü sinyalleri ve motor çalışma durumu başarılı bir şekilde kontrol edilmiştir.

Anahtar Kelimeler: İnternette otomobil kontrolü, Android ile Arduino haberleşmesi

ABSTRACT

DESIGN OF AN ANDROID AND WEB BASED REMOTE VEHICLE CONTROLLER

ERBİR, Muhammed Ali

Kırıkkale University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering, M. Sc. Thesis

Thesis Supervisor: Asst. Prof. Halil Murat ÜNVER

June 2015, 72 pages

In this study, a control system which communicates with IP protocol was designed. A real automobile is selected as a test platform. To control and communicate with automobile hardware an Arduino controller, an automatic engine starter module and relays are employed.

Because IP protocol has the advantages such as being able to be used both cabled and wireless, being cheap and practical, able to be connected by devices widely used in daily life; it is (IP protocol) selected as communication technology.

In order to be controlled easily with almost all computers and mobile phones, two interfaces, HTML and Android, are prepared.

System is applied on an ordinary automobile. Success is achieved from the tests.

Door locks, lights, signals, horn and motor situation of the test automobile is controlled successfully.

Keywords: Automobile Control over Internet, communication between Android and Arduino.

TEŐEKKÖR

Bu tez alıőmasının baőından sonuna kadar, yapmıő olduėum alıőma ve araőtırmalarda beni bilimsel olarak desteklediėi iin Tez Yöneticisi Hocam Sayın Yard. Do. Dr. Halil Murat ÜNVER'e ve alıőmalarım süresince maddi ve manevi fedakârlıklarda bulunan eőim Ayten ERBİR ve ocuklarım Abdullah Said ile Müfide Betül'e teőekkür ederim.

Tez alıőmamdan insanlıėa hayırlı hizmetler sunulacak őekilde istifade edilmesini temenni ederim.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
1. GİRİŞ	1
1.1. Uzaktan Cihaz Kontrolüne Genel Bir Bakış	1
1.2. Uzaktan Cihaz/Sistem Kontrolünün Gerekliliği ve Faydaları.....	2
1.3. Tez Çalışmasının Amacı	2
1.4. Literatür Taraması	3
1.4.1. Uzaktan Kumanda ve Kontrol için Haberleşme Yöntemleri	3
1.4.2. Gömülü Sistemlere Genel Bir Bakış	7
1.4.3. Kullanıcı Kontrol ve Gözlem Platformları.....	9
2. MATERYAL VE YÖNTEM	13
2.1. Materyal	13
2.2. IP Protokolle Sistem Kontrolü için Gerekli Donanım ve Bağlantıları.....	13
2.2.1. Sistem Kontrolü için Gerekli Donanım.....	13
2.2.2. Arduino Mega Denetleyici Kartı.....	14
2.2.3. Arduino Ethernet Modülü	15
2.2.4. Röle ve Röle Seti.....	16
2.2.5. Kablosuz Erişim Noktası (Wi-Fi Access Point).....	17
2.2.6. IP Protokolle Sistem Kontrolü için Gerekli Donanım Bağlantıları	18
2.3. IP Protokolle Sistem Kontrolü için Gerekli Yazılımın Hazırlanması ve Cihazlara Yüklenmesi	20
2.3.1. IP Protokolle Sistem Kontrolü için Gerekli Yazılımın Hazırlanması..	20
2.3.2. HTML Protokolü.....	21
2.3.3. Arduino Programlama Ara Yüzü	21
2.3.4. Android Programlama	25
2.3.5. Ağ Yapılandırılması	32
2.3.6. Hazırlanan Kodların Yüklenmesi.....	36
2.4. Tüm Donanımın Bir Araya Getirilip Sistemin Çalıştırılması	38
2.4.1. Tasarlanan Gömülü Sistemin Otomobile Bağlanması	38
2.4.2. Gömülü Sistemin Ağ Cihazına Bağlantısı.....	40
2.4.3. Sisteme Enerji Verilerek Sistemin Çalıştırılması.....	40
2.5. Projenin, Uzaktan Kontrolde Uygulanabilirliği	41
2.6. Projenin Maliyeti ve Mevcut Alternatifleri ile Mukayesesi	41
2.6.1. Projenin Maliyeti.....	41
2.6.2. Projenin Mevcut Alternatifleriyle Karşılaştırılması.....	42
3. BULGULAR	45
4. SONUÇ	46
4.1. Sonuçlar.....	46

4.2. Öneriler.....	46
KAYNAKLAR	47
EK 1. LED YAK-SÖNDÜR ARDUINO KODU.....	50
EK 2. ARDUINO ETHERNET WEBSERVER KODU	51
EK 3. BU TEZ ÇALIŞMASINDA ARDUINO HONDA JAZZ İÇİN ÖZEL OLARAK TASARLANMIŞ İNTERNET ÜZERİNDEN ARAÇ KONTROL KODU	54
EK 4. ANDROID STUDIO'DA HAZIRLANMIŞ “SEYYARE001” İSİMLİ UYGULAMANIN KODLARI	64
EK 5. ARAÇ İÇİ ÇOK HATLI İLETİM KABLOSU HATLARIN TAŞIDIKLARI SİNYALLER ÇİZELGESİ	70
EK 6. DONANIMIN 12 V'LUK AKÜDEN ÇEKTİĞİ AKIM BİLGİLERİ ÇİZELGESİ.....	72

ŞEKİLLER DİZİNİ

<u>ŞEKİL</u>	<u>Sayfa</u>
1.1. 12 kanallı bir kablosuz uzaktan kumanda verici ve alıcı sistemi	1
2.1. Arduino Mega Ön ve Arka Cihazları.....	14
2.2. Arduino Ethernet modülü.....	15
2.3. Arduino üzerine Ethernet modülü takılmış hal	15
2.4. Arduino Mega üzerine takılı bir Ethernet modülü, Arduino USB kablosu ile PC'ye, Ethernet modülü Ethernet kablosu ile yerel ağa bağlı durum.....	16
2.5. 8 kanallı röle seti	17
2.6. Tek LAN portlu bir kablosuz erişim noktası (Wi-Fi Access point).....	18
2.7. RJ45 soketli bir Ethernet kablosu	19
2.8. Bir USB kablosu ve uçlarının yakınlaştırılmış görüntüsü.....	19
2.9. Gömülü sistemin devresi.....	20
2.10. Arduino yazılımı ilk açılış ekranı	22
2.11. Arduino yazılımı örnek kodlarından “LED yakıp söndürme” (Blink) açılırken	23
2.12. Arduino yazılımı örnek kodlarında “Ethernet WebServer”ı açılırken.....	24
2.13. Android Studio ilk açılış ekranı	27
2.14. Yeni projenin adının girileceği pencere	28
2.15. Yeni projenin hangi Android sürümlerini destekleyeceğinin seçildiği pencere	28
2.16. Yeni projenin aktivite türünün seçildiği pencere	29
2.17. Varsayılan (default) yeni proje ekranı.....	30
2.18. Seyyare001 isimli Android uygulamasının görsel tasarımı	31
2.19. “repeater” modunda kullanılan ağ cihazı yapılandırma ayarları.....	32
2.20. Arduino kodunda IP yapılandırması	33
2.21. Bir IP kameranın ön ve arkadan görünüşü.....	34
2.22. Projede İnternet erişimi için kullandığım bir 3G – WiFi gateway cihazı	35
2.23. İnternette sisteme bağlanmak için router/gateway IP yönlendirme ayarları	35
2.24. Projede kullanılan Arduino Kodundan bir ekran görüntüsü	36
2.25. Ethernet modülü eklenmiş Arduino Mega Kartı.....	37

ÇİZELGELER DİZİNİ

<u>ÇİZELGE</u>	<u>Sayfa</u>
2.1. Proje maliyet tablosu.....	43

KISALTMALAR DİZİNİ

ADSL	Bakımsız Sayısal Abone Hattı
GPS	Küresel Konumlama Sistemi), Dünya üzerinde herhangi engelsiz bir görüş hattında, dört veya daha fazla uydusu ile her türlü hava koşulunda yer ve zaman bilgileri sağlayan uzay tabanlı uydu navigasyon sistemidir [22]
IDE	Tümleşik Geliştirme Ortamı
IP	İnternet Protokolü
IR	Kızıl Ötesi
LAN	Yerel Alan Ağı
RF	Radyo Frekansı
USB	Evrensel Seri Veri yolu
Web	Birbiriyle bağlantılı, İnternet üzerinde çalışan ve "www" ile başlayan adreslerdeki sayfaların görüntülenmesini sağlayan servis
Wi-Fi	Kablosuz Bağlantı Alanı

1. GİRİŞ

1.1. Uzaktan Cihaz Kontrolüne Genel Bir Bakış

İnsanođlu sürekli etrafındaki nesnelere hayatını kolaylaştırıcı şekilde kullanılmaya başlamıştır. Bunlara basitçe alet denilebilir. Her geçen gün ihtiyacın sevk etmesiyle ve arařtırmalar neticesinde, kullanılan aletler daha gelişmektedir. Elektriđin kullanılmaya başlanması ve yarı-iletkenlerin keşfedilmesinden sonra, bu gelişmeler hayal gücü ile yarışır hale gelmiştir. Genel olarak gelişmiş aletler makine, cihaz, ekipman olarak da adlandırılmaktadır. Hatta birden çok gelişmiş makinelerden müteşekkil gruplara sistem de denilmektedir. Günümüzde hayatın her alanında kullanılmakta olan makinelerin birçok zaman, yanına kadar ulaşarak kullanılmak zorluk teşkil etmektedir ve bazen de imkânsız olabilmektedir. Bu ihtiyaçtan uzaktan kumanda ve kontrol teknolojisi meydana çıkmıştır. Uzaktan kontrol alanında alınmış onlarca patent mevcuttur. [1]



Şekil 1.1. 12 kanallı bir kablosuz uzaktan kumanda verici ve alıcı sistemi

1.2. Uzaktan Cihaz/Sistem Kontrolünün Gerekliliđi ve Faydaları

İnsanođlunun kullanmakta olduđu makinelerin buldukları ortamlar, birçok zaman kullanıcının (bundan sonra operatör denilecek) rahatlıkla bulunabileceđi imkân ve özelliđe sahip deđildir. Ortamın dar olması (su kuyusu), uzak olması (tavandaki lambalar), insanın yaşayacağı imkânlara sahip olmaması (uydular), kontrol edilecek makine sayılarının fazlaca olması (dev bir fabrikadaki aydınlatma cihazları), ortamda canlıya zararlı maddelerin bulunması (kimyasal gazlar veya mayın) gibi sebepler insanođlunu makine ve sistemleri uzaktan kumanda ve gözleme mecbur kılmaktadır. [2]

Bununla beraber sadece mecburiyetten deđil, çok defa da kolaylık, konfor (uzaktan kumandalı TV'ler), hız ve pratiklik (kumandalı bahçe kapıları) gibi sebeplerle uzaktan kontrole başvurulmaktadır.

1.3. Tez Çalışmasının Amacı

Bu tez çalışması sürecinde sistemlerin uzaktan kumanda edilmesi ve kontrolü için deđişik ihtimaller irdelenmiştir. Ekonomik, pratiklik, kapasite ve geliştirilebilirlik açılarından en başarılı olarak;

- Arduino tabanlı bir gömülü sistem tasarlamak,
- Bu gömülü sistemle bir otomobilin istenilen donanımını kontrol ve kumanda etmek,
- Gömülü sisteme de uzaktan İnternet ile ulaşım haberleşmek

esasları tespit edilmiştir. Bu metot ve malzemeler kullanılarak gerçek bir ilk örnek tasarlanması ve Honda Jazz 2004 model bir araca başarılı şekilde uygulanması hedeflenmiştir.

1.4. Literatür Taraması

1.4.1. Uzaktan Kumanda ve Kontrol için Haberleşme Yöntemleri

Kumanda ve gözlem merkezinin, çalışan sistem parçalarından uzak mesafede olmasını ifade eden uzaktan kumandalı sistemlerde, kontrol merkezi ile kontrol edilen arasındaki bilgi alış verişi haberleşme diye tabir edilir. Günümüzde birçok alanda sistemler ve haberleşme düzenekleri elektronik teknolojisi kullanarak tasarlandığından, günümüzde haberleşme sistemlerinin tamamına yakını elektrondur. Ancak elektronik olmayan, diğer deyişle sadece elektrikli devrelerden ibaret olan, kontrol merkezleri ve haberleşme uygulamaları da (odalardaki basit lamba düğmeleri gibi) günümüzde kullanılmaktadır.

Kablolu Haberleşme

Çalışan sistem parçalarıyla, kontrol merkezi arasında iletken kablolar vasıtasıyla yapılan haberleşmeye kablolu haberleşme denir. Genelde, iki ya da daha fazla hat içeren kablolar kullanılır. Bilgi, hatlar arası voltaj farkı ile kodlanarak iletilir. Yakın mesafeler için gerektiğinde çok hatlı kablolar (IDE kablosu gibi) ve paralel haberleşme kullanılır. Mesafe uzadıkça hat sayısının fazla olması zahmetli ve masraflı olacağından seri haberleşme (ADSL hattı gibi) tercih edilir. Genelde aynı bina içinde bulunan masaüstü bilgisayarların arasında kullanılan Ethernet hatları da kablolu haberleşmenin gelişmiş bir örneğidir.

Kablosuz Haberleşme

Kontrol noktası/merkezi ve çalışan sistem parçaları arasına kablo çekmek pratik ve avantajlı olmadığı durumlarda (yeryüzü ile uzay arası uydu haberleşmesi gibi) kablosuz haberleşme kullanılır. Vericiden alıcıya giden sinyalin, iletken tellere ya da herhangi bir ek altyapıya ihtiyaç duymadan, havadan, sudan ya da boşluktan iletilmesi üzerine kurulmuş bir teknolojidir. [3] Sesle kontrol edilen cihazlar, kızılötesi kumandalı TV'ler, RF kullanan araç kumandaları kablosuz haberleşmenin

farklı örnekleridir. Bina içlerinde mobil cihazlarımızı binadaki Ethernet ağına bağlamak için kullanılan Wi-Fi Ethernet teknolojisi de kablosuz haberleşmenin en kabiliyetli örneklerindedir.

RF Haberleşme

Kablosuz Haberleşmenin en yaygın tipidir. Basit kablosuz zillerden, araç kumandalarına, cep telefonlarına, telsizlere varıncaya kadar tüm elektromanyetik dalgalarla haberleşen cihazların kullandığı teknolojidir. Fakat genel olarak RF denince, tek ya da birkaç farklı sinyalden ibaret olan basit ve tek yönlü olan RF haberleşme teknolojileri kastedilmektedir. RF sinyaller boşluk ve havada ilerleyebildikleri gibi, şeffaf olmayan maddelerde de ilerleyebilirler. Fakat değişik maddelere rast geldiklerinde değişik miktarda yansıma ve kırılmaya uğradıklarından farklı ortam ve mesafeler için farklı RF sistemler tasarlanmaktadır. [4]

IR Haberleşme

Ev cihazları kumandasında kullanılan en yaygın haberleşme yöntemidir. [5] Gözle görünmeyen Kızılötesi (IR) ışık sinyalleri kullanılarak genelde tek yönlü olarak kullanılırlar. Güneş bizzat bir Kızılötesi ışık kaynağı (IR gürültü kaynağı) olduğundan güneşli ortamda bu teknoloji tercih edilmemektedir. Aynı zamanda IR sinyaller ışık özelliği sergilediğinden doğrusal yayılırlar. Ancak şeffaf maddelerde ilerleyebilirler. Bu sebeple de şeffaf olmayan nesnelere (duvar gibi) geçemezler. Bu da ev ve işyerlerinde aynı odada birden fazla olmamak kaydı ile aynı model cihazlardan ve aynı model kumandalardan binlerce adet üretilip kullanılabilir. Aynı model cihaz için (mesela bir TV) farklı sinyaller gönderen kumandalara ihtiyaç duyulmaz.

Bluetooth Haberleşme

2.4 GHz civarında çalışan özel bir RF haberleşme teknolojisidir. Çift yönlü bilgi alış verişini destekleyen 10 m'den kısa mesafelerde yaygın kullanılan bir teknolojidir. [6] Cep telefonu ile bluetooth kulaklık arasındaki ses alış verişi en yaygın bluetooth uygulamasıdır. [7]

Genel Telefon Ağı (PSTN)

Yıllardır kablolu telefonlarda kullanılmakta olan haberleşme teknolojisidir. Son yıllarda kablolu telefon hatlarına kablosuz telefonlar ve bilgisayarlar da bağlanabilmektedir. Kablolu ve mobil tüm telefonları içine alan haberleşme ağına Genel Aktarmalı Telefon Şebekesi (PSTN) tabir edilir. [8]

Mobil Telefon Ağı Teknolojisi

İlk zamanlarda sadece kablosuz mobil telefon (araç/cep telefonu) olarak kullanılmaya başlanan bu RF teknolojisi, şimdi her türlü bilgi alışverişi için yaygın olarak kullanılmaktadır. 1G, GSM, 3G, HSDPA, 4G gibi değişik standartları tanımlanmış ve kullanılmaktadır. [9]

IP Protokolü ile Haberleşme

Bilgisayarları ve dengi, gelişmiş cihazları dublex (çift yönlü) haberleştirmek için kablolu veya kablosuz olarak kullanılabilen bir teknolojidir. Bilgi alan/veren tarafların kendine has adresi vardır ki buna IP adresi denir. IP adresleri her ağ grubu için ayrı yapılır. Dünya genelinde tek olan IP numaraları var olduğu gibi, alt ağlarda kullanılan bazı IP adresleri vardır ki binlerce defa, birbirine direk bağlı olmayan alt ağlarda aynı anda kullanılabilir. Bilgiler dijital olarak kodlanır ve paketlenir. Paketlerin büyüklüğü sınırlı tutulur. Bu projede de gelişmiş özellikleri ve dünyada yaygın olarak kullanılıyor olması sebebiyle IP protokolü kullanıldı. [10]

Ethernet Protokolü

Aynı kurum, bina veya yerleşke içinde olan, yerel ağlardaki bilgisayarları, IP Protokol kullanarak, birbirine bağlamak için kullanılan, RJ45 uçlu, 8 hatlı kablolar ile çalışan haberleşme teknolojisidir. Ethernet ağlarında, İnternet bağlantısı olmayabilir ya da bir gateway cihazı ile İnternete bağlantı sağlanabilir.

Her ne kadar Ethernet bağlantılı cihazların insanlar tarafından IP adresleri göz önüne alınmakta ise de Ethernet ağını sağlayan ağ cihazları hem IP adreslerini hem de her cihaza mahsus dünya üzerinde tek olduğu varsayılan MAC adreslerini de kullanmaktadır.

Projemizde Ethernet protokolü de kullanıldığından, kullanılan cihazların hem IP adresleri hem de MAC adresleri vardır. [11]

Kablolu ve Kablosuz Ethernet Bağlantısı

Ethernet ağındaki cihazlar kablolu olarak haberleşebildikleri gibi, wireless access point (kablosuz erişim noktası) ismini verilen cihazlar kullanılarak, mobil cihazlar da ağa bağlayabilir ve ağdaki kablolu veya kablosuz bağlı diğer cihazlarla haberleşebilirler.

İnternet

Dünya genelindeki (hatta uzaydan da olabilir) protokolü destekleyen tüm cihazların / yerel ağların tek bir IP adresi kullanarak birbiri ile haberleştiği teknolojidir. [12]

Kablolu İnternet

Dünya genelinde servis veren İnternet adreslerine ulaşmak için kullanılan bağlantı kablo ile yapılır. Bu tür bağlantı kablolu telefon hattı ile ya da Fiber / ADSL / uydu vs. ile İnternete bağlı olan Ethernet ağına (cihazına) kablolu bağlantı ile yapılabilir.

Kablosuz İnternet

Kablo kullanmadan İnternete bağlanabildiğimiz teknolojidir ki bu da genelde, aşağıda açıklanacağı gibi, iki şekilde olur.

Kablosuz Ethernet ile İnternet

İnternet bağlantısına sahip bir Ethernet ağına (cihazına) kablosuz bağlantı yaparak İnternete kablosuz bağlanabiliriz. Kablosuz Ethernet ağına kısaca WLAN denir. (13)

Mobil Hatlar Üzerinden İnternet

Mobil hatlar üzerinden İnternete kablosuz bağlantı yapılabilen teknolojidir. Önceleri sadece sesli telefon görüşmesi için tasarlanan ve kullanılan cep telefonu hatları, daha sonra yazılı mesaj ve İnternet bağlantısı hizmeti de sunmaya başladılar. Cep telefonu hattı ile İnternete bağlantı günümüzde çok yaygın kullanılmaktadır.

1.4.2. Gömülü Sistemlere Genel Bir Bakış

Elektriğin ilk kullanılmaya başlamasından bu yana tasarlanan devreler ve cihazlarımız sürekli olarak gelişmekte, her geçen gün daha kabiliyetli hale gelmektedir. Umumun ihtiyacı olarak tasarlanan ve sunulan binlerce elektrikli veya elektronik sistemler şu an hayatımıza nüfuz etmiş durumdadır. Bilgisayarlar, akıllı cep telefonları gibi umumun ihtiyacı düşünülerek üretilen cihazların yanı sıra, birçok

defa bazı hususi maksatlar için daha has görevlere sahip, daha ekonomik, daha hızlı, küçük kitlelere hatta bazen şahıslara has elektronik devreler tasarlamaya da ihtiyaç duyulmaktadır. Bu tür devre ve sistemlere gömülü sistemler denir. [14]

Yaygın Kullanılan Gömülü Sistem Platformları

Gömülü sistemlerin ana parçası mikro denetleyicilerdir. Günümüzde gömülü sistemler tasarlayan onlarca firma ve bunlara mikro denetleyici / mikro işlemci tedarik eden birçok firma vardır.

Samsung, Toshiba gibi kendi gömülü sistemleri için kendi işlemcisini tasarlayan firmalar var olduğu gibi; genel amaçlı kullanım için umuma mikro denetleyici üreten Atmel ve Microchip firmaları, mikro işlemci üreten Intel ve AMD firmaları sektörün en meşhur olanlarıdır.

Projemizde kullanılan Arduino Platformu da Atmel firmasının ürettiği ATMega isimli mikro denetleyicileri kullanan bir platformdur [15].

Arduino Mikro Denetleyici Platformunun Alternatifleri ile Karşılaştırılması

PIC ve Atmel şu an için piyasada kullanılan en yaygın, temini kolay, örnek ve kaynak çalışmaların çok olduğu, kullanımı pratik ve kabiliyetleri yüksek olan iki mikro denetleyici modelleridir. Arduino kartlar Atmel mikro işlemcileri barındıran bir platformdur. Doğrudan USB kablosu ile bilgisayarla haberleşebilen, C tabanlı ve tamamen ücretsiz yazılımı ile programlamada kolaylık sunan, geniş kütüphaneleri ücretsiz paylaşılan bir platform olmasından ötürü projede tercih edilmiştir.

1.4.3. Kullanıcı Kontrol ve Gözlem Platformları

Kullanıcı Kontrol ve Gözlem Platformlarına Genel Bakış

Tasarlanan tüm sistemlerde iç ve dış olarak iki taraf vardır. Sistemin içinde çoğu zaman sistemin dışından görünenden daha karmaşık fonksiyonlar bulunmaktadır ki bunlar sadece tasarımcıyı ilgilendirir. Bir de sistemlerin dıştan görünen tarafları vardır ki burası da bu sistemi kullananları ilgilendirir. Sistem mühendisleri tasarımı yaparken şunu göz önünde bulundururlar: Sistemi kullanan kişiler (operatörler, müşteriler) genelde sistemin iç yapısını anlamaktan uzakta olacağından, sistemi kullanan kişilerin olabildiğince kolay bir şekilde sistemle etkileşime girmeleri arzu edilir.

İşte bu nedenle, değişik elektrikli/elektronik sistemleri hemen her insan kolayca öğrenebilsin ve kullanabilsin diye ortak gözlem ve kontrol platformları tasarlanmıştır. Windows, Android gibi işletim sistemi platformları, çok karmaşık ve profesyonel bir yapıda olan bilgisayarları, cep telefonlarını, tabletleri okuma yazma bilen hemen her insanın kullanabilmesi amacıyla, kontrol ve gözlemi olabildiğince kolay hale getirmek için tasarlanmıştır. Aynı şekilde bilgi çağında teknoloji gövdesinin dolaşım sistemi olan İnternetin okuma yazma bilen herkes tarafından kullanılabilmesi için İnternet tarayıcıları olarak isimlendirilen uygulama platformları tasarlanmıştır.

Sadece Kişisel Bilgisayarlarda Kullanılan İşletim Sistemleri

Günümüzde sadece kişisel bilgisayarlarımızda kullanılan işletim sistemleri vardır ki: Apple firmasının sunduğu Mac OS, Microsoft'un sunduğu Windows, Sun firmasının SunOS ürünü örnek olarak verilebilir.

Sadece Cep Telefonlarında Kullanılan İşletim Sistemleri

Günümüzde sadece cep telefonlarında kullanılan işletim sistemleri vardır ki: Apple firmasının sunduğu iOS, Microsoft'un sunduğu Windows Mobile, Google Firmasının sunduğu Android ürünü örnek olarak verilebilir.

Haberleşme Kabiliyetli Cihazlarda Kullanılabilecek Ortak Platformlar

Tek başına işletim sistemi olmamasına karşın farklı işletim sistemlerinde ve farklı cihazlarda çalışabilmesi özelliğine sahip platformlarda vardır ki bunlara; Java, HTML, FLASH teknolojileri örnek verilebilir.

Kullanıcı Kontrol ve Gözlem Platformlarının Karşılaştırması

Proje başında geniş bir araştırma yapıldı. Bu araştırmada, son kullanıcı ve gözlem platformlarından hangisinin

- En yaygın portföye sahip olduğu,
- En çok farklı cihaz tarafından desteklendiği,
- Düşük özelliklere sahip makinelerce bile çalıştırılabileceği,
- Eski ve yeni versiyonlarının birbirine uyumluluğunun yüksek oluşu,
- Kod hazırlamanın alternatiflerine nispeten kolaylığı

gibi vasıflara bakılarak Windows, Linux, Android, Flash, Java, HTML teknolojileri karşılaştırıldı.

Android ve HTML Platformlarının Tercih Edilmesinin Sebepleri

Projenin başında yapılan araştırma sonucunda şu karara varıldı: Son kullanıcı ve gözlem platformlarından en yaygın portföye sahip, en çok farklı cihaz tarafından desteklenen, düşük özelliklere sahip makinelerin bile işleyebileceği, eski ve yeni versiyonlarının birbirine uyumluluğu yüksek olan, kod hazırlamanın alternatiflerine

nispeten zor olmadığı gibi vasıflara sahip olması sebebiyle HTML teknolojisinin projede kullanılması isabetli olacaktır.

Bununla beraber HTML ile rahatlıkla etkileşebilen ve her geçen gün popülaritesi artan Android işletim sistemi için de bir uygulama yapmanın, sistemi gözlem ve kontrol ederken cep telefonu ve tablet kullanımında daha pratik olacağı kanaatine varılarak, bir de Android uygulama hazırlandı.

Araçlarda İnternet Kullanımı

Araçlarda İnternet kullanımı, yakın zamanlara kadar gündemde olmayan bir uygulama idi. Son yıllarda cep telefonlarında olduğu gibi, araçlarda da İnternet bağlantısı özelliğinin yaygınlaştığı görülmektedir [24]. Mobil İnternet teknolojisinin canlanması, yaygınlaşıp, finanse edilebilir olması ile araçlarda İnternet kullanılmaya başlamıştır [16]. BMW firması 2010 yılında Avusturalya’da sattığı araçların araç bilgisayarlarına İnternete bağlanabilme özelliğini opsiyonel olarak eklemiştir [17].

Araçlarda İnternet kullanımının teknolojik yönü yeni bir araştırma sahası olduğu gibi, araçların ve araçlardaki teknolojinin insanın sosyal hayatına olan etkileri de çok sosyal bilimcilerce detaylı araştırılmayan konulardandır [18]. Hâlbuki haritalar, yol güzergâhlarındaki kaza ve yol durumu, hava durumu, nöbetçi hastane ve eczane vs. gibi anlık güncelleme gerektiren bilgilere araçlardaki haberleşme olanakları yüksek teknolojik donanım ile vakıf olunması, araç ile seyahat edenlerin hayatını planlarken yolculukları hangi kategoride değerlendireceğini doğrudan etkiler. Araçlarda İnternet bağlantısının bulunması sayesinde, bulunduğu ortamdaki trafiğin, hava şartlarının ve diğer etkenlerin anlık bilgileri aracı kontrol eden kişi veya mekanizmalarca bilinmesi aracın bizzat kendisi ve yolcuları için avantajlı olacaktır [24].

İnternet bağlantısı olan araç ile beraber diğer araçlarda da İnternet bağlantısının bulunması ise gerçek zamanlı olarak ortak bir düzenleme ve planlama yapılması; trafiğin akışı, güvenliği, hızı, yoğunluğu gibi durumları kolayca kontrol etme imkânı sağlanacaktır [16] [26].

Fung ve Leung'un kitaplarında [19] açıklandığı gibi İnternet artık hayatımızın bir ayrılmaz bütünü olmuştur. Avantaj ve dezavantajları ile hayatımızın her kısmını etkilemektedir. Hatta çoktan taşıtlarda (yük taşıtlarında) İnternet kullanımının ziraat, balıkçılık ve diğer önemli endüstrilere pazar, fiyat, nakliye ücretleri üzerindeki etkileri araştırılmaya başlanmıştır [20, 21].

Günümüz itibariyle İstanbul Büyükşehir Belediyesi, toplu taşıma araçlarında aktif olarak İnternet kullanmaktadır. İnternet donanımlı bu toplu taşıma araçlarının araçların, rotaları önceden planlanmakta ve umuma ilan edilmektedir. Bununla beraber, GPS bilgisi kullanılarak, ne zaman nerede buldukları ve yakınlaştıkları duraklara tahmini varış süreleri gerçek zamanlı olarak hesaplanmakta ve akıllı telefonlara ücretsiz kurulabilecek ilgili yazılımlar ile İnternette sürekli öğrenilebilmektedir [23].

Araçlardaki İnternet bağlantısı gerçek zamanlı ortak planlamalara imkân sağladığı gibi, kişisel olarak da kullanıcıların uzaktan araç hakkında bilgi edinmesine ve aracı donanımını kontrol etmelerine imkân sağlamaktadır [25].

Araçlardaki İnternet bağlantısının hangi teknolojiler kullanılarak yapıldığı ise başlı başına bir araştırma konusudur [24]. Kablosuz İnternet için kullanılan teknolojilerin farklı avantaj ve dezavantajlarının bulunması, araçlarda İnternet bağlantısının birden fazla yöntemle yapılabilir olmasını da gerekli kılmaktadır [24].

2. MATERYAL VE YÖNTEM

2.1. Materyal

Literatür taramasının neticesi olarak Arduino Mega denetleyicili, Ethernet destekli bir gömülü sistem tasarımı ile otomobilin donanımının kontrol edilmesi düşünülmüştür. Bu gömülü sistem içinde hem Android cihazlar ile hem de herhangi bir İnternet tarayıcısı ile haberleşebilecek bir sunucu yazılımı kodlanacaktır. Gömülü sistemin Android cihazlarla kolayca haberleşebilmesi için bir de Android uygulamasının hazırlanması kararlaştırıldı. Bu hedefe ulaşılabilmesi için şu malzemeler gerekli görülmüştür:

- Arduino Mega Mikro işlemci kartı
- Arduino Ethernet modülü,
- Arduino 8 kanallı röle seti,
- Kablosuz Erişim Noktası Cihazı
- Ethernet kablosu
- İletken kablolar
- Breadboard
- Arduino Yazılımı
- Android Yazılımı
- Yazılımları Koşturabilecek bir Bilgisayar

2.2. IP Protokolle Sistem Kontrolü için Gerekli Donanım ve Bağlantıları

2.2.1. Sistem Kontrolü için Gerekli Donanım

Donanımın ana kartı birçok gömülü sistemlerde olduğu gibi bir mikro denetleyicidir.

- Hazır ve Geniş kütüphaneler sahibi olması,
- Kolay temin edilebilirliği,
- Az kablolama gerektirmesi,

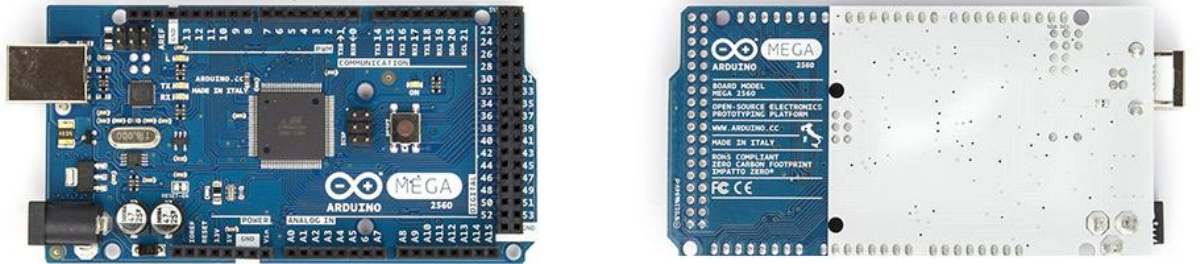
- PC ile aracı cihaz olmadan USB kablosu kullanılarak doğrudan haberleşebilmesi,
- Çok sayıda analog ve dijital porta sahip olması

gibi nedenlerle Arduino Mega denetleyici kartı tercih edilmiştir. Kontrol edilecek sisteme İnternet veya Ethernet üzerinden ulaşmak arzu edildiğinden, sistemin Ethernet ağına bağlanabilmesi için bir Arduino Ethernet modülü kullanılmıştır. Kullanıcıdan gelen sinyallere göre ya da otomatik olarak mikro denetleyicinin yüksek akım çeken araç donanımını kontrol edebilmesi için transistörler ve röleler kullanılmıştır.

Araç civarında kablosuz ağ yayabilmesi ve mevcutsa yakınlarda bulunan İnternet bağlantısı ile kontrol sistemi arasında köprü vazifesi görerek bağlantı kurabilmesi için bir kablosuz erişim noktası (Wi-Fi Access point) kullanıldı. Bu modüllerin detayları aşağıda izah edilecektir.

2.2.2. Arduino Mega Denetleyici Kartı

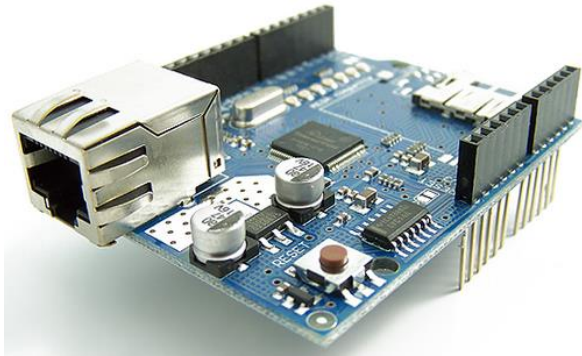
Arduino kartları Atmel mikro denetleyici tabanlı, kolay devre kurulmasını amaçlayarak tasarlanmış bir platformdur. Arduino kartlara rahatlıkla bağlanabilecek çok sayıda hazır modül tasarlanmış olup, hepsinin de kütüphaneleri ve kodlama örnekleri mevcuttur. Bu sayede her hangi bir modülünü arduinoya bağlamak ve örnek bir çalışan devre gerçekleştirmek oldukça pratiktir. Kullanıcılar kendi çaba ve kabiliyetleri ile bu hazır çalışabilen kart ve modüllerin işlevlerini geliştirebilirler.



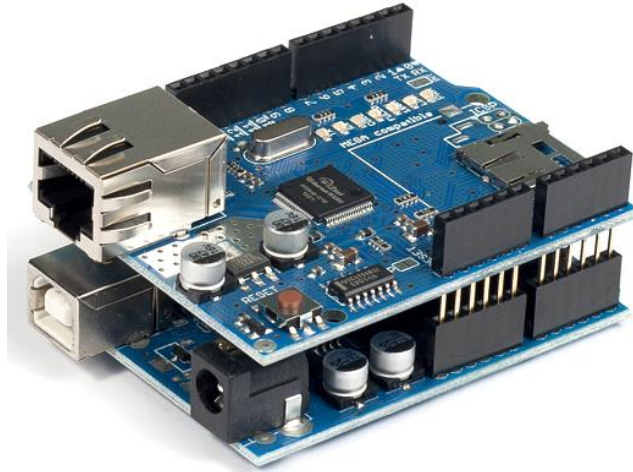
Şekil 2.1. Arduino Mega Ön ve Arka Cihetleri

2.2.3. Arduino Ethernet Modülü

Arduino Ethernet modülü, Arduino kartların bazılarında (mega, uno gibi) doğrudan üzerine monte edilebilecek şekilde tasarlanmış, herhangi bir Ethernet Ağ bağlantısına (LAN), RJ45 soketli Ethernet kablosu ile, doğrudan bağlanabilecek şekilde tasarlanmış bir ara modüldür.



Şekil 2.2. Arduino Ethernet modülü



Şekil 2.3. Arduino üzerine Ethernet modülü takılmış hal

Arduino içine uygun kod yüklendiğinde ve Ethernet modülü ağa kablo ile bağlandığında, Arduinoya ağ içinden doğrudan IP adresi ile ulaşabilmektedir. Ağ

dışından ise Ethernet modülün bağlandığı ağın, router/gateway cihazına uygun IP yönlendirmesi (IP forwarding) yapılarak yerel ağ dışından veya İnternette herhangi yerden rahatlıkla Arduinoya ulaşılabilen, bilgi alış verişi yapılabilmektedir.



Şekil 2.4. Arduino Mega üzerine takılı bir Ethernet modülü, Arduino USB kablosu ile PC'ye, Ethernet modülü Ethernet kablosu ile yerel ağa bağlı durum

2.2.4. Röle ve Röle Seti

Röle küçük bir akım ve voltaj ile yüksek akım ve voltajı kontrol etmeye yarayan bir elektrik devre elemanıdır. Projede kullanılan modeller kontrol girişine 12 V_{DC} ve 5mA akım verilerek, 230V 10A'e kadar olan akım ve voltajı anahtarlayabilecek kapasiteye sahiptirler.



Şekil 2.5. 8 kanallı röle seti

Kullanılan röle setinin 1 adet VCC isimli $12V_{DC}$ güç girişi ve 1 adet GND isimli 0V girişi vardır. Setin içindeki 8 rölenin her birine has IN1, IN2, şeklinde isimlendirilmiş ayrı girişler vardır ki, bu girişlere 0V bağlandığında röleler ON (iletir) konuma, 5V verildiğinde ise röleler OFF (iletmez) konuma geçerler.

Rölelerin çıkışında 3'er pin vardır ki, ortadaki "common" pin, röle pasif konumda solundaki NC pinine, röle aktive edilince ise sağındaki NO pinine temas eder.

2.2.5. Kablosuz Erişim Noktası (Wi-Fi Access Point)

- Sistemin civarında kablosuz ağ yayını yapabilme,
- Herhangi bir Wi-Fi destekli cihazla sisteme 200 m'ye kadar mesafeden doğrudan kablolu (LAN) veya kablosuz ağ (WLAN) ile bağlanabilme,
- İnternet bağlantısı mümkün ise yerel ağ dışından gelen komutları arduinoya yönlendirme

Vazifelerini yapması için sisteme bir adet kablosuz erişim noktası eklenmiştir. Proje hareketli bir araçta gerçekleştirileceğinden, kablolu ağ bağlantısına gerek duyulmayacağı varsayılarak tek kablolu portlu bir cihaz ile iktifa edilmiştir.



Şekil 2.6. Tek LAN portlu bir kablosuz erişim noktası (Wi-Fi Access point)

2.2.6. IP Protokolle Sistem Kontrolü için Gerekli Donanım Bağlantıları

Sistemin güç gereksinimini doğrudan otomobilin 12V'luk aküsünden temin edilmiştir. Zaten kullanılan arduino ve ethernet modülü, röle seti ve kablosuz erişim noktası hepsi de 12V besleme ile çalışabilmektedir.

Ethernet modülü Arduino üzerine tam oturacak şekilde tasarlandığından Arduino ile bağlantısı oldukça basittir. Şekil 2.4'te görünmektedir.

Kablosuz erişim noktası Arduino Ethernet modülüne bir Ethernet kablosu ile bağlanmıştır.



Şekil 2.7. RJ45 soketli bir Ethernet kablosu

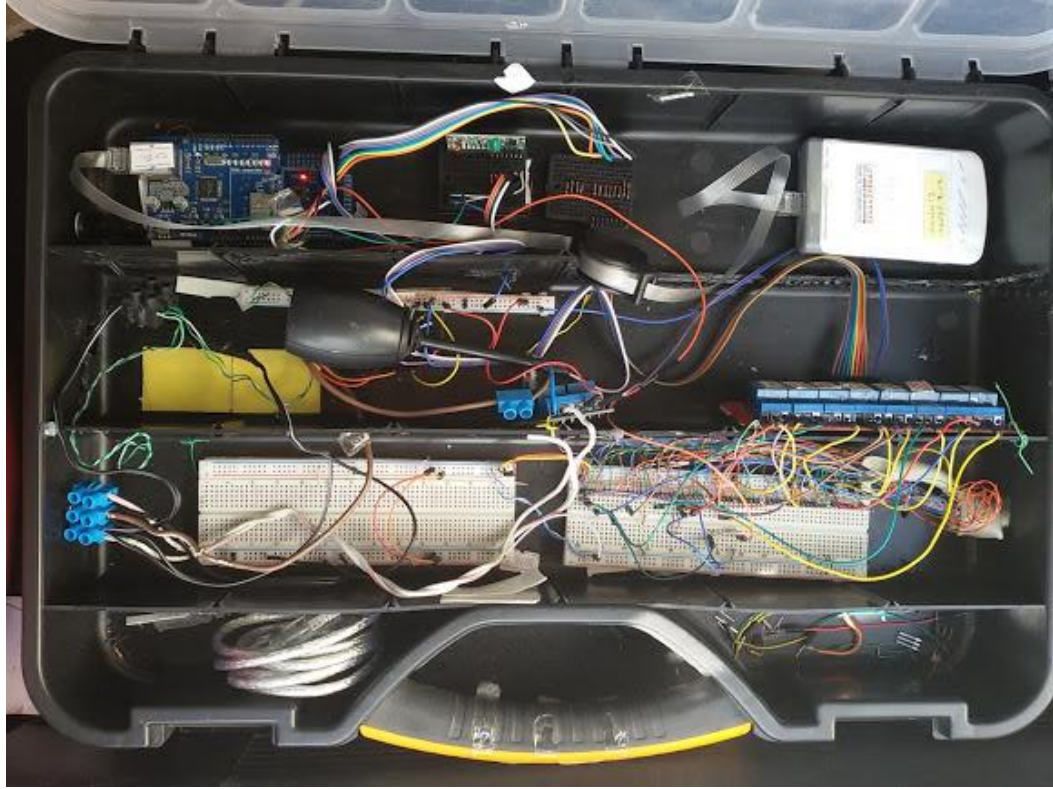
Arduino denetleyici kod aktarmak için bilgisayara bir USB kablo ile bağlanmalıdır.



Şekil 2.8. Bir USB kablosu ve uçlarının yakınlaştırılmış görüntüsü

Aracın, sadece motorunun çalıştırılması haricinde, tüm donanımını kontrol için (kapı kilitleri, farlar, korna, sinyaller vs.) sadece 0V veya 12V sinyali gönderilmesi yeterli olmaktadır. Ancak motoru çalıştırmak için marş motorunu aktive etmek ve araç çalıştığında marş motorunu devre dışı bırakmak için ayrıca bir devre tasarlanmıştır. Bu motor marş devresine, sadece motorun durumunu değiştir sinyali gönderilecektir. Bu devre, motor durgun halde ise motoru otomatik çalıştıracak, motor çalışır halde ise motoru otomatik olarak durduracaktır.

Bu motor marş modülünün çalışma prensipleri ve bağlantıları konu ile doğrudan alakalı olmadığından bu raporda detaylandırılmayacaktır. Dolayısıyla bu modüle, kontrol sisteminden tek bir sinyal kablosu çekilmesi yeterli olacaktır. Arduino ile röleler arası bağlantı ise basit iletken kablolar ile yapılmıştır.



Şekil 2.9. Gömülü sistemin devresi

2.3. IP Protokolle Sistem Kontrolü için Gerekli Yazılımın Hazırlanması ve Cihazlara Yüklenmesi

2.3.1. IP Protokolle Sistem Kontrolü için Gerekli Yazılımın Hazırlanması

IP Protokolü, tüm dünyada İnternet bağlantısı için kullanılan, kaynak makineden gönderilen bilgi paketlerinin hedef makineye sayılardan oluşan IP adresleri kullanılarak ulaştırılmasını esas alan haberleşme protokolüdür. OSI modelinde ağ katmanındadır.

İlk dünya genelinde kullanılan IP4 protokolünde, dünyadaki tüm cihazlara, sayıca fazla olmalarından, özel birer IP adresi verilemeyeceği düşünülerek, bazı IP adresleri sadece yerel ağlarda kullanılmak üzere ayrılmıştır. 10.0.0, 172.16.0, 192.168 gibi bazı değerlerle başlayan bu özel ağ adresleri (private network addresses) dünyada binlerce yerel ağda (LAN) aynı anda kullanılabilir. Zira yerel ağdaki tüm bilgisayarlar ağ dışından tek bir cihazmış gibi görünmekte ve İnternete bağlanırken ağdaki tüm cihazlar aynı IP adresini kullanmaktadır. Ağ içindeki farklı cihazlara dışarıdan ayrı ayrı ulaşabilmek için ise tüm ağın ortak kullandığı tek bir IP adresinin arkasına port numarası denilen bir değer eklenmektedir.

IP protokolde her cihazın bir IP adresi vardır. Mesela projedeki kontrol ünitesi olan Ethernet modülü eklenmiş Arduino denetleyicinin IP adresi 192.168.2.100, kablosuz erişim noktasının IP adresi 192.168.2.2, 3G-WiFi gateway cihazının IP adresi 192.168.2.1, IP kameranın IP adresi 192.168.2.3 tür.

2.3.2. HTML Protokolü

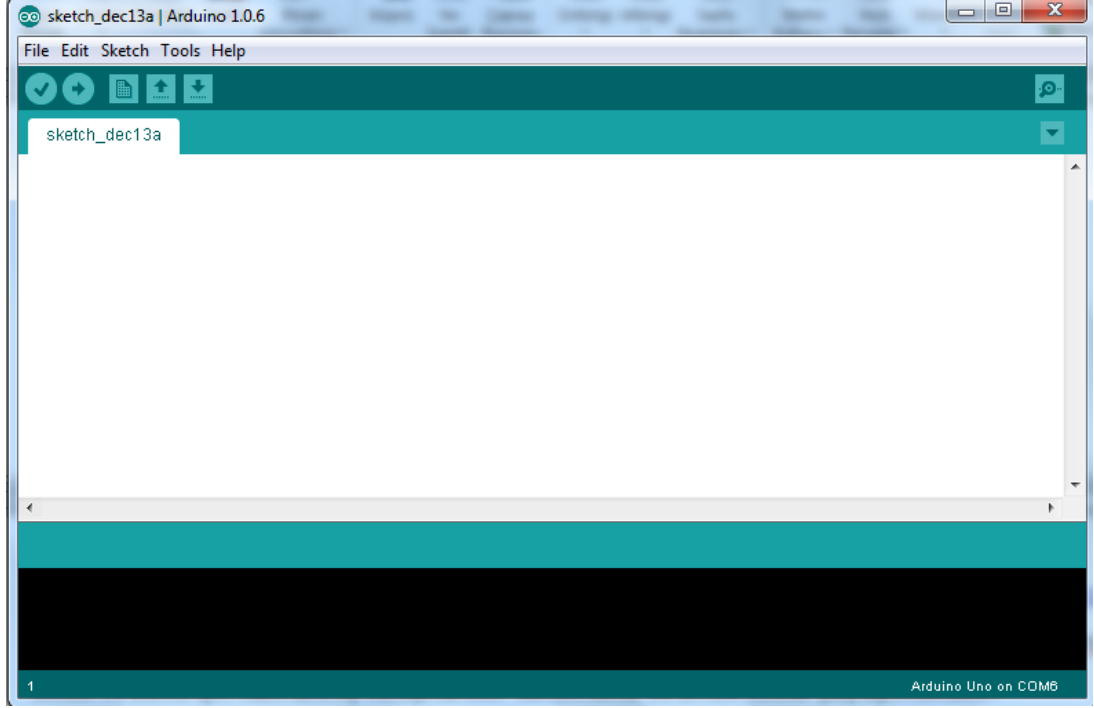
HTML ifadesi İngilizce “Hyper-Text Markup Language” kelimelerinin baş harflerinden alınmıştır. İnternetin kullanıcılar seviyesinde iletişim kuralıdır.

Genel olarak sıradan bilgisayar kullanıcıları İnterneti en kolay olan uygulama katmanında kullanırlar. Bu projede de sistem kontrolü için, en üst ve en kolay olan, umuma hitap eden HTML programlama tercih edilerek, sistemin içine HTML kodu gömüldü. Böylelikle, kullanıcı herhangi bir İnternet tarayıcı ile sisteme bağlanıp kontrol ve kumanda yapabilecektir.

2.3.3. Arduino Programlama Ara Yüzü

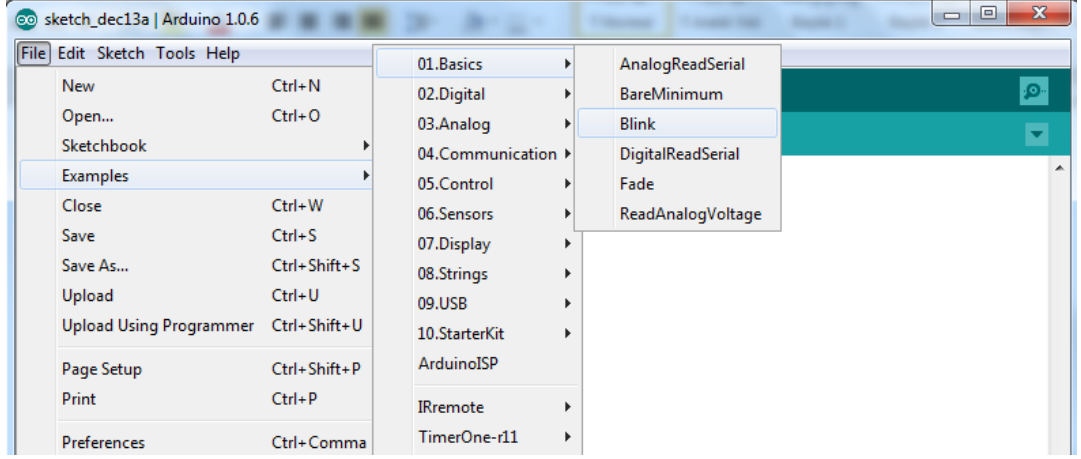
Arduino kartlarını programlamak üzere hazırlanmış ve ücretsiz olarak, <http://www.arduino.cc/> adresinde sunulan yazılımın ismi de yine Arduino'dur.

- Bahsi geen siteden Arduino programının setup (kurulum) dosyası indirilip bilgisayara kuruldu.



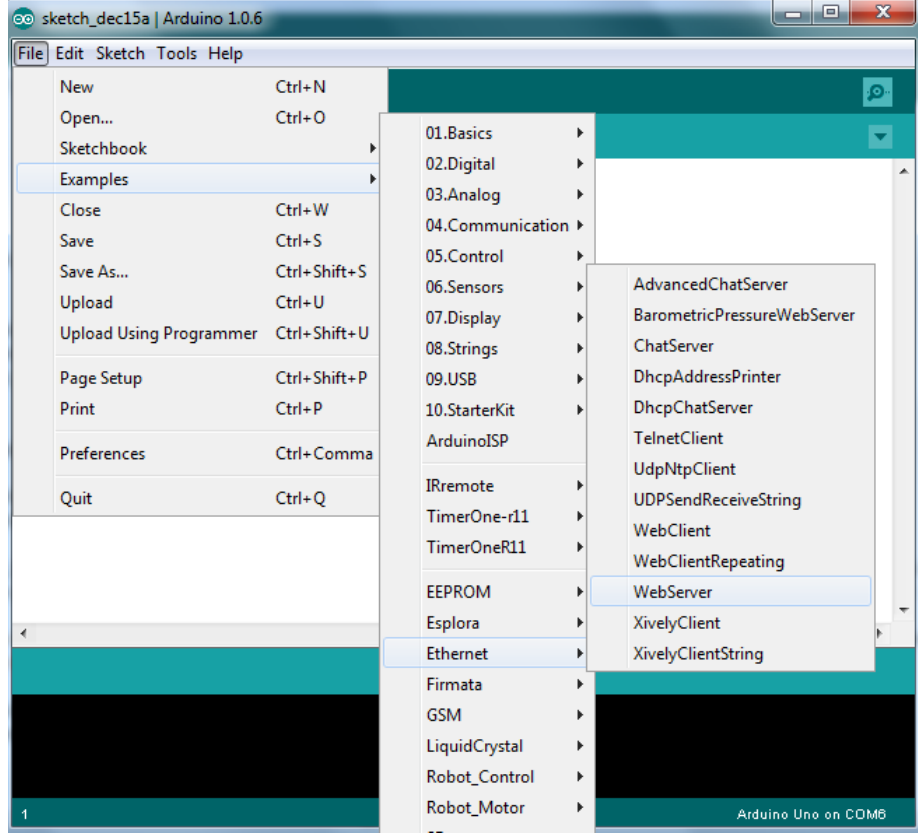
Şekil 2.10. Arduino yazılımı ilk açılışı ekranı

- Arduino Mega kartı bir usb (yazıcılar için olan tip) kablo ile doğrudan bilgisayara bağlandı.
- Deneme için yazılımın örnek kodlarından LED yakıp söndürme kodunu açılıp ve Arduino'ya yüklendi. Bu LED yakıp söndürme Arduino kodu ek olarak sunulmuştur.



Şekil 2.11. Arduino yazılımı örnek kodlarından “LED yakıp söndürme” (Blink) açılırken

- Hiçbir ek bağlantı yapmadan Arduino üzerindeki 13 nolu dijital pine bağlı LED’in, kodda belirtildiği şekilde yanıp söndüğü gözlemlendi. Bu sayede sistemin program atma ve Arduinoyu çalıştırma kısmının sorunsuz çalıştığı test edilerek gözlemlendi.
- Daha sonra Arduino Mega kartımın üzerine monte edilen Ethernet modülü için hazırlanmış olan örnek kod açılarak yüklendi ve test edildi. Bu Arduino WebServer kodu ek 2 olarak sunulmuştur.



Şekil 2.12. Arduino yazılımı örnek kodlarında “Ethernet WebServer”ı açılırken

En son olarak ise aylarca süren AR-GE sürecinde, defalarca tasarım, test, geliştirme, sorun giderme işlemlerini tekrarlayarak başarılı bir şekilde istenilen neticeyi veren Arduino koduna ulaşıldı. Bu kod da teze ek 3 olarak sunulmuştur.

Bu sistemin çalışma algoritması şudur:

- Aracın marş sistemi, farları, kapıları, sinyalleri, kornası gibi donanımların arduinonun hangi pinine nasıl bağlandığı ile koda başlanmıştır.
- Programda tamsayı tipinde zamanlayıcılar, donanımın durumlarını hatırlamak için boolean değişkenler gibi bazı değerler ve değişkenler tanımlanmıştır
- Programın ana fonksiyonunda asıl olarak ağa bağlı başka cihazlar ve İnternet üzerinden gelebilecek komutları bekleyen ve gözleyen bir döngü çalışmaktadır.
- Fakat bununla beraber, pratiklik açısından sadece ağ cihazları ile değil, daha basit kumanda, buton vs. gibi yöntemlerde sisteme entegre edilebilmesi için döngü

içerisinde sistem, IP protokol dışında da giriş kabul edebilecek şekilde düzenlemeler yapılmıştır.

- IP protokolü haricinde de mesela basit bir RF kumanda ile veya IR kumanda ile de sistem aktive edilebilmektedir. Bu özellik sadece pratik olduğu için ilave edilmiştir. Zira araç kullanıcısının her vakit HTML destekli veya Android bir cihaz bulması ve bunu kullanması pratik olmayacağı düşünülmüştür.
- Sistem hem kullanıcı isteği ile şekillenmekte hem de bazı otonom özelliklerle çalışabilmektedir. Örneğin farları kullanıcı kendisi açabileceği gibi, araç çalışıp da şarj etmeye başlayınca farlar otomatik çalışabilmektedir. Ya da kullanıcı farları söndürebileceği gibi, araç motor durduktan belirli süre sonra akü enerjisini korumak için sistem farları otomatik olarak söndürmektedir.
- Hazırlanan Arduino kodu, işlemleri yaparken eğer PC'ye takılı ise yaptığı işlemleri PC'ye istenilen adımlarda USB kablosu üzerinden bildirmektedir.
- Aynı şekilde Arduino kartı tam bir aktif server olarak çalışıp, sistemin anlık olarak gözlemlediği araç donanımının durumunu gerçek zamanlı olarak ağdan yapılan sorgulara cevap olarak sunmaktadır.
- Arduino sistemine herhangi bir HTML tarayıcısı ile rahatlıkla bağlanılabilmektedir.
- Ayrıca sistem, hazırlanan Android programı ile de rahatlıkla çalışabilmektedir.

Hazırlanan bu Arduino kodunun denetleyici kartına aktarılması ve test edilmesi ilerleyen bölümlerde izah edilecektir.

2.3.4. Android Programlama

Android, şu anda Google firmasının geliştirilen, Linux çekirdeği üzerine kurulu bir cep telefonu işletim sistemidir. Doğrudan kullanıma müsaade eden bir ara yüze sahiptir. İlk zamanlar cep telefonu, tablet gibi taşınabilir cihazlar için hazırlanmış olan Android işletim sistemi şimdi TV'lerde, araçlarda, kol saatlerinde dahi kullanılmaktadır. Android gerçek hayatta olana çok benzer şekilde dokunmatik ekranda sürükleme, içe toplama, dışa sündürme, cihazı eğdirme gibi davranışları desteklemektedir. Açık kaynak kodlu bir sistemdir. Desteklenen uygulama uzantısı

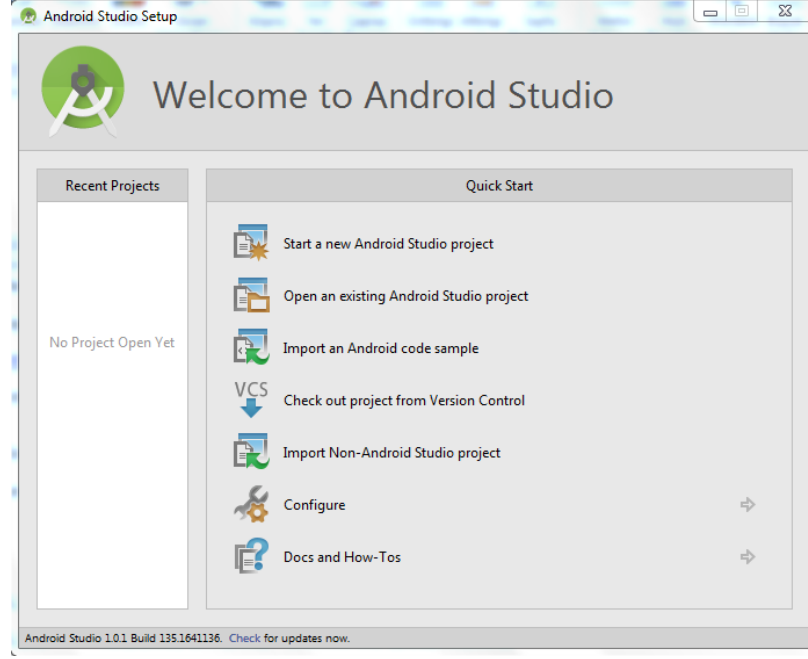
".apk"dır. Her geçen gün popülaritesi artan sistem için, an itibariyle 1 milyondan fazla hazırlanmış uygulama mevcuttur. Uygulamalar Google Play Store isimli bir uygulama veya <https://play.google.com/store> adresli İnternet sitesi aracılığı ile cihaza kurulabilmektedir.

Bizzat Google tarafından desteklenen Android Studio isimli ücretsiz IDE'si <http://developer.Android.com/sdk/index.html> adresinde ücretsiz sunulmaktadır. Bununla beraber Android için yazılım hazırlamaya uygun birçok ücretli ve ücretsiz platform mevcuttur.

Android Studio Java kullanılarak hazırlanmış, Java Programlama Diline çok benzer ve diğer Java IDE'leriyle hazırlanmış kodların rahatlıkla entegre edilebileceği bir yapıdadır.

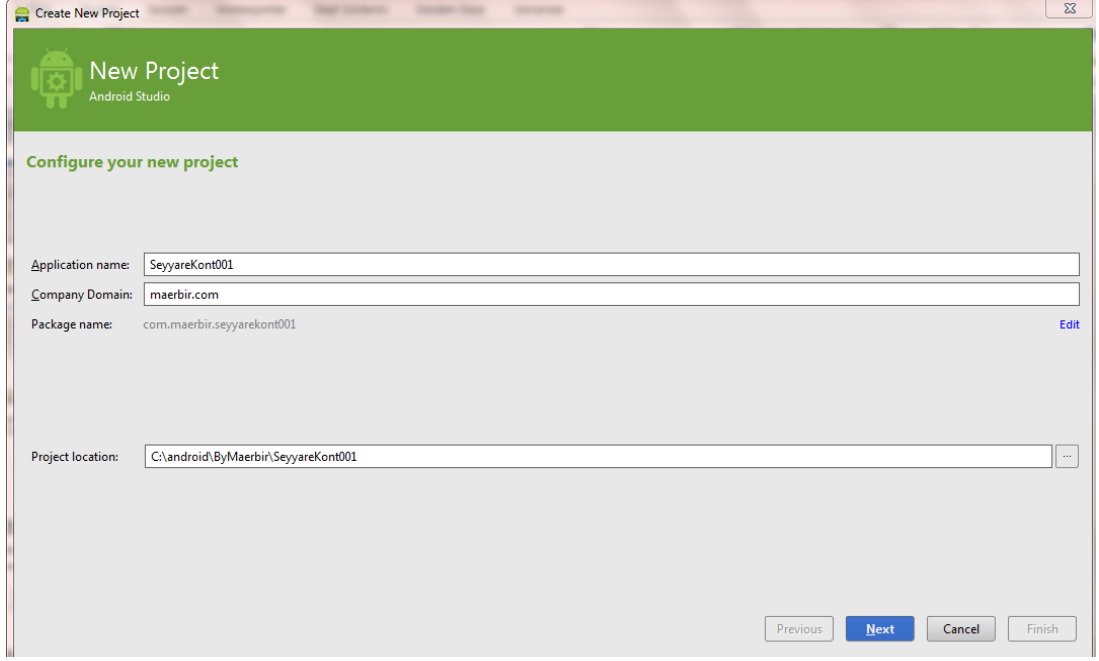
Android Studio yazılımı yukarıda adresi verilen siteden indirip bilgisayara kuruldu. Android Studio kurulurken bilgisayarda Java Development Kit bulamadığını ve kurulması gerektiği uyarısı ile karşılaşıldı. Uyarıda bildirilen linki tıklanarak, açılan <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html> adresli İnternet sayfasından bilgisayara uygun olan (64 bit Windows) kurulum dosyası indirilip kuruldu.

Sonra Android Studio Yazılımının kurulumuna kalınan yerden devam edildi. Program kuruldu ve çalıştırıldı. İlk açılış ekranı aşağıdadır:

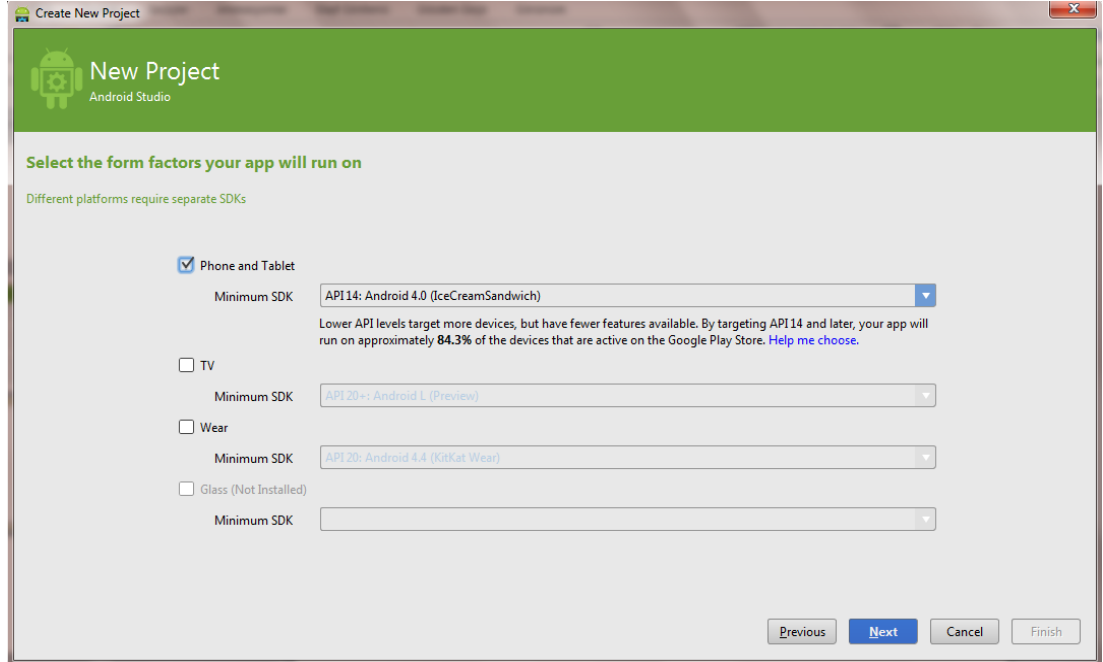


Şekil 2.13. Android Studio ilk açılış ekranı

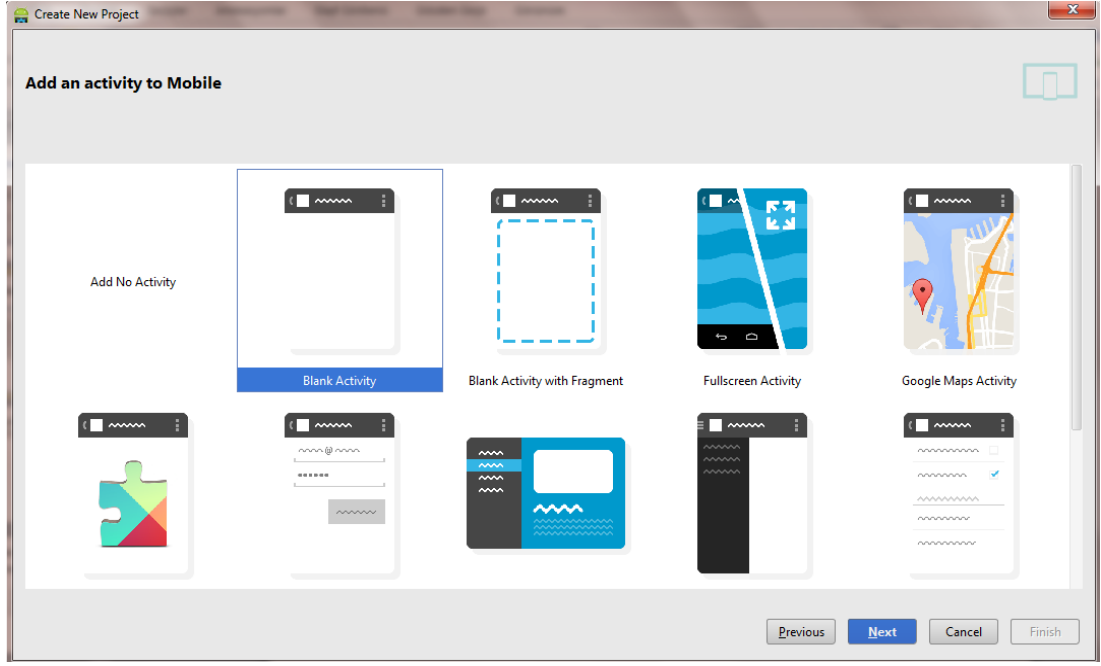
Bu ekrandan yeni bir Android projeye başlama seçildi. Sırasıyla açılan pencerelere gerekli bazı bilgiler girildi. İlk Android projesi “hello world” karşıda belirdi.



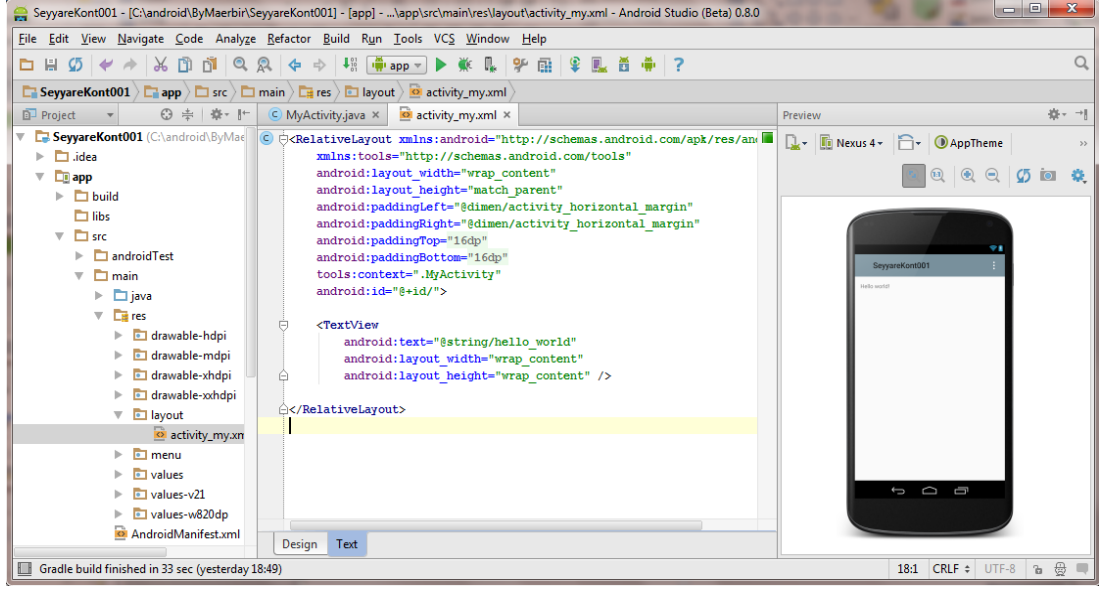
Şekil 2.14. Yeni projenin adının girileceği pencere



Şekil 2.15. Yeni projenin hangi Android sürümlerini destekleyeceğinin seçildiği pencere



Şekil 2.16. Yeni projenin aktivite türünün seçildiği pencere



Şekil 2.17. Varsayılan (default) yeni proje ekranı

Android Studio programında varsayılan olarak proje ilk açıldığında çalışmaya hazır bir Android uygulamasıdır. Derleyip emule edilirse ya da bir Android cihaza yüklenirse ekranda “Hello World” yazan bir mesaj çıkacaktır.

Şimdi sıradaki işlem, hazır çalışan Android uygulaması içine ihtiyacı giderecek gerekli kodların eklenmesi olacaktır.

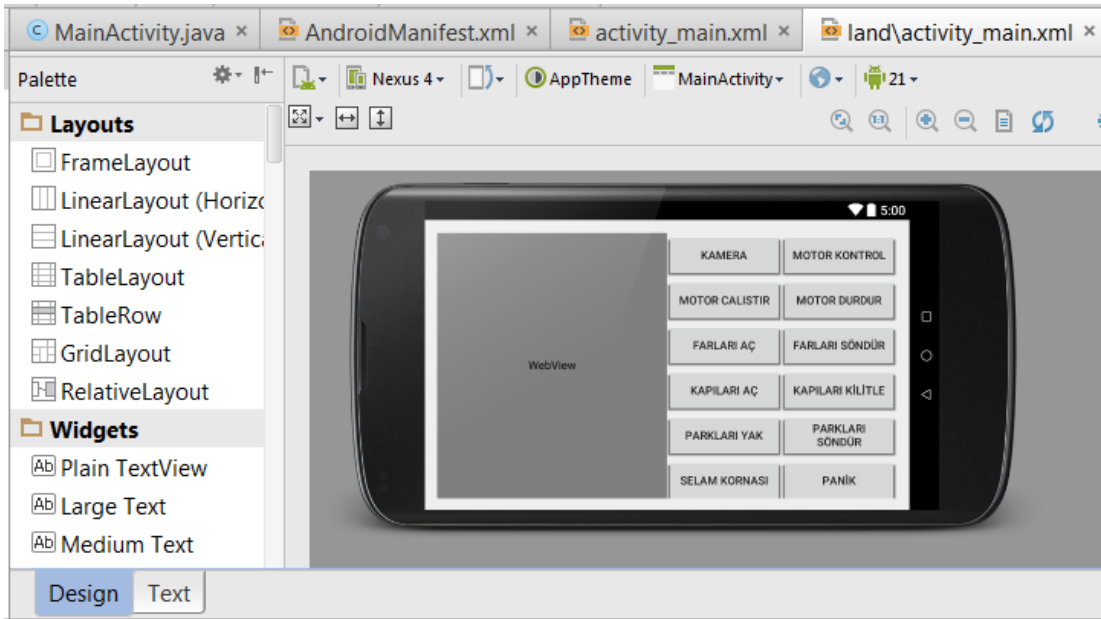
Android Studio, programlama yapılırken onlarca değişik dosyalar oluşturur. Bunlardan sadece gerekli olanlar değiştirilerek proje gelişir. Bu dosyalardan proje için ilgilenecekler şunlardır:

- MyActivity.java,
- activity_my.xml,
- Strings.xml
- AndroidManifest.xml

“MyActivity.java” isimli dosya programın bel kemiğidir, ana dosyasıdır. Tüm gerekli fonksiyonlar bu dosyaya yazılır ya da bu dosya içinden çağrılarak kullanılırlar.

“activity_my.xml” isimli dosya uygulamanın görsel tasarımını barındırır. Kullanıcıya görünen nesnelere yerleri ve isimleri bu dosyada mevcuttur. “String.xml” dosyası yazılımda kullanılacak ifadelerin topluca değiştirilmesini sağlayan dosyadır. Özellikle programın diller arası geçiş gibi amaçlar için pratiklik sağlayan bir dosyadır.

“AndroidManifest.xml” isimli dosya ise hazırlanacak uygulamanın Android cihazda hangi izinlere ihtiyaç duyacağını ve bunlar için gerekli bilgilerin kodlandığı dosyadır. Yukarıdaki 4 dosyanın içeriği ek olarak verilmiştir, bakılabilir. Ayrıca “activity_my.xml” isimli dosya görsel olarak da değiştirilebilmektedir. İstenilen tasarım görsel olarak “sürükle bırak” yöntemi ile hazırlandı, sonra nesnelere üzerine çift tıklanarak gerekli bilgiler girildi. Tasarım aşağıdadır:



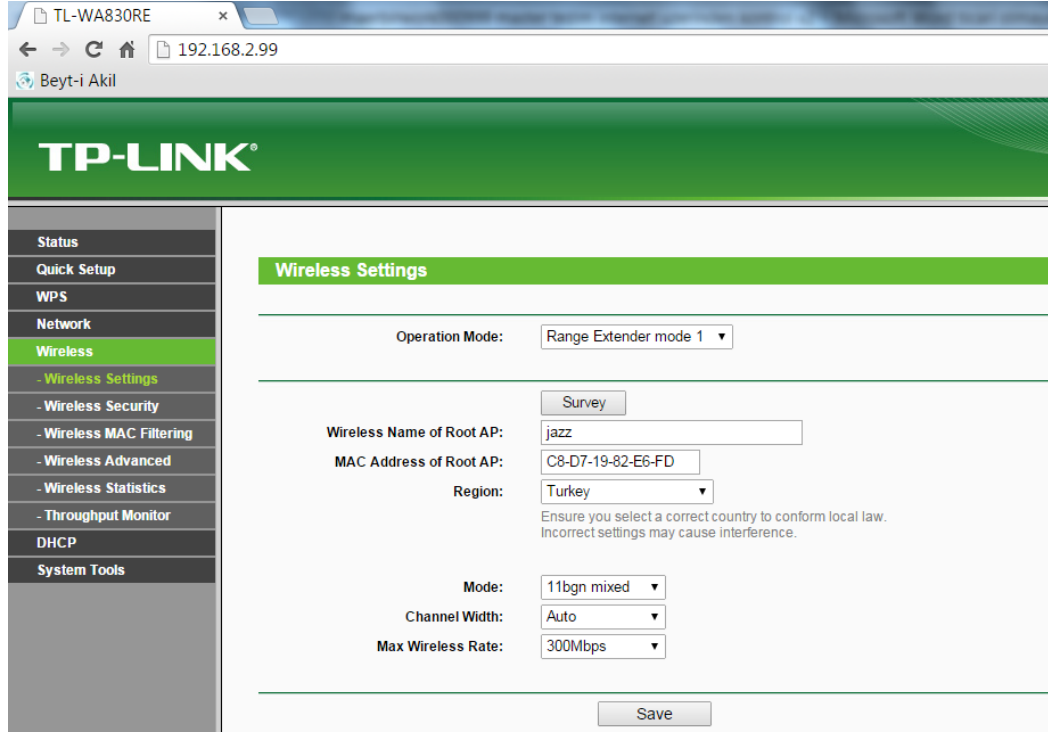
Şekil 2.18. Seyyare001 isimli Android uygulamasının görsel tasarımı

Hazırlanan bu Android kodunun Android bir cihaza aktarılması ve test edilmesi ilerleyen bölümlerde izah edilecektir.

2.3.5. Ağ Yapılandırılması

Repeater Cihazının Yapılandırılması

Projede imkân dâhilinde ise İnternete bağlanabilmek, değilse en azından sistemin kendi kablosuz ağını yaymak ve araç civarından araca Wi-Fi Ethernet ağı ile bağlanabilmek hedef idi. Bu amaçla bir adet kablosuz erişim noktası kullanıldı. Projede kullanılan Arduino denetleyici kartı da bir kablolu Ethernet modülü ile bu erişim noktasına, RJ45 soketli Ethernet kablosu kullanılarak, bağlandı. Ağ cihazı bir “repeater” olarak yapılandırıldı. Ta ki, eğer tanımlanan kablosuz İnternet ağına ulaşılabilirse hem bu Wi-Fi ağının mesafesini genişletsin, hem de denetleyici kartına İnternet bağlantısı sunsun. Ağ cihazı aşağıda görüldüğü şekilde yapılandırıldı:



The image shows a web browser window displaying the TP-Link wireless settings page for a device named TL-WA830RE. The browser address bar shows the IP address 192.168.2.99. The page has a green header with the TP-LINK logo. On the left side, there is a navigation menu with options: Status, Quick Setup, WPS, Network, Wireless (selected), - Wireless Settings, - Wireless Security, - Wireless MAC Filtering, - Wireless Advanced, - Wireless Statistics, - Throughput Monitor, DHCP, and System Tools. The main content area is titled "Wireless Settings" and contains the following configuration options:

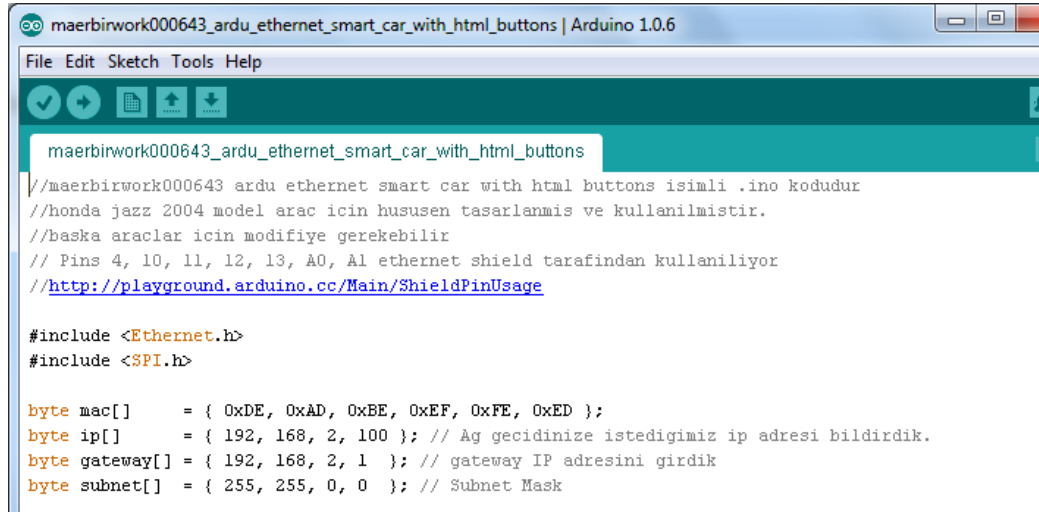
- Operation Mode: Range Extender mode 1
- Survey button
- Wireless Name of Root AP: jazz
- MAC Address of Root AP: C8-D7-19-82-E6-FD
- Region: Turkey
- Mode: 11bgn mixed
- Channel Width: Auto
- Max Wireless Rate: 300Mbps
- Save button

Below the Region dropdown, there is a note: "Ensure you select a correct country to conform local law. Incorrect settings may cause interference."

Şekil 2.19. “repeater” modunda kullanılan ağ cihazı yapılandırma ayarları

Arduino Ethernet Kartın Yapılandırılması

Arduino denetleyici kartına ilave edilen Ethernet modülünün çalışabilmesi için gerekli yapılandırma kodları Arduino koduna eklendi.



```
maerbirwork000643_ardu_ethernet_smart_car_with_html_buttons
//maerbirwork000643 ardu ethernet smart car with html buttons isimli .ino kodudur
//honda jazz 2004 model arac icin hususen tasarlanmis ve kullanilmistir.
//baska aracler icin modifiye gerekebilir
// Pins 4, 10, 11, 12, 13, A0, A1 ethernet shield tarafından kullaniliyor
//http://playground.arduino.cc/Main/ShieldPinUsage

#include <Ethernet.h>
#include <SPI.h>

byte mac[]      = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[]       = { 192, 168, 2, 100 }; // Ağ gecidinize istedigimiz ip adresi bildirdik.
byte gateway[] = { 192, 168, 2, 1 }; // gateway IP adresini girdik
byte subnet[]  = { 255, 255, 0, 0 }; // Subnet Mask
```

Şekil 2.20. Arduino kodunda IP yapılandırması

Yukarıdaki kod parçasında görüldüğü üzere, gömülü sisteme

- Ethernet protokolü için gerekli olan bir MAC adresi,
- Ağda IP protokol için gereken kendine has birer IP adresi, gateway adresi ve alt ağ maskesi tanımlandı.

Bu sayede yerel Wi-Fi ağına bağlı, herhangi bir İnternet tarayıcı ile IP adresi kullanılarak ya da Seyyare001 uygulaması ile direk gömülü sisteme ulaşıp kontrol etmek mümkün olacaktır.

IP Kameranın Yapılandırılması ve Bağlantısı

IP kameraların yapılandırılması ve kullanılması oldukça basittir. Herhangi bir Ethernet ağına doğrudan kablolu ya da opsiyonel olarak kablosuz bağlanabilirler. Kendilerine ait IP adresinden server olarak çalışırlar ve istek olduğu durumlarda görüntü ya da video sunabilirler.



Şekil 2.21. Bir IP kameranın ön ve arkadan görünüşü

IP kameraya, bağlı olduğu yerel ağa mensub bir cihazla dâhili IP'si ile bağlanılabilir. Mesela projedeki IP kameranın IP'si 192.168.2.90, port numarası ise 8080 olacaktır.

GSM - WiFi Modemin Yapılandırılması

Harici ağlardan da sisteme ulaşmak mümkün olacaktır. Bu işlem, ağı harice/İnternete bağlayan router/gateway cihazında gerekli IP yönlendirmesi (IP forwarding) ayarları ile sağlanabilir ve aşağıdaki resimde verildiği şekilde bu yapılandırma sağlanmıştır.



Şekil 2.22. Projede İnternet erişimi için kullandığım bir 3G – WiFi gateway cihazı

Hızlı Kurulum

Bağlantı Ayarları

Kablosuz Ağ

DHCP

Güvenlik

Firewall

LAN IP Filtresi

-> Sanal Sunucu

Özel Uygulamalar

DMZ Ayarları

SIP ALG Ayarları

UPnP Ayarları

NAT Ayarları

Sistem

Sanal Sunucu

Harici bilgisayarların WWW, FTP veya LAN tarafından sağlanan diğer hizmetlere erişmesini sağlamak için sanal bir sunucu yapılandırın.

- **IP adresi:** Hizmetleri sağlamak için LAN'da bulunan bir bilgisayarı atayın.
- **LAN/WAN bağlantı noktası:** Hizmetleri sağlayan bilgisayarın bağlantı noktası. Bu tek bir bağlantı noktasıdır ve LAN/WAN Bağlantı Noktasının değer aralığı 1-65535'dir.
- **Protokol:** Hizmetler tarafından uygulanan protokoller.
- **Not:** Ayarlar, "Uygula" düğmesi tıklanana kadar geçerli olmayacak.

Sanal Sunucular Listesi

Ad	WAN Bağlantı Noktası	LAN IP Adresi	LAN Bağlantı Noktası	Protokol	Durum	Seçenekler
smart_car	1000	192.168.2.100	80	TCP/UDP	Açık	Düzenle Sil
smart_office	1001	192.168.2.101	80	TCP/UDP	Açık	Düzenle Sil

[Ekle](#)

[Uygula](#)

Şekil 2.23. İnternette sistemle bağlanmak için router/gateway IP yönlendirme ayarları

Eğer IP kameraya da yerel ağ dışından (mesela İnternette) bağlanılmak istenirse onun IP'sine de yönlendirme yapılması gerekecektir.

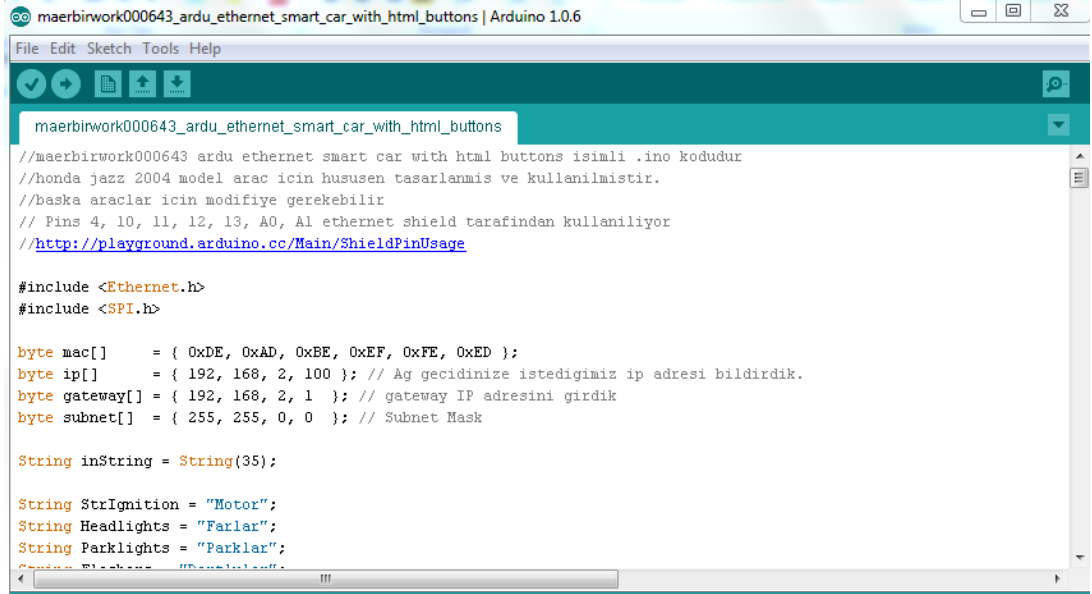
2.3.6. Hazırlanan Kodların Yüklenmesi

Şimdi bir önceki bölümde tasarlanan ve hazırlanan kodlar gömülü sisteme ve kontrol için kullanılacak uygulama Android cihaza yüklenecektir.

Arduino Kodunun Mikro Denetleyici Kartına Yüklenmesi

Mikro denetleyiciye kod hazırlamak için kullanılan Arduino isimli yazılım, aynı zamanda kodu Arduino denetleyici kartlarına aktarma görevini de yapabilmektedir. Bunun için yapılması gerekenleri aşağıda sıra ile uygulanacaktır.

- Arduino yazılımında gerekli kodu hazırlandı.



```
maerbirwork000643_ardu_ethernet_smart_car_with_html_buttons | Arduino 10.6
File Edit Sketch Tools Help
maerbirwork000643_ardu_ethernet_smart_car_with_html_buttons
//maerbirwork000643 ardu ethernet smart car with html buttons isimli .ino kodudur
//honda jazz 2004 model arac icin hususen tasarlanmis ve kullanilmistir.
//baska aracler icin modifiye gerekebilir
// Pins 4, 10, 11, 12, 13, A0, A1 ethernet shield tarafından kullaniliyor
//http://playground.arduino.cc/Main/ShieldPinUsage

#include <Ethernet.h>
#include <SPI.h>

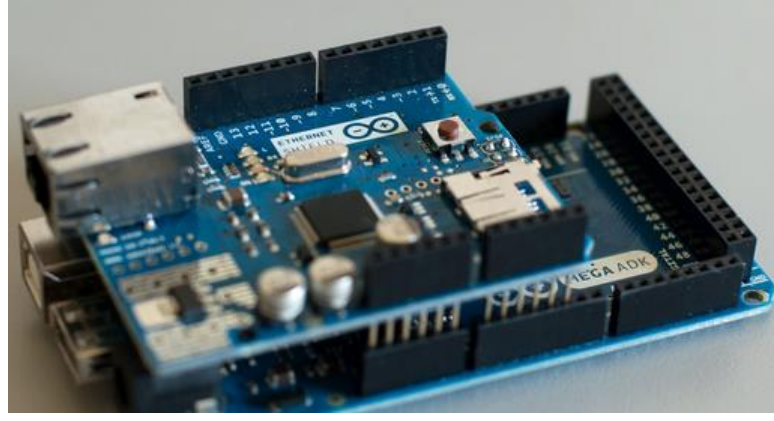
byte mac[]      = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[]       = { 192, 168, 2, 100 }; // Ağ gecidinize istedigimiz ip adresi bildirdik.
byte gateway[] = { 192, 168, 2, 1 }; // gateway IP adresini girdik
byte subnet[]  = { 255, 255, 0, 0 }; // Subnet Mask

String inString = String(35);

String StrIgnition = "Motor";
String Headlights = "Farlar";
String Parklights = "Parklar";
String StrEngine = "Motorlu";
```

Şekil 2.24. Projede kullanılan Arduino Kodundan bir ekran görüntüsü

- Arduino kartını temin edildi.



Şekil 2.25. Ethernet modülü eklenmiş Arduino Mega Kartı

- Arduino kartı bir USB kablosu ile bilgisayara bağlandı.
- Arduino Yazılımından uygun kart ayarları seçildi.
- Yükle butonuna tıklanarak ilgili makine kodunu oluşturup Arduino kartına yüklendi.

Gömülü sisteme kod yükleme işi böylelikle tamamlandı.

Android Kodunun Android Cihaza Yüklenmesi

Android cihaza kod hazırlamak için kullanılan Android Studio isimli yazılım aynı zamanda, hazırlanan kodu Android CİHAZLARA aktarma görevini de yapabilmektedir. Bunun için yapılması gerekenler aşağıdaki sıra ile uygulandı.

- Android cihaz için gerekli kod hazırlandı.
- Android cihaz bir USB kablosu ile bilgisayara bağlandı. Bu işlem birkaç farklı Android cihazla ayrı ayrı denendi. Bazılarını Android Studio hemen tanıdı. Ancak Samsung marka olan Android cihazlar ancak firmanın ücretsiz sunmuş olduğu KIES isimli bilgisayar yazılımı kurulduktan sonra Android Studio tarafından tanınabilmektedir.
- Android Studio Yazılımında kodu çalıştır butonuna tıklandı. Kodun nerede çalıştırılarak denemek istendiğini soran ekranda bilgisayara bağlanan Android cihaz seçildi.

- Hazırlanan Android uygulaması Android cihaza yüklendi ve hemen çalışmaya başladı.

2.4. Tüm Donanımın Bir Araya Getirilip Sistemin Çalıştırılması

Son olarak tüm donanımın uygun şekilde birbirine kablolu ya da kablosuz şekilde bağlanmasına gelindi.

2.4.1. Tasarlanan Gömülü Sistemin Otomobile Bağlanması

Kontrol etmek düşünülen tüm donanımlara kablolarla ulaşım bilgi alış verişini sağlandı. Motoru çalıştırmak ve durdurmak için kontak sistemine, araç ön ve arka aydınlatmaları için butonlardan rölelere giden kablolarla, dörtlüler için dörtlü buton düğmesine bağlı kablolarla vs. bağlantılar yapıldı. Aşağıda detaylandırılacaktır.

Sistemde kullanılan tüm elektronik devreler $12V_{dc}$ ile çalıştığından herhangi bir adaptör kullanmadan doğrudan aracın aküsünden beslenebilmektedir.

Ancak sistemin birçok parçası kendi içinde $12V$ 'u $5V$ 'a düşürüp çalışmaktadır. Mesela Arduino kartı rölelere kontrol sinyali $5V$ olarak göndermektedir. Ancak rölelerin çalışması için gerekli olan beslemesi de yine $12V$ 'tur.

Kapıları Kilitlemek ve Kilitleri Açmak için Gerekli Bağlantılar

Kapıların kilitlerini açmak ya da kapıları kilitlemek için araç içindeki ilgili buton kullanıldı. Butonun 3 konumu bulunmaktadır. Buton normalde pasif olan orta konumdadır. Butona 3 hat bağlı; birisi sürekli $12V$, diğer ikisi ise normalde $0V$, birisine $12V$ bağlanırlarsa kapılar kilitlenmektedir. Diğerine $12V$ bağlanırsa kapı kilitleri açılmaktadır. Mevzubahis iki kablodan gömülü sisteme bağlantı yapılarak istenilen durumlarda istenilen kabloya $12V$ gönderilebilecek şekilde bağlantı yapıldı.

Gömülü sistem gerekli durumlarda röleler aracılığı ile istenen kabloya 12V darbesi gönderiyor duruma getirildi.

Motor Çalışmasını Başlatmak ve Durdurmak için Gerekli Bağlantılar

Motoru çalıştırmak ve durdurmak için aracın kontak mekanizması kullanıldı. Kontakın 4 konumu bulunmaktadır. “O” kapalı konumu, “I” bazı donanıma enerji verildiği durum, “II” tüm donanımın enerjilendirildiği durum ve “III” marşa basma konumu. Bu konumlara ait bağlantılar, piyasada 60 TL fiyata temin edilebilen oto-marş modülüne bağlandı. Oto-marş modülü, kontrol sinyali ile otomatik olarak motoru çalıştıran ya da durdurabilen bir modüldür. Bu modül gömülü sistem tarafından kontrol edilmektedir. Bu modülün kontrol girişi normalde 0V’tur ve 12V darbe ile tetiklenmektedir. Aracın motorunu çalıştırması için çift pozitif darbe, durdurması için tek pozitif darbe gereklidir.

Ön ve Arka Park Lambaları ve Farlar için Gerekli Bağlantılar

Park lambaları ve farların aktive edilmesi için direksiyonun solundaki kumanda kollarına bağlı kablolar kullanıldı. Normalde 12V bulunan bu kablolar, gömülü sistem gerekli durumlarda röle aracılığı ile 0V göndererek park lambalarını ve farları aktive edebilmektedir. Röleleri deaktive ederek 0V bağlantısı kesildiğinde ise aracın devreleri, kumanda koluna bağlı sinyal kablosunu tekrar 12V’a yükselterek, park lambalarını ve farları kapamaktadır.

Dörtlü Sinyalleri Kontrol için Gerekli Bağlantılar

Dörtlü sinyallerin aktive edilmesi için kumanda butonu kullanıldı. Dörtlü kontrol butonuna ikişer adet kablonun girip çıktığı ve bunların sağ ve sol sinyalleri ayrı ayrı kontrol edebileceği tespit edildi. Buton yerine röleler kullanarak bu kabloları kısa devre yaptırarak dörtlü sinyalleri aktive edilebilmektedir.

Araç Motorunun Çalışıp Çalışmadığını Anlamak için Gerekli Bağlantılar

Aracın motorunun çalışıp çalışmadığının anlaşılması için aracın gösterge panelindeki şarj lambasını kullanıldı. Aracın şarj lambasının çalışma prensibi şöyledir; kontak “II” konumunda iken lambanın bir kutbuna aküden pozitif voltaj uygulanmakta ve lamba ışık saçmaktadır. Lambanın diğer kutbu ise şarj dinamosuna bağlıdır. Aracın çalışmaya başlamasıyla şarj dinamosu voltaj üretmeye başlayınca, mevzubahis lambanın her iki kutbu aynı voltaja sahip olduğundan şarj lambası sönmektedir. Bu lambanın şarj dinamosuna bağlı kutbuna ulaşıldı. Bu noktada bir kablo ile gömülü sisteme bağlandı. Bu kabloda pozitif voltaj (14V civarında) var olup olmadığı, dolayısıyla araç motorunun çalışıp çalışmadığı gözlemlendi.

Korna için Gerekli Bağlantılar

Kornanın aktive edilmesi için direksiyonun orta noktasına bağlı kablolar kullanıldı. Normalde 12V bulunan bu kabloları, gömülü sistem gerekli durumlarda röle aracılığı ile 0V'a bağlayarak kornayı aktive edebilmektedir. Röleleri deaktive ederek 0V bağlantısı kesildiğinde ise aracın devreleri, bu sinyal kablosunu tekrar 12V'a yükseltmektedir.

2.4.2. Gömülü Sistemin Ağ Cihazına Bağlantısı

Gömülü sistem zaten RJ-45 soketli bir Ethernet modulüne sahip olduğundan, erişim noktası (Access point) ve tekrarlayıcı (repeater) özelliklerine sahip, piyasada mevcut olan en düşük özellikli ve fiyatlı kablosuz ağ cihazı ile dahi sistem tam verimle çalışabilmektedir.

2.4.3. Sisteme Enerji Verilerek Sistemin Çalıştırılması

Tüm tasarımı, kodlamaları ve bağlantıları tamamlanan sisteme enerji verilerek denendi. Sistem başarı olarak çalıştığı gözlemlendi.

2.5. Projenin, Uzaktan Kontrolde Uygulanabilirliđi

Projede;

- Kullanılan alet ve elemanların uygun fiyatlı olması,
- Piyasadan rahatlıkla temin edilebilirlikleri,
- Kullanılan yazılımların tamamının tam sürümlerinin ücretsiz olması,
- Donanım kontrolü için röle kullanılmış olması,
- Haberleşme alt yapısı olarak profesyonel özelliklere sahip olan Ethernet ve İnternet teknolojilerinin kullanılmış olması,

Sebeplerinden ötürü, tasarlanan ve gerçekleştirilen bu sistem, bilumum uzaktan kontrol ve kumanda gereksinimlerini karşılayabilecek özellikleri taşımaktadır.

2.6. Projenin Maliyeti ve Mevcut Alternatifleri ile Mukayesesi

Projenin maliyeti ve mevcut alternatifleri ile kıyası yapılarak sistemin avantajları gösterilecektir.

2.6.1. Projenin Maliyeti

Projede kullanılan elemanların maliyetleri 30.01.2015 günü itibariyle çizelge 2.1'de sunulmuştur. Deđerler Amerikan Doları cinsindedir.

Çizelge 2.1. Proje maliyet tablosu

Malzeme Tanımı	Adet fiyatı	Miktarı	Toplam Tutarı
Arduino Mega	45.55	1	45.55
Arduino Ethernet modülü	16.56	1	16.56
Breadboard	4.14	5	20.70
Wi-Fi Erişim Noktası Cihazı	24.84	1	24.84
Jumper Kablolar, erkek erkek (1 top)	10.35	1	10.35
Jumper Kablolar, dişi-dişi (40 li)	9.52	1	9.52
Devre kutusu (mano 24'' organizer)	7.87	1	7.87
Devre kutusu (torpido gözüne ara bağlantı için)	5.80	1	5.80
Oto-marş modülü	24.84	1	24.84
8li röle seti	13.25	1	13.25
Çok hatlı kablo 3m (bagajdan aracın motor kısmına çekilen)	4.14	3	12.42
Kontrol ve kumanda için herhangi bir İnternete bağlanabilen cihaz; telefon, tablet, bilgisayar vs.		-	
(İsteğe bağlı) 3G – Wi-Fi Gateway	21.95	1	21.95
Genel Toplam (\$)		1	210.35

2.6.2. Projenin Mevcut Alternatifleriyle Karşılaştırılması

Başarı ile sonuçlandırılan bu proje farklı açılardan alternatifleri ile mukayese edilecektir.

Mevcut Alternatiflerden Bazıları

Şu an için bu projeye alternatif olarak birkaç dev otomobil firmalarının ürünleri ve bazı özel firmaların uygulamaları mevcuttur. Mesela AUDI firması sadece kendi araçlarına has olarak benzer bir sistem tasarlamıştır. Henüz satışa çıkmamış olan bu çalışmanın, aşağıdaki linkte demo videosuna erişilebilir: <http://www.youtube.com/watch?v=IYILkueSxyk>

Benzer şekilde MERCEDES firmasının da bir otomatik ilerleyebilen araç üzerinde çalışmaları vardır, demosuna şu linkten ulaşılabilir: <http://www.youtube.com/watch?v=LHqB47F12vI>

Özellikler Açısından Karşılaştırma

Yukarıda bahsedilen otomobil firmalarının, üzerinde çalışmakta oldukları sistemler, sadece kendi firmalarına aittir. Bu tez sürecinde gerçekleştirilen proje ise tüm motorlu araçlara uygulanabilir bir niteliktedir.

Şu an için gerçekleştirilen bu projede aracın tüm donanımlarının uzaktan kontrol ve kumanda edilebilmesi mümkündür. Rakip firmalarda ise fazladan otomatik sürüş özelliği mevcut olmakla beraber, uzaktan kumanda ve kontrol özelliği ise kısmen mevcuttur.

Fiyat Açısından Karşılaştırma

Tez projesinin prototip parça maliyeti 200 \$ (500 TL) civarında iken, mevzubahis firmaların fiyatları ise henüz belli değildir. Ancak, zaten bu firmaların düşük seviye araçları dahi Türkiye’de ve dünyadaki ekser insanların finanse edemeyeceği düzeydedir. Hal böyle iken otomatik veya uzaktan kontrolle ilerleyebilme özelliğine sahip modellerinin fiyatları ile tez projesinin kıyas edilmesi gülünç olacaktır.

Geliştirilebilirlik Açısından Karşılaştırma

Sistemin tüm teknik bilgi, belge ve kodları tarafımızdan hazırlanmış olduğundan, gömülü sistemin tüm donanım ve yazılımı düzeltme ve geliştirilmeye açık durumdadır. Bu da, sistemi geliştirme ve genişletme hususunda hayal gücü ile sınırlanan bir özgürlük sunmaktadır.

Şu an için sistemin tüm elektrikli donanımı uzaktan bir insan tarafından kontrol edilebildiği gibi otomatik olarak da kontrol edebilir alt yapıya sahiptir. İnternet ve Ethernet üzerinden kontrol edilebildiği gibi, rahatlıkla IR kumanda ya da bir RF kumanda ile de tüm donanım kontrol edebilir.

Bu derece kişiselleştirme ve genişletme imkânı, şu an piyasadaki hiçbir alternatifinde mevcut değildir.

3. BULGULAR

Gerçekleştirilen bu tez projesi İnternet/Ethernet üzerinden motorlu taşıtlar ve her türlü elektrikli sistemi rahatlıkla kontrol ve kumanda edebilir özelliklerde olması ile uzaktaki aracı ve sistemlerin durumunu istenilen zamanda ve İnternet bağlantısı olan her mekândan gözlenebilir ve kontrol edilebilir hale getirmesiyle güvenlik hususunda fevkalade kullanışlıdır. Mesela kontrol edilen aracın çalışıp çalışmadığı, kapılarının kilitli mi, açık mı olduğu, farlarının çalışır konumunda unutulup unutulmadığı gibi birçok bilgi şu an için uzaktan gözlemlenebilmektedir.

Araç istenilen zamanda ve mesafeden çalıştırabilir, durdurabilir, kapı kilit konumları değiştirebilir olması gibi sebeplerle, aracı (veya herhangi bir elektrikli sistemi) mesela kullanmadan 15 dk önceden çalıştırarak, kışın araca binmeden ısınması, yazın ise önceden çalışarak serinlemesi sağlanabilmektedir.

Hatta belirli aralıklarla (mesela her gün saat 08:00'de) aracın otomatik olarak çalışarak düzenli araca binme saati olan (mesela 08:45) vakitte kışın ısınmış, yazın serinlemiş hale gelmesi sağlanabilir.

Ya da tek bir tuşla hem aracın motorunu, hem farları çalışır konuma, hem de kapı kilitleri açık konuma getirilebilir. Sistem bu gibi tamamen hayal gücü ile sınırlı konforu sunabilir.

4. SONUÇ

4.1. Sonular

Tez sresince Arduino denetleyici tabanlı Ethernet protokol kullanan HTML ve Android ara yzl, yksek kapasiteli bir uzaktan kumanda ve gzlem, diđer deęişle kontrol sistemi tasarlandı. Bu sistem Honda marka Jazz model 2004 yılında retilmiř bir otomobilde test edildi ve %100 bařarılı sonular elde edildi. Bylelikle rakiplerine gre ok daha esnek, ekonomik, her ara modeline uyumlu, Hem Trk Milleti, hem tm insanlık iin ok aıdan kullanılabilir bir sistem gerekleřtirildi.

4.2. neriler

řimdilik sadece otomobilin elektrikselsel noktaları kullanılarak sistem gerekleřtirildi. Eđer aracın direksiyon, gaz ve fren pedalları, vites kolu gibi mekanik aksamı da yardımcı bazı donanım ilave edilirse, hem tam otonom ve hem de İnternette tam kontrol edilebilir bir prototip gerekleřtirilebilir.

Ayrıca sisteme bir GPS modl eklenip aracın anlık konumu tespit edilebilir. Aracın konumu ve konuma ait gncel trafik ve hava durumu analiz edilerek, ara kullanıcısına (gerek otonom denetleyici birimlerine gerekse řofre) srř kalite ve konforunu artırıcı uyarılar yapılabilir. Hatta aracın bazı donanımı otonom olarak da kontrol edilebilir. rneęin hava kořulları, trafięin yoęunluęu, virajların keskinlięi gibi parametreler kullanılarak gvenli srř hızı tespit edilebilir, alternatif yollar tavsiye edilebilir ya da ndeki aracı takip mesafesi deęiřtirilebilir.

řofr koltuęuna yerleřtirilecek bir sensrle, aracın řofr koltuęu boř iken otomatik olarak fren yapması saęlanabilir.

Ara rotası nceden belirlenerek, aracın dneceęi křelere yaklařınca otomatik olarak yavařlaması ve (veya) sinyal vermesi saęlanabilir.

KAYNAKLAR

- [1] Anonim, “İnternet üzerinden araç kontrolü konulu patentler”, https://www.google.com.tr/?tbs=pts&gws_rd=cr,ssl&ei=R7_fVLeRD8q7UZqig4gB#tbs=pts&q=vehicle+control+over+İnternet (erişim tarihi 14 Şubat 2015)
- [2] Anonim, Mart 2014, “Remote Control: Material Handling at a Distance – Endüstri Haberleri”, http://news.thomasnet.com/imt/2001/06/01/remote_control (erişim tarihi 7 şubat 2015)
- [3] Anonim, “Kablosuz İletişim – Wikipedia, Ücretsiz Ansiklopedi”, http://tr.wikipedia.org/wiki/Kablosuz_ileti%C5%9Fim (erişim tarihi 6 Şubat 2015)
- [4] Anonim, “Radio Frequency – Wikipedia, Ücretsiz Ansiklopedi”, http://en.wikipedia.org/wiki/Radio_frequency (erişim tarihi 6 Şubat 2015)
- [5] Anonim, “IR Communication – Sparkfun Tutorials”, <https://learn.sparkfun.com/tutorials/ir-communication> (erişim tarihi 6 Şubat 2015)
- [7] Anonim, “A Look at the Basics of Bluetooth Technology – Web Sitesi”, <http://www.bluetooth.com/Pages/Basics.aspx> (erişim tarihi 6 Şubat 2015)
- [6] Anonim, “Bluetooth – Wikipedia, Ücretsiz Ansiklopedi”, <http://en.wikipedia.org/wiki/Bluetooth> (erişim tarihi 6 Şubat 2015)
- [8] Anonim, “Public Switched Telephone Network – Wikipedia, Ücretsiz Ansiklopedi”, http://en.wikipedia.org/wiki/Public_switched_telephone_network (erişim tarihi 6 Şubat 2015)
- [9] Anonim, “Mobile Phone – Wikipedia, Ücretsiz Ansiklopedi”, http://en.wikipedia.org/wiki/Mobile_phone (erişim tarihi 6 Şubat 2015)

- [10] Anonim, “IP Protocol – Wikipedia, Ücretsiz Ansiklopedi”, http://en.wikipedia.org/wiki/Internet_Protocol (erişim tarihi 6 Şubat 2015)
- [11] Anonim, “Ethernet – Wikipedia, Ücretsiz Ansiklopedi”, <http://en.wikipedia.org/wiki/Ethernet> (erişim tarihi 6 Şubat 2015)
- [12] Anonim, “Internet – Wikipedia, Ücretsiz Ansiklopedi”, <http://en.wikipedia.org/wiki/Internet> (erişim tarihi 6 Şubat 2015)
- [13] Anonim, “Wireless LAN – Wikipedia, Ücretsiz Ansiklopedi”, http://en.wikipedia.org/wiki/Wireless_LAN (erişim tarihi 6 Şubat 2015)
- [14] Barret, 2010, Embedded System Design with the Atmel AVR Microcontroller, 1. Bölüm. Morgan & Claypool Publishers,
- [15] Anonim, “What is Arduino?”, <http://arduino.cc/en/Guide/Introduction> (erişim tarihi 15 Şubat 2014)
- [16] Gerard Goggin, 2012, “Driving the Internet: Mobile Internets, Cars, and the Socials”, Future Internet Open Access Magazine, ISSN 1999-5903, MDPI - Open Access Publishing
- [17] Bavyera Motor Fabrikaları, “BMW Avustralya’da araçtan İnternete Bağlan”, Online ulaşılabilir: <http://www.bmw.com.au/com/en/insights/newsandevents/latest-news/in-car-Internet-access.html> (erişim tarihi 15 Şubat 2015)
- [18] Sheller, M.; Urry, J. The city and the car. Internatioanl Journal of Urban Regional Research, 2000, Volume 24, 737–757.
- [19] Leung, L.; Fung, A.; Lee, P., Eds. Embedding into Our Lives: New Opportunities and Challenges of the Internet; Chinese University Baskısı: Hong Kong, Çin, 2009.

- [20] Donner, J.; Gitau, S.; Marsden, G. Exploring mobile-only İnternet use: results of a training study in urban South Africa, *Int. J. Commun.* 2011, 5, 574–597.
- [21] Flores-Roux, E.; Mariscal Avilés, J. Oportunidades y Desafíos de la Ancha Móvil. Available online: http://www.acorn-redecom.org/papers/2011Flores-Roux_Espanol.pdf (erişim tarihi 28 Şubat 2015)
- [22] Anonim, “Küresel Konumlama Sistemi - Wikipedia, Ücretsiz Ansiklopedi”, <http://tr.wikipedia.org/wiki/GPS> (erişim tarihi 28 Şubat 2015)
- [23] Anonim, “Mobiett - Akıllı Durak Yazılımı – Google Play Store”, <https://play.google.com/store/apps/developer?id=%C4%B0ETT+Genel+M%C3%BCd%C3%BCr%C3%BC%C4%9F%C3%BC> (erişim tarihi 28 Şubat 2015)
- [24] Z. Zhu, L. Zhang, “Supporting Mobility for İnternet Cars.” *IEEE Communications Magazine*, 0163-6804/11, Mayıs 2011
- [25] Anonim, <http://www.lexus.com/enform/mobile/index.html> (erişim tarihi 01 Mar 2015)
- [26] A. Jameel, M. Stuempfle, D. Jiang, A. Fuchs. “Web on Wheels: Toward İnternet-Enabled Cars”. *IEEE Magazine*, 0018-9162/98, Ocak 1998

EK 1. LED YAK-SÖNDÜR ARDUINO KODU

/*

Blink

Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the Uno and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at <http://arduino.cc>

This example code is in the public domain.

modified 8 May 2014

by Scott Fitzgerald

*/

```
// the setup function runs once when you press reset or power the board
```

```
void setup() {
```

```
  // initialize digital pin 13 as an output.
```

```
  pinMode(13, OUTPUT);
```

```
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {
```

```
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
```

```
  delay(1000);           // wait for a second
```

```
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
```

```
  delay(1000);           // wait for a second
```

```
}
```

EK 2. ARDUINO ETHERNET WEBSERVER KODU

/*

Web Server

A simple web server that shows the value of the analog input pins.
using an Arduino Wiznet Ethernet shield.

Circuit:

* Ethernet shield attached to pins 10, 11, 12, 13

* Analog inputs attached to pins A0 through A5 (optional)

created 18 Dec 2009

by David A. Mellis

modified 9 Apr 2012

by Tom Igoe

*/

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
// Enter a MAC address and IP address for your controller below.
```

```
// The IP address will be dependent on your local network:
```

```
byte mac[] = {
```

```
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

```
IPAddress ip(192,168,1,177);
```

```
// Initialize the Ethernet server library
```

```
// with the IP address and port you want to use
```

```
// (port 80 is default for HTTP):
```

```
EthernetServer server(80);
```

```
void setup() {
```

```
  // Open serial communications and wait for port to open:
```

```

Serial.begin(9600);
while (!Serial) {
  ; // wait for serial port to connect. Needed for Leonardo only
}
// start the Ethernet connection and the server:
Ethernet.begin(mac, ip);
server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());
}

void loop() {
  // listen for incoming clients
  EthernetClient client = server.available();
  if (client) {
    Serial.println("new client");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        // if you've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
        if (c == '\n' && currentLineIsBlank) {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close"); // the connection will be closed after
          completion of the response
          client.println("Refresh: 5"); // refresh the page automatically every 5 sec
          client.println();
        }
      }
    }
  }
}

```

```

client.println("<!DOCTYPE HTML>");
client.println("<html>");
// output the value of each analog input pin
for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
  int sensorReading = analogRead(analogChannel);
  client.print("analog input ");
  client.print(analogChannel);
  client.print(" is ");
  client.print(sensorReading);
  client.println("<br />");
}
client.println("</html>");
break;
}
if (c == '\n') {
  // you're starting a new line
  currentLineIsBlank = true;
}
else if (c != '\r') {
  // you've gotten a character on the current line
  currentLineIsBlank = false;
}
}
}
// give the web browser time to receive the data
delay(1);
// close the connection:
client.stop();
Serial.println("client disconnected"); } }

```


EK 3. BU TEZ ÇALIŞMASINDA ARDUINO HONDA JAZZ İÇİN ÖZEL OLARAK TASARLANMIŞ İNTERNET ÜZERİNDEN ARAÇ KONTROL KODU

```
//maerbirwork000643 ardu ethernet smart car with HTML buttons isimli .ino
kodudur

//honda jazz 2004 model arac icin hususen tasarlanmis ve kullanilmistir.
//baska araclar icin modifiye gerekebilir
// Pins 4, 10, 11, 12, 13, A0, A1 ethernet shield tarafından kullaniliyor
//http://playground.arduino.cc/Main/ShieldPinUsage

#include <Ethernet.h>
#include <SPI.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 1, 100 }; // Ag gecidinize istedigimiz ip adresi bildirdik.
byte gateway[] = { 192, 168, 1, 1 }; // gateway IP adresini girdik
byte subnet[] = { 255, 255, 255, 0 }; // Subnet Mask

String inString = String(35);

String StrIgnition = "Motor_:";
String Headlights = "Farlar_:";
String Parklights = "Parklar_:";
String Flashers = "Dortluler_:";
String Outlets12V = "Cakmakliklar_:";
String DoorLocks = "Kapilar_:";
String StrHorn = "Korna_:";
String StrHornGreet = "Dit_Dit_:";
String StrPanik = "PANIK_:";

unsigned long previousMillis=0; // millis() fonksiyonunu kullanarak sainyede bir
defa islem yaptirmak sayaclari guncellemek icin kullanacagim
```

int interval_seconds = 1000; //1000 ifadesi 1000 milisaniyeye yani 1 saniyeye denk gelmek icindir.

```
#define value_set_timer_lights 60 //farlar motor durduktan bu kadar sn sonra sonecek, 60 sn 1 dk yapar,
int timer_lights_off_latey = value_set_timer_lights; //farlar motor durduktan bu kadar sn sonra sonecek
#define value_set_timer_12Vsockets 60
int timer_12Vsockets_off_latey = value_set_timer_12Vsockets; //arac ici 12V soketlerin bu kadar sn sonra enerjisi kesilecek
#define value_set_timer_wifi 3600
#define value_set_timer_wifi_oneday 86400 //aracin wifi agi bu kadar zaman sonra otomatik kapanacak 1 gun = 24 x 60 x 60 = 86400 sn
int timer_wifi_off_latey = value_set_timer_wifi; //sistem resetlediginde aracin kendi yaydigi wifi bu kadar sn sonra kapanacak
```

```
int relay[] = {00, 22, 23, 24 ,25 ,26 ,27 , 28, 29 }; // LED Pinleri 0 Kullanılmıyor
#define po_headlights 22 //0V'a baglaninca on farlar yanar, acik devre birakilince söner
#define po_parklights 23 //0V'a baglaninca park lambalari yanar, acik devre birakilince söner
#define po_flashers 24 //0V'a baglaninca dortluler yanar, acik devre birakilince söner
#define po_12Voutlets 25 //dijital 1 yani 5V verilince role cekeer ve cakmakliklara 12V verilir.
#define po_horn 26 //0V'a baglaninca korna oter, acik devre birakilince susar
#define po_doorUnlock 27 //12V'a baglaninca kapılar kilitlenir
#define po_doorLock 28 //12V'a baglaninca kapılar kilitleri acilir
#define po_ignition 29 //motor durgunken 2 kere pespese 12V tetiklenirse otomars modulu araci calistirir, motor otomars modulu ile calisirken 1 kez tetiklenirse otomars modulu aracin motorunu durdurur.
#define po_wifi 30 //dijital 1 verilince access oint sistemine aktive edilir
```

```
#define pi_charging 40 //sarj dinamosundan gelen voltaji, voltaj bolucu ile bu pine
baglandi. arac sarj ederken 5V civarinda yani dijital 1 degilse 1.5V civarinda yani
dijital 0.
#define pi_mars_talebi 41 //dijital 1 gelirse baska unitelerden mars istegi var kabul
edilir
#define pi_door_unlock 42 //dijital 1 gelirse baska unitelerden kapi kilit ac istegi var
kabul edilir
#define pi_door_lock 43 //dijital 1 gelirse baska unitelerden kapilari kilitle istegi var
kabul edilir
```

```
boolean state_headlights = 0; //farlar yanarken "1", sonukken "0"
boolean state_parklights = 0; //parklar yanarken "1" sonukken "0"
boolean state_flashers = 0; //dortluler yanarken "1", sonukken "0"
boolean state_12Voutlets = 0; //cakmakliklara enerji verilirken "1" verilmezken "0"
boolean state_doorlocks = 0; // kapilar en son kilitlendi ise "1" kiltler acildi ise "0"
boolean state_charging = 0; //arac sarj edidyorsa "1", etmiyorsa "0"
boolean state_horn = 0; // korna oterken "1" otmezken"0"
boolean state_wifi = 1;
```

```
String txt_btn_headlights = "yak"; //varsayilan olarak far dugmesinde "yak" yaziyir,
farlar yaninca "sondur" yazacak
String txt_btn_parklights = "yak"; //varsayilan olarak park lambalari dugmesinde
"yak" yaziyir, farlar yaninca "sondur" yazacak
String txt_btn_flashers = "yak"; //varsayilan olarak dortlu sinyaller dugmesinde
"yak" yaziyir, farlar yaninca "sondur" yazacak
String txt_btn_12Voutlets = "on"; //varsayilan olarak cakmakliklara enerji ver
dugmesinde "yak" yaziyir, farlar yaninca "sondur" yazacak
String txt_btn_doorlocks = "Kilitleri_Ac"; //kilitleri ac dugmesinde hep ayni yazi
var
String txt_btn_ignition = "calistir"; //kilitleri ac dugmesinde hep ayni yazi var
String txt_btn_horn_greet = "Selam_Kornasi";
String txt_btn_panik = "PANIK";
```

```

// Açılışta LED'ler kapalı pozisyonda

EthernetServer server(80);
String data;
void setup()
{
  pinMode(po_headlights, OUTPUT);
  pinMode(po_parklights, OUTPUT);
  pinMode(po_flashers, OUTPUT);
  pinMode(po_12Voutlets, OUTPUT);
  pinMode(po_doorUnlock, OUTPUT);
  pinMode(po_doorLock, OUTPUT);
  pinMode(po_ignition, OUTPUT);
  pinMode(po_horn, OUTPUT);
  pinMode(po_wifi, OUTPUT); digitalWrite(po_wifi, HIGH);

  Serial.begin(9600);
  Ethernet.begin(mac, ip,gateway,subnet);
  server.begin();

  Serial.println("Serial READY");
  Serial.println("Ethernet READY");
  Serial.println("Server READY");
}

void loop()
{
  EthernetClient client = server.available();

  if(digitalRead(pi_charging)) {state_charging = 1; update_timers_as_charging(); }
  //arac calisiyor mu kontrol, eger sarj dinamosu voltaj uretiyorsa motor calisiyor demektir.

```

```
else {state_charging = 0; update_timers_as_not_charging(); } // arac sarja
baslayınca hemen farlari yak ve cakmakliklara enerji ver, sarj kesilince 1dk icinde
farlari ve cakmakliklari kapa
if(digitalRead(pi_mars_talebi)) fcn_otomars_trig();
if(digitalRead(pi_door_unlock)) fcn_door_unlock();
if(digitalRead(pi_door_lock)) fcn_door_lock();
```

```
unsigned long currentMillis = millis();
if ((unsigned long)(currentMillis - previousMillis) >= interval_seconds)
{
    if(state_charging) {update_timers_as_charging(); } //sarj ederken her saniye
basinda kontrol edilecek durumlar
    else {update_timers_as_not_charging();}
    previousMillis = currentMillis;
}
```

```
if(client)
{
    // HTTP talebi boş satırda bitecek.
    boolean current_line_is_blank = true;
    while (client.connected())
    {

        if(client.available())
        {
            char c = client.read();
            // http çağrısı bitti!
            if (inString.length() < 35) { inString.concat(c); }

            if (c == '\n' && current_line_is_blank)
            {
                // Standart HTML Başlıkları gönder
```

```

        client.println("HTTP/1.1 200 OK"); client.println("Content-Type: text/html");
client.println(); client.println("<html><body><form method=get>");
        client.println(" <p align= 'center' >
maerbirwork000643_ardu_ethernet_smart_car_with_html_buttons</p>");
        // LED Buton Döngüsünü Başlat

if(state_charging) { client.println("<br> Su anda arac calisiyor,"); }
else { client.println("<br> Su an arac calisMIyor,"); }

if(inString.indexOf(StrIgnition+"=calistir")>0 )
    { if(state_charging) { client.println("<br> Su anda ZATEN arac calisiyor,");
}
        else { Serial.println("motor calisacak"); fcn_otomars_trig(); delay(100);
fcn_otomars_trig(); txt_btn_ignition = "durdur"; }
    }

else if(inString.indexOf(StrIgnition+"=durdur")>0 )
    { if(state_charging) { Serial.println("motor duracak"); fcn_otomars_trig();
txt_btn_headlights = "calistir"; }
        else { client.println("<br> Su anda ZATEN arac calisiyor,"); }
    }

client.println("<br>"+StrIgnition+" <input type=submit
name="+StrIgnition+" value="+txt_btn_ignition+">");

//farlar
if(inString.indexOf(Headlights+"=yak")>0 )
    { Serial.println("Farlar Aktif"); fcn_turnon_lights(); txt_btn_headlights =
"sondur"; }
else if(inString.indexOf(Headlights+"=sondur")>0 )
    { Serial.println("Farlar Pasif"); fcn_turnoff_lights(); txt_btn_headlights =
"yak"; }

```

```

        client.println("<br>" + Headlights + " <input type=submit
name="+Headlights+" value="+txt_btn_headlights+">");

//park lambalari
if(inString.indexOf(Parklights+"=yak")>0 )
    { Serial.println("Park Lambalari Aktif"); digitalWrite(po_parklights,
HIGH); txt_btn_parklights = "sondur"; }
else if(inString.indexOf(Parklights+"=sondur")>0 )
    { Serial.println("Park Lambalari Pasif"); digitalWrite(po_parklights, LOW);
txt_btn_parklights = "yak"; }
    client.println("<br>" + Parklights + " <input type=submit name="+Parklights+"
value="+txt_btn_parklights+">");

//dortluler
if(inString.indexOf(Flashers+"=yak")>0 )
    { Serial.println("Park Lambalari Aktif"); digitalWrite(po_flashers, HIGH);
txt_btn_flashers = "sondur"; }
else if(inString.indexOf(Flashers+"=sondur")>0 )
    { Serial.println("Park Lambalari Pasif"); digitalWrite(po_flashers, LOW);
txt_btn_flashers = "yak"; }
    client.println("<br>" + Flashers + " <input type=submit name="+Flashers+"
value="+txt_btn_flashers+">");

//12V outlets cakmakliklar
if(inString.indexOf(Outlets12V+"=on")>0 )
    { Serial.println("12V Soketler Aktif"); digitalWrite(po_12Voutlets, HIGH);
txt_btn_12Voutlets = "off"; }
else if(inString.indexOf(Outlets12V+"=off")>0 )
    { Serial.println("12V Soketler Pasif"); digitalWrite(po_12Voutlets, LOW);
txt_btn_12Voutlets = "on"; }
    client.println("<br>" + Outlets12V + " <input type=submit
name="+Outlets12V+" value="+txt_btn_12Voutlets+">");

```

```

//Kapilar
if(inString.indexOf(DoorLocks+"=Kilitleri_Ac")>0 )
    { Serial.println("Kapi Kilitleri Acildi"); fcn_door_unlock();
txt_btn_doorlocks = "Kilitle"; }
else if(inString.indexOf(DoorLocks+"=Kilitle")>0 )
    { Serial.println("Kapilar Kilitledi"); fcn_door_lock(); txt_btn_doorlocks =
"Kilitleri_Ac"; }
client.println("<br>"+DoorLocks+" <input type=submit
name="+DoorLocks+" value="+txt_btn_doorlocks+">");

if(state_doorlocks) client.println("<br> Sistem tarafından Son olarak Kapilar
Kilitlendi,");
else client.println("<br> Sistem tarafından Son olarak Kapi Kilitleri
Acildi,");

//Selam Kornasi DIT DIT
if(inString.indexOf(StrHornGreet+"=Selam_Kornasi")>0 )
    { Serial.println("DIT DIT korna selami"); digitalWrite(po_horn, HIGH);
delay (200); digitalWrite(po_horn, LOW); delay (100);
digitalWrite(po_horn, HIGH);delay (200); digitalWrite(po_horn, LOW);}
client.println("<br>"+StrHornGreet+" <input type=submit
name="+StrHornGreet+" value="+txt_btn_horn_greet+">");

//PANİK
if(inString.indexOf(StrPanik+"=PANIK")>0 )
    { Serial.println("PANIK"); digitalWrite(po_flashers, HIGH);
digitalWrite(po_horn, HIGH); delay (500); digitalWrite(po_horn, LOW); delay
(500);
digitalWrite(po_horn, HIGH); delay (500); digitalWrite(po_horn, LOW);
delay (500); digitalWrite(po_horn, HIGH); delay (500); digitalWrite(po_horn,
LOW); delay (500);
digitalWrite(po_horn, HIGH); delay (500); digitalWrite(po_horn, LOW);
digitalWrite(po_flashers, LOW); }

```



```

        client.println("<br>" + StrPanic + " <input type=submit name="+StrPanic+"
value="+txt_btn_panic+">");

        client.println("</form></html></body>"); // Standart HTML Başlıkları kapat
        break;
    }
    if (c == '\n') {
        // Yeni satır başlat
        current_line_is_blank = true;
    }
    else if (c != '\r') {
        // Geçerli hat üzerinde karakter alındı.
        current_line_is_blank = false;
    }

}

}

// Tarayıcıya bilgi alış süresini bildir!
delay(1); inString = ""; client.stop();
}
}

void update_timers_as_charging() {fcn_turnon_lights(); fcn_energise_12Vsockets();
fcn_energise_wifi(); }
void update_timers_as_not_charging()
{
    if(timer_lights_off_latey)  {timer_lights_off_latey--;}  else
{fcn_turnoff_lights();}
    if(timer_12Vsockets_off_latey) {timer_12Vsockets_off_latey--;} else {
fcn_passivate_12Vsockets(); }
    if(timer_wifi_off_latey)  {timer_wifi_off_latey--;}  else
{fcn_passivate_wifi(); }

```

```
}
```

```
void fcn_otomars_trig() {digitalWrite(po_ignition, HIGH); delay (300);  
digitalWrite(po_ignition, LOW); delay (100);}
```

```
void fcn_door_unlock() {digitalWrite(po_doorUnlock, HIGH); delay(300);  
digitalWrite(po_doorUnlock, LOW); state_doorlocks = 0;}  
void fcn_door_lock() {digitalWrite(po_doorLock, HIGH); delay(300);  
digitalWrite(po_doorLock, LOW); state_doorlocks = 1;}
```

```
void fcn_turnon_lights() {digitalWrite(po_headlights, HIGH); state_headlights=1;  
digitalWrite(po_parklights, HIGH); state_parklights=1; timer_lights_off_latey =  
value_set_timer_lights;}  
void fcn_turnoff_lights() {digitalWrite(po_headlights, LOW); state_headlights=0;  
digitalWrite(po_parklights, LOW); state_parklights=0; }
```

```
void fcn_energise_12Vsockets() {digitalWrite(po_12Voutlets,HIGH);  
state_12Voutlets=1; timer_12Vsockets_off_latey = value_set_timer_12Vsockets;}  
void fcn_passivate_12Vsockets() {digitalWrite(po_12Voutlets,LOW);  
state_12Voutlets=0; }
```

```
void fcn_energise_wifi() {digitalWrite(po_wifi,HIGH); state_wifi = 1;  
timer_wifi_off_latey = value_set_timer_wifi;}  
void fcn_passivate_wifi() {digitalWrite(po_wifi,LOW); state_wifi = 0;}
```

EK 4. ANDROID STUDIO'DA HAZIRLANMIŞ “SEYYARE001” İSİMLİ UYGULAMANIN KODLARI

“MyActivity.java” isimli dosya

```
package com.maerbir.beyt001;

import Android.support.v7.app.ActionBarActivity;
import Android.os.Bundle;
import Android.view.View;
import Android.webkit.WebView;
import Android.webkit.WebViewClient;
import Android.widget.Button;

public class MainActivity extends ActionBarActivity {
    private WebView ObjWebView; //mWebView isminde bir WebView tipi class
    tanımladım
    String StrUrlHome = "http://192.168.2.101";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ObjWebView = (WebView) findViewById(R.id.webView01);
        ObjWebView.getSettings().setJavaScriptEnabled(true);
        ObjWebView.loadUrl(StrUrlHome); //WebView01 widgetı kontrol eden
        mWebView classı hangi url'yi acmaya calissin
        ObjWebView.setWebViewClient(new ClassWebViewClient()); //url linkinin
        default web tarayicisi ile egil ayni pencerede acilmasi icin
        //mWebView.setWebViewClient(new MyWebViewClient());
        addButtonClickListener();
    }
    //http linki acarken default browser değil de kendi içinde açılması için bu classı
    tanımladım
    private class ClassWebViewClient extends WebViewClient{
        @Override
```

```
public boolean shouldOverrideUrlLoading(WebView webView01, String url)
{
    ObjWebView.loadUrl(StrUrlHome);
    return true;
}
}
public void addButtonClickListener() {
    Button ObjBtnConnectHome = (Button)
findViewById(R.id.btn_connect_to_car);
    ObjBtnConnectHome.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ObjWebView.loadUrl(StrUrlHome);
        }
    });
}
}
```

“activity_my.xml” isimli dosya

```
<RelativeLayout xmlns:Android="http://schemas.Android.com/apk/res/Android"
    xmlns:tools="http://schemas.Android.com/tools"
    Android:layout_width="match_parent"
        Android:layout_height="match_parent"
    Android:paddingLeft="@dimen/activity_horizontal_margin"
        Android:paddingRight="@dimen/activity_horizontal_margin"
        Android:paddingTop="@dimen/activity_vertical_margin"
        Android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity"
    Android:id="@+id/LayoutMain">
```

```
<LinearLayout
    Android:orientation="vertical"
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:layout_centerVertical="true"
    Android:layout_centerHorizontal="true"
    Android:id="@+id/LLvert01"
    Android:gravity="bottom">
```

```
<WebView
    Android:layout_width="match_parent"
    Android:layout_height="match_parent"
    Android:id="@+id/webView01"
    Android:layout_gravity="center_horizontal"
    Android:layout_weight="1" />
```

```
<Button
    Android:layout_width="fill_parent"
    Android:layout_height="wrap_content"
    Android:text="@string/tr_connect"
    Android:id="@+id/btn_connect_to_car"
```

```
Android:layout_gravity="center_horizontal"
```

```
Android:elegantTextHeight="false"
```

```
Android:textSize="40dp" />
```

```
<TextView
```

```
Android:layout_width="wrap_content"
```

```
Android:layout_height="wrap_content"
```

```
Android:textAppearance="?Android:attr/textAppearanceSmall"
```

```
Android:text="by maerbir.com"
```

```
Android:id="@+id/tV_by_mae"
```

```
Android:layout_gravity="center_horizontal" />
```

```
</LinearLayout>
```

```
</RelativeLayout>
```

“String.xml” isimli dosya

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
  <string name="app_name">Beyt001</string>
```

```
  <string name="action_settings">Ayar Çek</string>
```

```
  <string name="tr_connect">Eve Baglan</string>
```

```
</resources>
```

“AndroidManifest.xml” isimli dosya

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns:Android="http://schemas.Android.com/apk/res/Android"  
    package="com.maerbir.beyt001" >
```

```
<application
```

```
    Android:allowBackup="true"
```

```
    Android:icon="@drawable/ic_launcher"
```

```
    Android:label="@string/app_name"
```

```
    Android:theme="@style/AppTheme" >
```

```
<activity
```

```
    Android:name=".MainActivity"
```

```
    Android:label="@string/app_name" >
```

```
<intent-filter>
```

```
    <action Android:name="Android.intent.action.MAIN" />
```

```
    <category Android:name="Android.intent.category.LAUNCHER" />
```

```
</intent-filter>
```

```
</activity>
```

```
</application>
```

```
<uses-permission Android:name="Android.permission.INTERNET"/>
```

```
</manifest>
```

EK 5. ARAÇ İÇİ ÇOK HATLI İLETİM KABLOSU HATLARIN TAŞIDIKLARI SİNYALLER ÇİZELGESİ

Hat No	Adı	Görevi
-1	0V / GND / toprak / şase	0V yada Toprak yada GND
0	İç Kilit Butonu Kilit Ucu	0V'a temas ettirilirse kapı kilitleri açılır
1	İç Kilit Butonu Aç Ucu	0V'a temas ettirilirse kapılar kilitlenir
2	arızalı	Arızalı
3	Park Lambaları	Normalde 12V var, 0V verilince Parklambaları yanar
4		
5	Kapılar Açık sinyali	Kapılar kapalı iken 12, Kapılar açılınca 0V olur, bu hatta hariçten 0V verilirse iç aydınlatma lambaları yanar.
6	Şarj Lambası	Gösterge Panelindeki Şarj Lambası, Kontak açık ve Şarj etmezken 0V, şarj dinamosu şarja başlayınca 12V
7	yedek	İleride ihtiyaç duyulursa diye ayrıldı
8	yedek	İleride ihtiyaç duyulursa diye ayrıldı
9	yedek	İleride ihtiyaç duyulursa diye ayrıldı
10	korna	0V verilirse korna aktif olur
11	Kontak 1 konumu	Anahtar I konumunda yada direksiyon kenarındaki güç düğmesi aktif edilirse 12V vardır
12	Kontak 2 konumu	Anahtar II konumunda yada motor çalıştırma modülü aktif edildi ise 12V vardır
13	Kontak 3 konumu	Anahtar III konumunda veya motor çalıştırma modülü marşa basıyorsa 12V vardır.
14	yedek	İleride ihtiyaç duyulursa diye ayrıldı
15	yedek	İleride ihtiyaç duyulursa diye ayrıldı
16	Oto marş sistemi	+5 verilince oto marş sistemine enerji iletilir

	energize sinyali	
17	Motor Start Stop	Motor durgunken 2 kere +12 pulse verilirse motor start alır, motor çalışırken 1 kere 12 V pulse verilirse motor Stop konumuna geçer (eğer anahtar II konumunda değilse) Not: bu pulse Kontakın III konumuna getirilmesi ile de verilebilir bunun için 13 nolu pin 17 nolu pine kısa devre edilmelidir.
18	yedek	İleride ihtiyaç duyulursa diye ayrıldı
19	yedek	İleride ihtiyaç duyulursa diye ayrıldı
20	Aktif Low Röle 1	0V verilince sağ sinyali aktive eder, 12V yada açık devre yapılırsa boşta
21	Aktif Low Röle 2	0V verilince sol sinyali aktive eder, 12V yada açık devre yapılırsa boşta
22	Aktif Low Röle 3	Araç içi 12V çakmaklık soket enerji kontrolü, 0V'a bağlanınca aktif
23	Aktif Low Röle 2	İleride ihtiyaç duyulursa diye ayrıldı, henüz bir iş atanmadı
24	Farlar	Normalde 12V var, 0V verilince Farlar yanar (uzun veya kısa direksiyon solundaki koldan kontrol edilir)
25	Toprak	(IR kontrol işlemcisinden)
26	5V	(IR kontrol işlemcisinden)
27	IR sinyali	(IR kontrol işlemcisine)

**EK 6. DONANIMIN 12 V'LUK AKÜDEN ÇEKTİĞİ AKIM BİLGİLERİ
ÇİZELGESİ**

Eleman	Akımı
Dörtlü sinyaller	8 A
Park Lambaları	2.13 A
12 Kontrol Kalemı	110 mA
RF Alıcı Devresi	10 mA
Honda Cpu Ardu Uno	30 mA