

**İSTANBUL AYDIN ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**

**ARDUINO UNO RC CAR**

**LİSANS TEZİ**

**Özgür BEKTAŞ**

**Yazılım Mühendisliği**

**Anabilim Dalı : Herhangi Mühendislik, Bilim  
Tez Danışmanı : Prof. Dr. Hasan İbrahim BALIK**

**HAZİRAN 2015**



**Özgür BEKTAŞ**, a **B.Sc.** student of IAU **Faculty of Engineering** student ID **B1105.090051**, successfully defended the **thesis** entitled “**ARDUINO UNO RC CAR**” , which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

**Tez Danışmanı :**      **Prof. Dr. Hasan Hüseyin BALIK**      .....

İstanbul Aydın Üniversitesi

**Teslim Tarihi : 1 Haziran 2015**



## **ÖNSÖZ**

Dünyada ve ülkemizde Arduino'nun kullanımı giderek yaygınlaşmaktadır. İnternette Arduino hakkında birçok İngilizce kaynak olmasına rağmen Türkçe kaynak bulmak oldukça zor. Umarım bu tez Arduino ile proje geliştirecek insanlar için bir kaynak olabilir.

Başta tez danışmanım ve bölüm başkanım Prof. Dr. Hasan Hüseyin Balık olmak üzere projenin gelişimi sırasında bana yardımcı olan Robotistan, Robot Kutusu firmalarının teknik destek ekiplerine çok teşekkür ederim.

Haziran 2015

Özgür BEKTAŞ  
(Öğrenci)

## İÇİNDEKİLER

	<u>Page</u>
ÖNSÖZ.....	Error! Bookmark not defined.
İÇİNDEKİLER.....	Error! Bookmark not defined.
KISALTMALAR.....	Error! Bookmark not defined.
FİĞÜR LİSTESİ .....	Error! Bookmark not defined.
<b>ÖZET .....</b>	<b>12</b>
1.1 Tezin Amacı .....	14
1.2 Sistem Çalışması .....	14
.....	15
<b>2. REMOTE CONTROL .....</b>	<b>15</b>
2.1 Tanım.....	15
2.2 Remote Control Hakkında Bilgi .....	15
<b>3. MALZEME LİSTESİ .....</b>	<b>17</b>
3.1 Arduino Uno R3 .....	17
3.1.1 Arduino Uno Teknik Özellikler .....	18
3.2 Ethernet Shield .....	19
3.2.1 Ürün Özellikleri.....	20
3.3 Motor driver L298N .....	21
3.4 RC Car Robot Kiti ve DC Motorlar .....	23
3.4.1 Ürün Özellikleri.....	24
3.5 M3 Montaj Sütunları .....	24
3.5.1 Ürün özellikleri .....	25
3.6 TP-LINK TL-MR3020 Portatif 3G/4G Kablosuz N Router .....	25
3.6.1 Ürün ÖzellikleriŞekil 3.10: TP-Link Router Özellik Listesi [13].....	26
3.7 12 Volt Kuru Akü .....	27
3.7.1 Ürün Özellikleri.....	27
3.8 Ataba 6 Volt 12 Volt Kuru Akü Şarj Aleti .....	28
3.8.1 Ürün Özellikleri.....	28
3.9 Arduino IDE .....	29
3.10 HTML.....	30
3.11 Visual Studio 2013 .....	32
<b>4. ARDUINO UNO RC CAR .....</b>	<b>32</b>
4.1 Projenin Amacı .....	23
4.2 Sistem Nasıl Çalışır .....	233
4.2.1 Arduino.....	23
4.2.1.1 Yazılan Kod.....	234
<b>SONUÇ .....</b>	<b>64</b>
<b>REFERANSLAR.....</b>	<b>65</b>
<b>ÖZGEÇMİŞ.....</b>	<b>66</b>



## KISALTMALAR

<b>AC</b>	: Alternating current
<b>API</b>	: Application programming interface
<b>C</b>	: C (Programing Language)
<b>C#</b>	: C Sharp (Programing Language)
<b>C++</b>	: Cee plus plus (Programing Language)
<b>DC</b>	: Direct Current
<b>FTDI</b>	: Future Technology Devices International
<b>FTP</b>	: File Transfer Protocol
<b>GCC</b>	: GNU Compiler Collection
<b>GNU</b>	: GNU's Not Unix
<b>GUI</b>	: Graphical User Interface
<b>HTML</b>	: HyperText Markup Language
<b>IO (I/O)</b>	: Input / Output
<b>IOREF</b>	: Input / Output Reference
<b>IP</b>	: Internet Protocol
<b>LAN</b>	: Local Area Network
<b>LiPo</b>	: Lithium Polymer
<b>mA</b>	: Milliampere
<b>MAC</b>	: Media Access Control
<b>mAh</b>	: Milliampere-hour
<b>Mbps</b>	: Megabits per second
<b>MCU</b>	: Microcontroller Unit
<b>MHz</b>	: Megahertz
<b>micro-SD</b>	: micro - Secure Digital
<b>NAT</b>	: Network address translation
<b>PoE</b>	: Power over Ethernet
<b>RC</b>	: Remote Control
<b>SPI</b>	: Serial Peripheral Interface
<b>WEP</b>	: Wired Equivalent Privacy
<b>Wi-Fi</b>	: Wireless Fidelity





## FIGÜR LİSTESİ

	<b><u>Page</u></b>
Şekil 1.1: Sistem Çalışması.....	15
Şekil 2.1: Leonardo Torres.....	16
Şekil 3.2: Arduino Uno Kart Şematiği.....	18
Şekil 3.3: Ethernet Shield.....	19
Şekil 3.4: Ethernet Shield Kart Şematiği.....	20
Şekil 3.5: Arduino Motor Sürücüsü.....	21
Şekil 3.6: L298N Datasheet.....	23
Şekil 3.7: Rc Car Robot Kiti ve DC Motorlar.....	23
Şekil 3.8: M3 Montaj Sütunları.....	24
Şekil 3.9: Tp-Link MR3010 Router.....	25
3.6.1 Ürün ÖzellikleriŞekil 3.10: TP-Link Router Özellik Listesi.....	26
Şekil 3.11 : 12 Volt Kuru Akü.....	27
Şekil 3.12: Ataba Kuru Akü Şarj Cihazı.....	28
Tablo 3.4 : Ataba 660 Cihaz Özellikleri.....	28
Şekil 3.13: Arduino IDE.....	29
Şekil 3.14: Arduino Uno Proje Örnekleri.....	30
Şekil 3.15: HTML.....	31
Şekil 3.16: Tim Berners-Lee.....	31
Şekil 4.1: Arduino Uno RC Car.....	31



## **ARDUINO UNO RC CAR**

### **ÖZET**

Günümüzde bir çok elektronik projede kullanılan Arduino programlanabilir çipi hem yazılımcılar hem de elektronikçiler için eşsiz bir çalışma ortamı sağlıyor. Gelişmekte olan teknoloji ile birlikte programlanabilir çiplerin kullanıcısı da artmaktadır. İhtiyacınız olan tek şey hayalgücünüzü zorlamak.

Bu projenin amacı, herhangi bir kapsama alanı sıkıntısı olmadan, internete bağlı Arduino Uno ile hazırlanmış olduğum arabanın kontrolünü yine internet üzerinden sağlamaktır.

Bu projenin sonunda, Arduino ile proje geliştirenler bu çipe entegre olan birçok destekleyici çipin kullanımını ve geliştirdikleri cihazların internet bağlantısı ile daha özgür kapsama alanlarına kavuşturabileceklerini göreceklerdir.



## 1. GİRİŞ

Teknolojinin gelişmesi ile birlikte, yaşam alanlarımızda da birtakım değişiklikler yaşamaya başladık. Mesela uzaktan kumandalı ışıklar, cep telefonu ile kontrol edilebilen kombiler, hareket sensörleri ile hırsızları tespit edebilen cihazlar. Bunlar yakın geçmişte hayal gibiydi. Bu sistemlerin temelini programlanabilir çipler oluşturmaktadır. Arduino firması bu amaca hizmet eden programlanabilir çipler piyasaya çıkarmaktadır. Kullanım açısından basit boyut ve özellik olarak birçok seçenek sunan Arduino, performansı ile de programcıları mutlu etmektedir. Bu projede Arduino Uno modeli ile bir RC Car geliştireceğim. [1].

### 1.1 Tezin Amacı

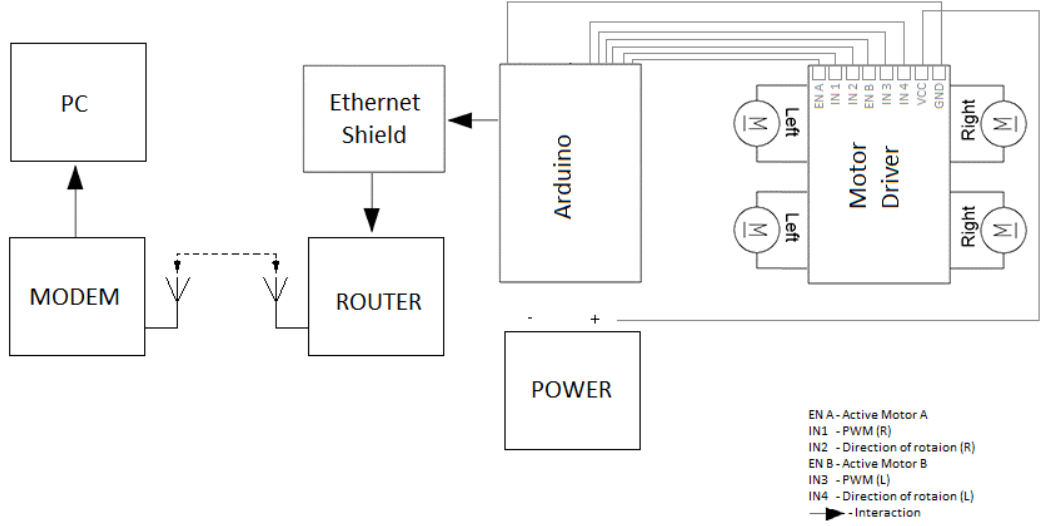
Bu tezin amacı, kullanımı giderek yaygınlaşan Arduino ve entegre sistemleri için anlaşılabilir bir kaynak yaratmaktır. Projede kullanacağımız çipler ve ek donanımlar ,

- Arduino Uno R3
- Arduino Ethernet Shield with POE
- Arduino Motor Driver L298N
- TP-Link TL-MR3020 Wireless and Router
- USB 3G Modem
- 12 Volt 1.3 Amper Kuru Akü

Bu parçaların birbirleri ile entegre bir şekilde haberleşmesi projenin asıl amacıdır.

### 1.2 Sistem Çalışması

Arduino RC car üzerindeki Ethernet Shield yardımı ile kendi içerisinde bir server oluşturur. Ethernet shield Tp-Link MR2030 yardımı ile internet ağına bağlanır. Ethernet shield programlanırken tanımlanmış olan server ip adresi ile internet olan herhangi bir cihazdan kurulan server a erişim sağlanır. Web sitesi gibi açılan sayfada Arduino RC Car için hareket tanımlamaları bulunmaktadır. Bu web sayfası kullanılarak Arduino Rc Car'ın kontrolü sağlanır. [1].



**Şekil 1.1:** Sistem Çalışması [3].

## 2. REMOTE CONTROL

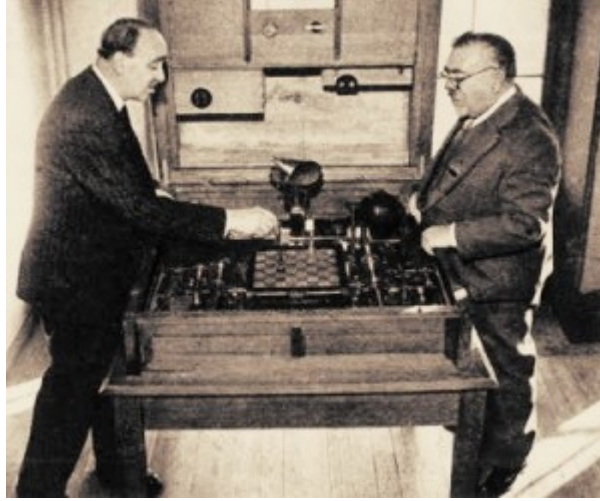
### 2.1 Tanım

Remote Control elektronik devrelerin işleyişine dışardan kablosuz bir şekilde müdahale edebilmek için geliştirilmiş sistemdir. Bu sistemin temellerini Nikola Tesla ve Leonarda Torres Quevedo atmıştır. Günümüzde hemen hemen her elektronik devre için kullanılabilen bu sistem askeri amaçlı birçok projede de kullanılmıştır. İlk çıktığı yıllarda elektromanyetik dalgalar ile sistem kontrolü sağlayan RC , teknolojinin gelişmesi ile internet üzerinden de kullanılabilir. [2].

### 2.2 Remote Control Hakkında Bilgi

Remote Control ilk olarak 1899 yılında Nikola Tesla tarafından hareket eden araçların elektronik mekanizmalarını control edebilmek için “Cihaz Tekniği” ismi ile geliştirildi. [3].

Leonardo Torres Quevedo 1903 yılında, Telekino adlı tasarımını Paris Bilim Akademisi'nde bir gösteri ile tanıttı. Bunun üzerine Fransa, İspanya, Birleşik Krallık ve ABD bu cihazın patentini aldı. Telekino emirleri elektromanyeti dalgalar ile alıp yerine getiren bir cihazdı. Remote Control alanında dünyada bir ilk olmuştur. Leonardo Torres bunun akabinde 1906 yılında kralın ve büyük bir kalabalığın önünde bir botu uzaktan kumanda etmiş ve başarılı olmuştur. Bu tekniği askeri amaçlı kullanmak isteyen Torres parasal sıkıntılardan dolayı çalışmalarını tamamlayamamıştır. [4].



Şekil 2.1: Leonardo Torres [6].

1932 yılında ilk uzaktan kumandalı model uçak icat edilmiştir. Bu süreçten sonra bu teknik yoğun olarak 2. Dünya Savaşı sırasında kullanılmıştır. Sonuç olarak Alman yapımı Wasserfall Füzesi ortaya çıktı. [5].



### 3. MALZEME LİSTESİ

#### 3.1 Arduino Uno R3



**Figure 3.1 :** Arduino Uno R3 [6]

Arduino, Processing/Wiring dilini kullanarak çevre elemanları ile temel giriş çıkış uygulamalarını gerçekleştiren açık kaynaklı fiziksel programlama platformudur. Arduino ile bağımsız olarak interaktif uygulamalar gerçekleştirilebilirsiniz. Aynı zamanda Arduinoyu bilgisayar ile Flash, Processing, MaxMSP, C Sharp gibi bir çok yazılım üzerinden yada kendi yazdığınız yazılımlarla haberleştirerek de kullanabilirsiniz.[7]

Arduino UNO R3, Arduino Uno'un en son çıkan modelidir. Bundan önceki modellerde (Uno, Duemilanove) bulunan tüm özellikleri desteklemektedir. UNO R3 modeliyle birlikte önceki versiyonlardaki 8U2 modeli yerine 16U2 modeli kullanılmıştır. Bu şekilde daha hızlı veri aktarımı daha az hafıza kullanılarak gerçekleştirilmiştir. Linux ve Mac bilgisayarlarda Arduino'yu bilgisayara bağlamak için herhangi bir driver'a ihtiyaç yoktur. Windows bilgisayarlarda Arduino IDE yazılımı içinde gelen inf dosyasını bilgisayarınıza tanıtmamız yeterlidir. Bu şekilde Arduino'nuzu bilgisayarınıza klavye, mouse, joystick ve benzeri aksesuarlar gibi takıp kullanılabilir hale getirebilirsiniz.

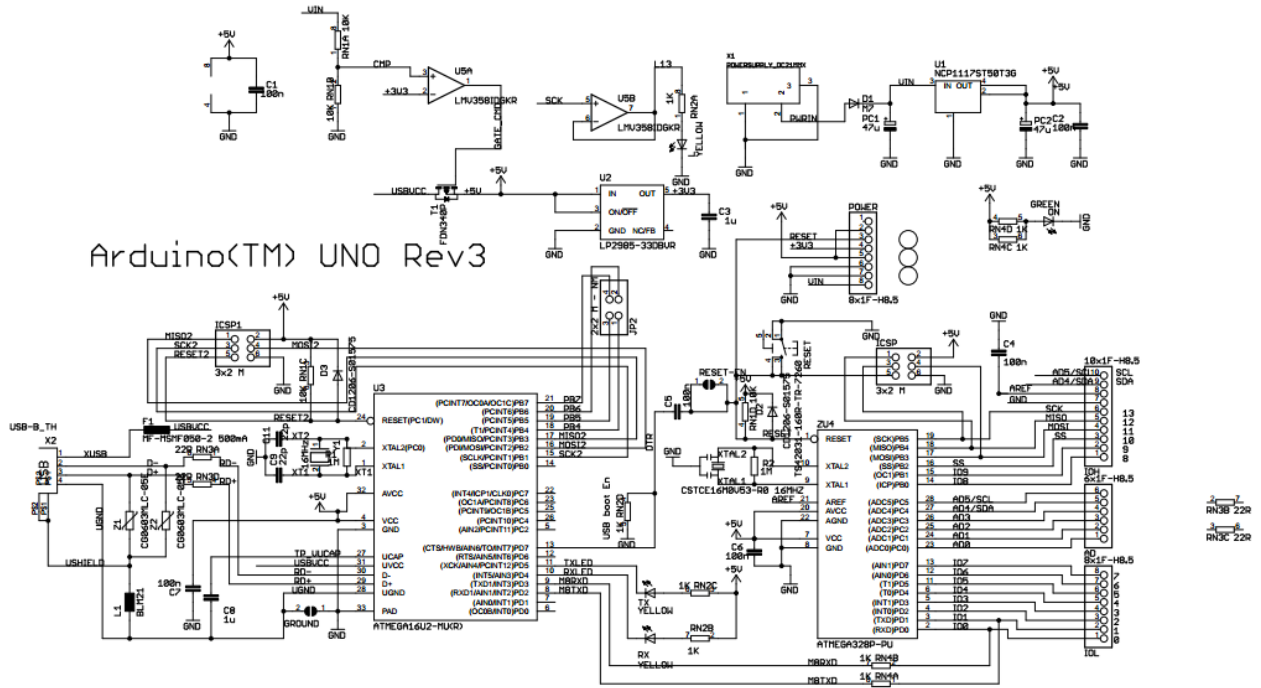
UNO R3 ekstradan SDA ve SCL pinlerine sahiptir bu pinler kart yerleşiminde AREF pininin yanında bulunmaktadır. Bununla birlikte kart üzerinde önceki versiyonlardan farklı olarak reset pininin yanına iki yeni pin eklenmiştir. Biri shieldlere kart üzerinden besleme sağlamak amacıyla IOREF pini diğeri ise ileride kullanılmak üzere ayrılmış boş bağlantısız pindir. UNO R3 piyasada bulunan tüm shieldler ile

uyumlu olup yeni pinleri ile de bundan sonra üretilecek olan yeni shieldlere de uyumlu haldedir.

### 3.1.1 Arduino Uno Teknik Özellikler

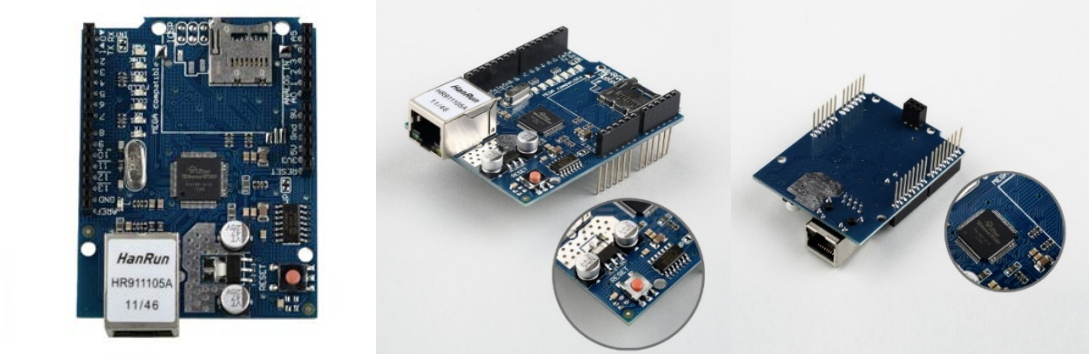
Tablo 3.1

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz



Şekil 3.2: Arduino Uno Kart Şematiği [8]

### 3.2 Ethernet Shield



Şekil 4.3: Ethernet Shield [9]

Arduino Ethernet Shield, Arduino platformları için hazırlanmış, ethernet üzerinden internete bağlanmanızı sağlar. Wiznet W5100 entegresini üzerinde barındıran bu kart hem TCP hem de UCP ile uyumludur. Arduino Ethernet Shield 4 eş zamanlı soket bağlantısını destekler. Shield üzerindeki standart RJ45 ethernet soketine ethernet kablosunu bağlayarak, Arduino ethernet kütüphanesi ile hızlı bir şekilde Arduino'yu internete bağlayabilirsiniz. [9]

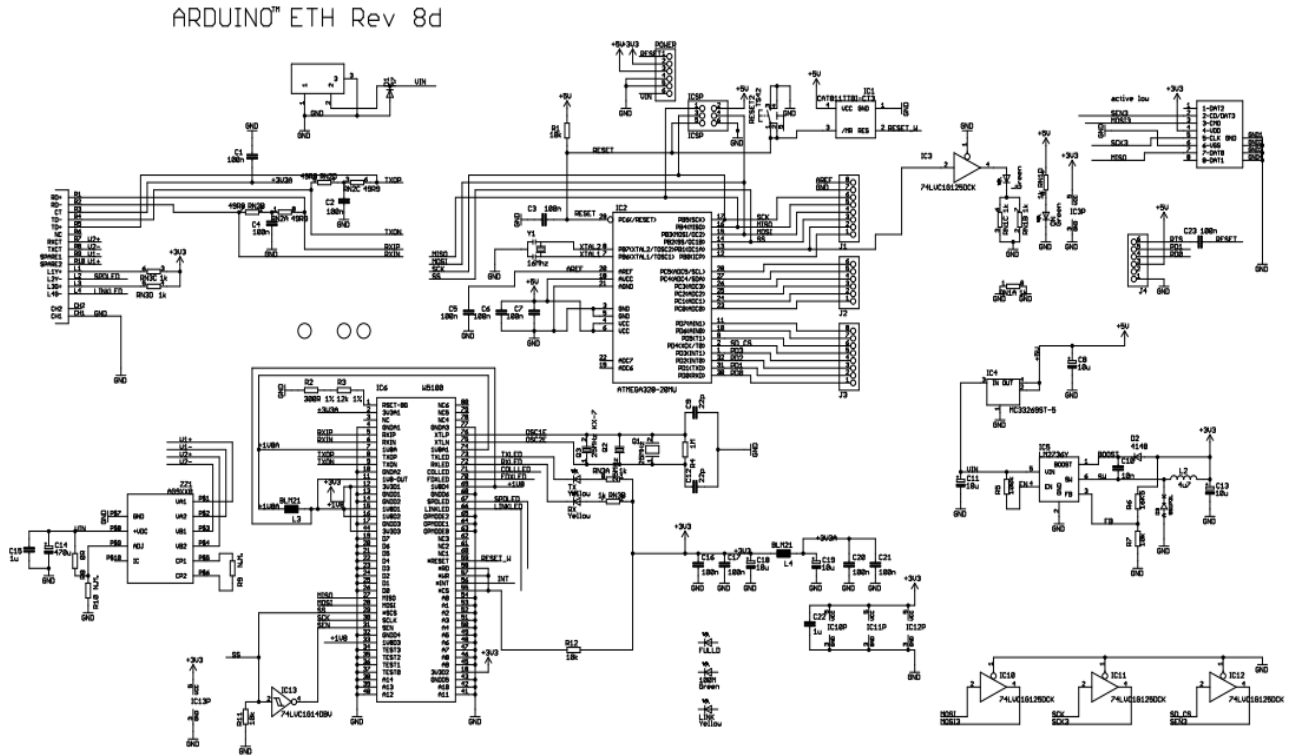
Kart üzerinde SD kart yuvası vardır. Bu sayede çeşitli bilgileri SD kart üzerine yazabilir ve okuyabilirsiniz. Bunun için Arduino Uno ve Mega ethernet kütüphanelerini kullanabilirsiniz. [9]

Arduino, Ethernet Shield işlemcisi olan Wiznet W5100 ile ve SD kart SPI hattı üzerinden haberleşmektedir. Bu pinler UNO'da 11,12 ve 13, MEGA'da 50,51 ve 52. pinlerdir. Her iki modelde de 10. pin W5100'ün, 4. pin ise SD kartın seçim pinidir. Bu pinler genel giriş çıkış pini olarak kullanılmazlar. Arduino MEGA'da donanımsal SS pini (53. pin) SD kart yada W5100 için kullanılmamaktadır. Fakat bu pin mutlaka çıkış pini olarak ayarlanmalıdır. [9]

Unutulmaması gereken önemli bir nokta W5100 ve SD kart aynı SPI hattını kullandığından aynı anda sadece biri aktif olmaktadır. Eğer iki haberleşme aynı anda yapılmak istenirse kullanılan kütüphanenin ortak olmasına dikkat edilmelidir. Aynı anda ikisini kullanmayacaksınız kullanmadığınız kısmı pasif etmeniz gerekmektedir. SD kartı pasif etmek için 4. pini çıkış olarak gösterip bu pinden 5V (high) çıkış vermeniz gerekmektedir. W5100'ü pasif hale getirmek için ise 10. pini çıkış olarak gösterip 5V (high) çıkış vermeniz gerekmektedir.

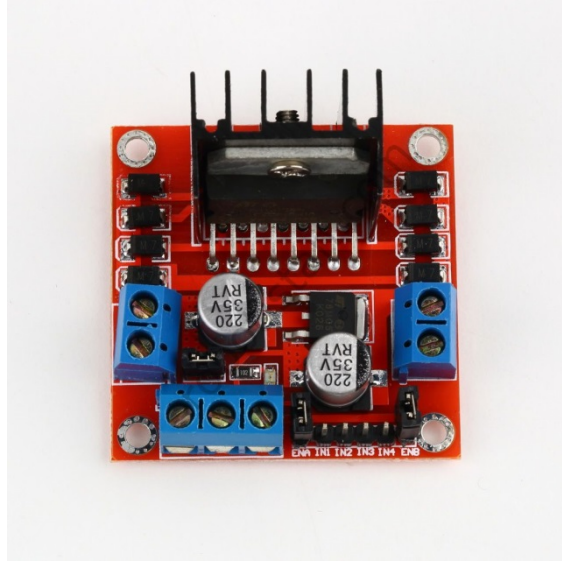
### 3.2.1 Ürün Özellikleri

- IEEE 802.3af protokolünü destekler.
- Wiznet W5100 entegresi üzerinden çalışır.
- Arduino Uno, Mega ve Seedeuno ile uyumludur.



Şekil 5.4: Ethernet Shield Kart Şematiği [10]

### 3.3 Motor driver L298N



Şekil 6.5: Arduino Motor Sürücüsü [11]

24V'a kadar olan motorları sürmek için hazırlanmış olan bu motor sürücü kartı, iki kanallı olup, kanal başına 2A akım vermektedir. Kart üzerinde L298N motor sürücü entegresi kullanılmıştır. Sumo, mini sumo, çizgi izleyen robotlarda ve çok çeşitli motor kontrol uygulamalarında kullanılabilir.

DC motorlardan ayrı olarak step motor kontrolüne de imkan sağlamaktadır.

#### 3.3.1 Ürün Özellikleri

Motor sürücü kartının üzerindeki giriş ve çıkışlar aşağıdaki gibi kullanılmaktadır.

- **ENA:** Sol motor kanalını aktif etme pini
- **IN1:** Sol motor 1. Girişi
- **IN2:** Sol motor 2. Girişi
- **IN3:** Sağ motor 1. Girişi
- **IN4:** Sağ motor 2. Girişi
- **ENB:** Sağ motor kanalını aktif etme pini

- **MotorA:** Sol motor çıkışı
- **MotorB:** Sağ motor çıkışı
- **VCC:** Besleme voltaj girişi(4.8V-24V)
- **GND:** Toprak bağlantısı
- **5V:** 5V çıkışı

Ürün üzerinde ayrıca jumper takılmış pinler de yer almaktadır. Bu pinler opsiyonel kullanımlar ve farklı özellikleri aktif etmek için kullanılabilir. [14]

**CSA:** A motor sürücü kanalının akım çıkışıdır. Buradan jumper çıkarılarak çekilen akım değeri analog voltaj olarak okunabilmektedir.

**CSB:** B motor sürücü kanalının akım çıkışıdır. Buradan jumper çıkarılarak çekilen akım değeri analog voltaj olarak okunabilmektedir.

**V1:** IN1 girişini direk olarak 5V'a çeken pull-up direncine bağlı olan bir jumperdir. Bu sayede siz toprağa çekmedikçe pine sürekli olarak 5V gelecektir.

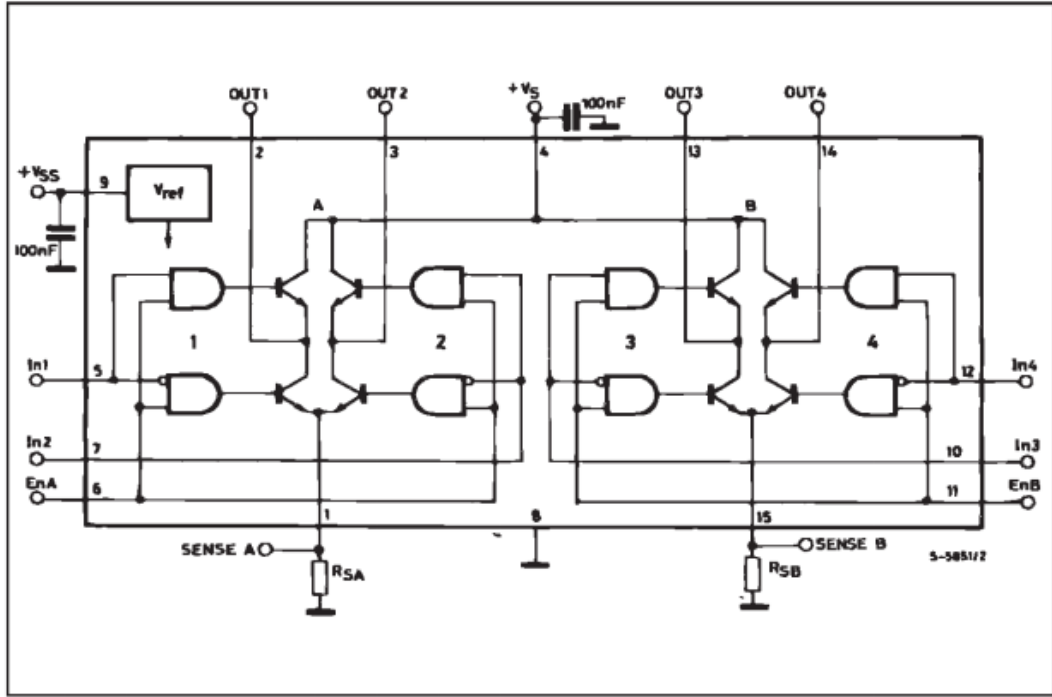
**V2:** IN2 girişini direk olarak 5V'a çeken pull-up direncine bağlı olan bir jumperdir. Bu sayede siz toprağa çekmedikçe pine sürekli olarak 5V gelecektir.

**V3:** IN3 girişini direk olarak 5V'a çeken pull-up direncine bağlı olan bir jumperdir. Bu sayede siz toprağa çekmedikçe pine sürekli olarak 5V gelecektir.

**V4:** IN4 girişini direk olarak 5V'a çeken pull-up direncine bağlı olan bir jumperdir. Bu sayede siz toprağa çekmedikçe pine sürekli olarak 5V gelecektir.

**5V-EN:** 7805 hattını aktif ve pasif hale getiren jumperdir. Takılı durumda kartın 5V çıkışı aktif olur ve buradan 5V çekilebilir. Çıkarıldığı durumda ise bu hat pasif konuma geçer.

L298N motor sürücü kartının datasheet tablosu aşağıdaki gibidir.



Şekil 7.6: L298N Datasheet [12]

### 3.4 RC Car Robot Kiti ve DC Motorlar



Şekil 8.7: Rc Car Robot Kiti ve DC Motorlar [16]

Hazır RC Car projeleri için tasarlanmış bu ürün sayesinde birçok projeye hareket kazandırabiliyoruz. Ürün 2 adet board ve 4 adet DC motordan oluşmaktadır.

### 3.4.1 Ürün Özellikleri

- Şasi Boyutu: 21.5cm x 14cm
- Lastik Boyutu: 7cm x 3cm
- Motor Boyutu: 7cm x 3.8cm x 2cm

**Tablo 3.2** DC Motor Teknik Özellikler

Voltage	DC 3V	DC 5V	DC 6V
Current	100 MA	100 MA	120 MA
Reduction rate		48:1	
RPM (With tire)	100	190	240
Tire Diameter		66 mm	
Car Speed (M/min)	20	39	48
Motor Weight (g)		50	
Motor Size		70mm*22mm*18mm	
Noise		<65dB	

### 3.5 M3 Montaj Sütunları



**Şekil 9.8:** M3 Montaj Sütunları



Bu sütunlar sayesinde arabaya çift katlı bir yapı kazandırılabilir. Böylece kartların ve materyallerin araba üzerine daha rahat yerleşimi sağlanıyor.

### 3.5.1 Ürün özellikleri

**Tablo 3.3:** M3 Pin Özellikleri

Name	Unit price	Number of	List Price
3 * 8 through pillars	0.25	16	4
15 +6 pillars	0.45	20	9
20 +6 pillars	0.5	10	5
40 +6 Tongzhu in	A	10	10
M3 nut	0.05	60	3

### 3.6 TP-LINK TL-MR3020 Portatif 3G/4G Kablosuz N Router



**Şekil 10.9:** Tp-Link MR3010 Router [13]

Uzaktan kontrol edilebilen projeler için ideal, performans konusunda tatmin edici TP-Link firmasının çıkarmış olduğu bu modem ile bir çok proje gerçekleştirebilirsiniz. Ürünün en dikkat çekici özelliği ise boyutudur. Bu tarz projelerde yer kaplayan cihazlar projeyi yer bakımından zora sokmaktadır. Bu modemi tercih etmemin asıl sebebi de ufak boyutlu olmasıdır.

### 3.6.1 Ürün Özellikleri

DONANIM ÖZELLİKLERİ	
Arabirim	1 10/100Mbps WAN/LAN Port, 3G/4G için USB 2.0 Portu, Güç kaynağı için bir mini USB portu
Düğme	Hızlı Güvenli Kurulum Düğmesi, Reset Düğmesi, Mod Anahtar
Harici Güç Kaynağı	5VDC/1.0A
Boyutlar ( W x D x H )	2.9 x 2.6 x 0.9 in. (74 x 67 x22 mm)
Anten Tipi	Dahili Anten

KABLOSUZ ÖZELLİKLERİ	
Kablosuz Standartları	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Frekans	2.4-2.4835GHz
EIRP	<20dBm
Kablosuz Modları	3G Router, Seyahat Router (AP), WISP Client Router
Kablosuz Güvenlik	64/128 bit WEP, WPA-PSK/WPA2-PSK, Kablosuz MAC Filtrelemesi

YAZILIM ÖZELLİKLERİ	
DHCP	Sunucu, DHCP Client Listesi, Adres Reservation
Port Yönlendirme	Virtual Server, Port Triggering, DMZ, UPnP
Erişim Kontrolü	Ebeveyn Kontrolü, Yerel Yönetim Kontrolü, Host Listesi, Erişim Zamanlama, Kural Yönetimi
Güvenlik	Firewall, MAC filtreleme, Denial of Service (DoS)

DİĞERLERİ	
Sertifika	CE, FCC, RoHS
Paket İçeriği	TL-MR3020 router, Hızlı Kurulum Kılavuzu, Güç Adaptörü, USB Kablosu, Ürün CD'si, RJ45 Ethernet Kablosu
Sistem Gereksinimleri	Microsoft® Windows® 98SE, NT, 2000, XP, Vista™ veya Windows 7, MAC® OS, NetWare®, UNIX® veya Linux.
Ortam	Çalışma Sıcaklığı: -10°C~60°C Depolama Sıcaklığı: -40°C~80°C Çalışma Nemi: 10%~90% yoğuşmasız Depolama Nemi: 5%~90% yoğuşmasız

Şekil 11.10: TP-Link Router Özellik Listesi [13]

### 3.7 12 Volt Kuru Akü



Şekil 12.11 : 12 Volt Kuru Akü [14]

Projede güç kaynağı olarak tercih ettiğim materyal kuru aküdür. Li-po pillerin yarattığı tehlikeden dolayı daha az risk arzeden kuru aküler boyut olarak büyük olsalarda dayanıklılık konusunda daha iyi durumdadır.

#### 3.7.1 Ürün Özellikleri

- Voltaj: 12V
- Kapasite: 1.3Ah
- Ağırlık: 0.57kg
- Boyutlar: 97mm x 43mm x 52mm

### 3.8 Ataba 6 Volt 12 Volt Kuru Akü Şarj Aleti



Şekil 13.12: Ataba Kuru Akü Şarj Cihazı [15]

Şarj edilebilir akülerin ve pillerin kullanımı kadar güvenli şarjı da büyük önem taşır. Bu bakımdan profesyonel bir firma tarafından yapılmış bir şarj aleti kullanmanın önemi tartışmasız büyüktür. Ataba firması bu konuda kendini kanıtlamış yerli firmalardan birisidir.

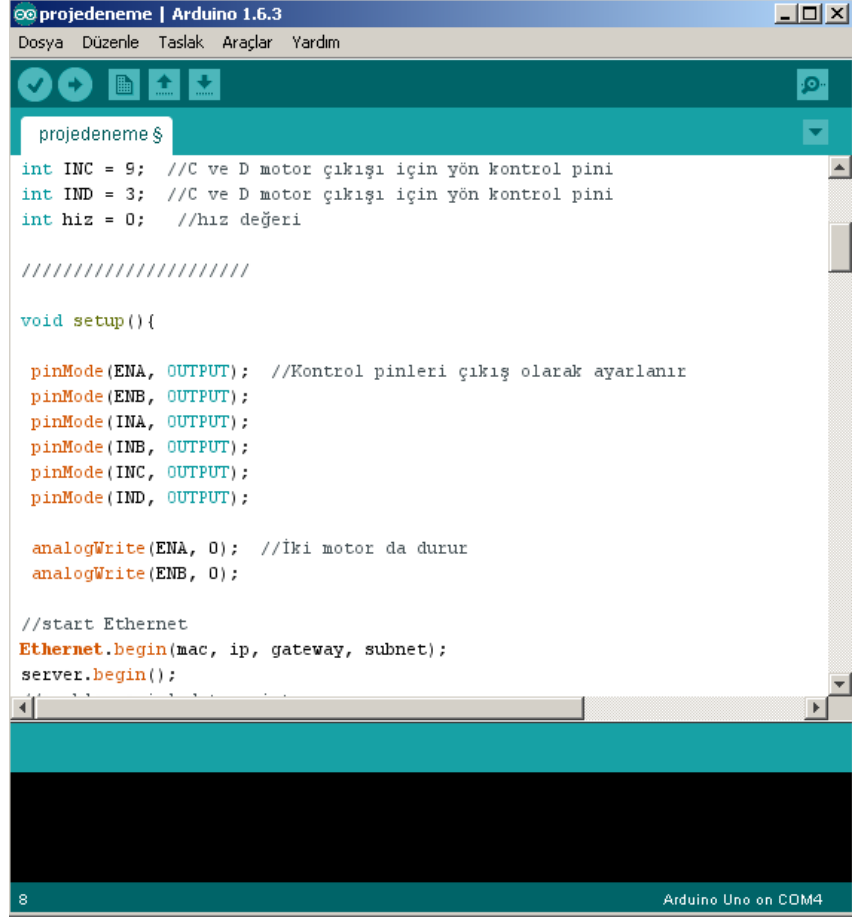
#### 3.8.1 Ürün Özellikleri

Tablo 14.4 : Ataba 660 Cihaz Özellikleri [15]

<b>Marka</b>	Ataba
<b>Model</b>	AT-660, AT 660 , AT660
<b>Giriş Voltajı</b>	230 V / 50 Hz
<b>Çıkış Voltajı</b>	6 V - 12 V DC
<b>Güç</b>	11.5 W
<b>Akım</b>	500 mA
<b>Led Gösterge</b>	VAR
<b>Barkod</b>	080558512057
<b>Menşei</b>	China
<b>Özellikler</b>	Kademeli

### 3.9 Arduino IDE

Arduino IDE Programı Java dilinde yazılmış, Arduino kartlarını programlarken ve yazılan kodları Arduino kartlarına yüklerken kullandığımız Arduino firmasının çıkarmış olduğu bir yazılımdır. Bu proje geliştirilirken Arduino IDE 1.6.3 sürümü kullanılmıştır. [16]



```
projedeneme $
int INC = 9; //C ve D motor çıkışı için yön kontrol pini
int IND = 3; //C ve D motor çıkışı için yön kontrol pini
int hiz = 0; //hız değeri

////////////////////

void setup(){

  pinMode(ENA, OUTPUT); //Kontrol pinleri çıkış olarak ayarlanır
  pinMode(ENB, OUTPUT);
  pinMode(INA, OUTPUT);
  pinMode(INB, OUTPUT);
  pinMode(INC, OUTPUT);
  pinMode(IND, OUTPUT);

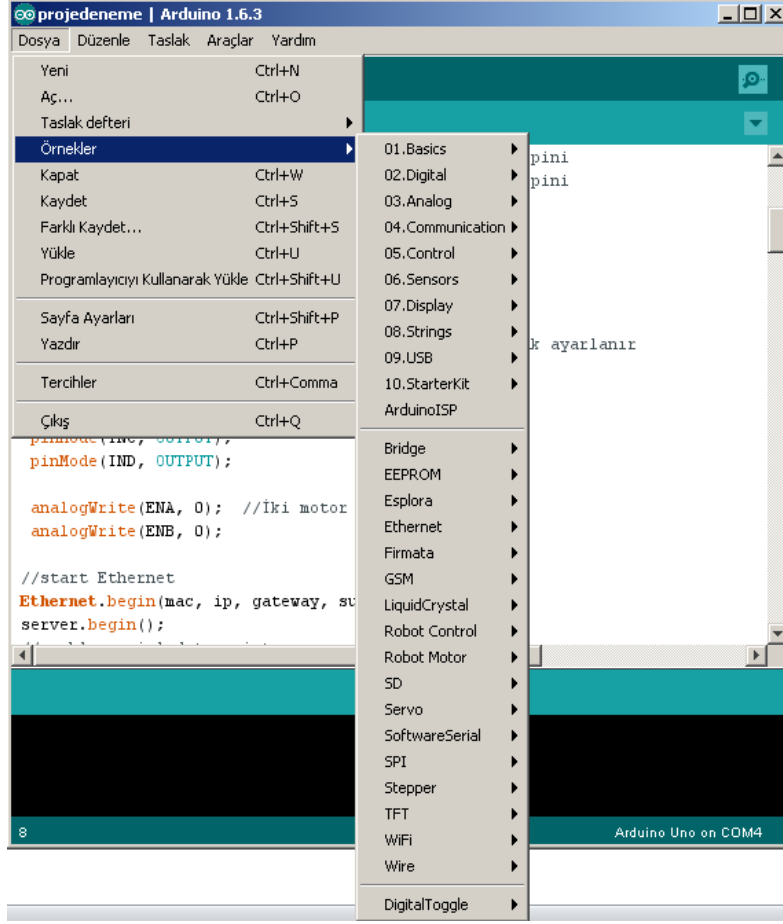
  analogWrite(ENA, 0); //İki motor da durur
  analogWrite(ENB, 0);

  //start Ethernet
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin();
}

8 Arduino Uno on COM4
```

Şekil 15.13: Arduino IDE

Bu yazılım Arduinonun resmi sitesinde ücretsiz ve açık kaynaklı olarak dağırılmaktadır. Yazılımı bilgisayara kurduktan sonra program içerisinde Arduino kart modelini ve bağlı olduğu portu belirleyip, programlamaya başlayabilirsiniz. Arduino IDE 'nin içerisinde birçok örnek proje mevcuttur. Gerekli donanımı edindikten sonra bu hazır projeler ile ilk denemelerinizi gerçekleştirebilirsiniz.

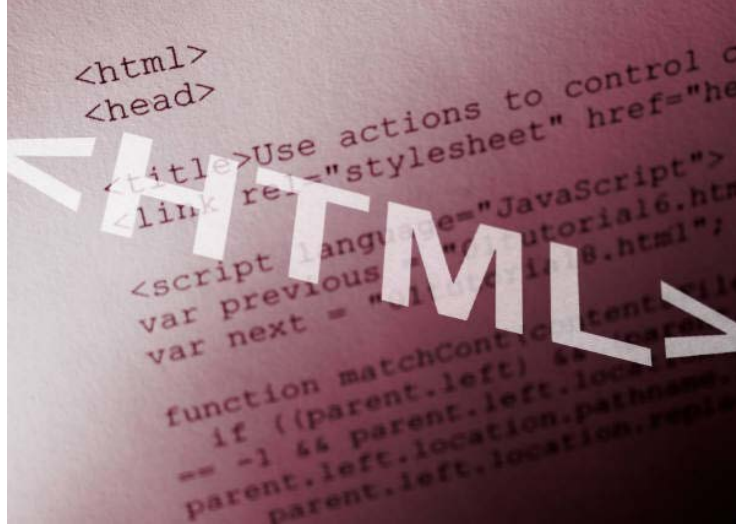


Şekil 16.14: Arduino Uno Proje Örnekleri

### 3.10 HTML

Türkçede Hiper Metin İşaretleme Dili , İngilizcede Hypertext Markup Language olarak bilinen bu yazılım dili web sayfalarını hazırlarken kullanılan bir metin işaretleme dilidir. [17]

HTML bir programlama dili değildir. Sadece HTML ile program yazılamaz. Bu dili yorumlayabilen programlar aracılığı ile bir program yazılabilir. Programlama dili olmamasının temel sebebi budur. [17]



**Şekil 17.15:** HTML

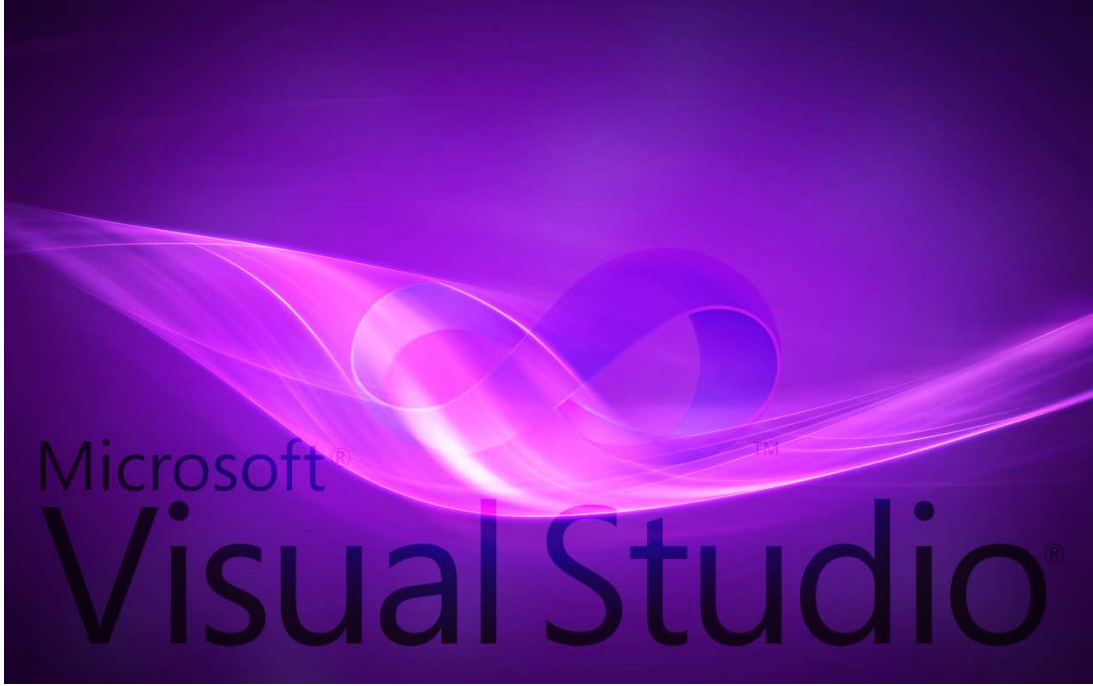
HTML in temelleri CERN ' de bilgi paylaşımını kolaylaştırmak adına 1980 yılında Tim Berners-Lee tarafından atılmıştır. Daha sonra 1990 yılında Worl Wide Web (WWW) sistemini kurmuştur. HTML hala günümüzün vazgeçilmez dilleri arasındandır. [23]



**Şekil 18.16:** Tim Berners-Lee

### 3.11 Visual Studio 2013

Visual Studio Microsoft tarafından geliştirilmiş tümleşik bir geliştirme platformudur. Virsüal Studio ile Microsoft Windows, Windows Mobile, .Net Framework, web uygulamaları, form uygulamaları ve web servisleri geliştirilebilmektedir.

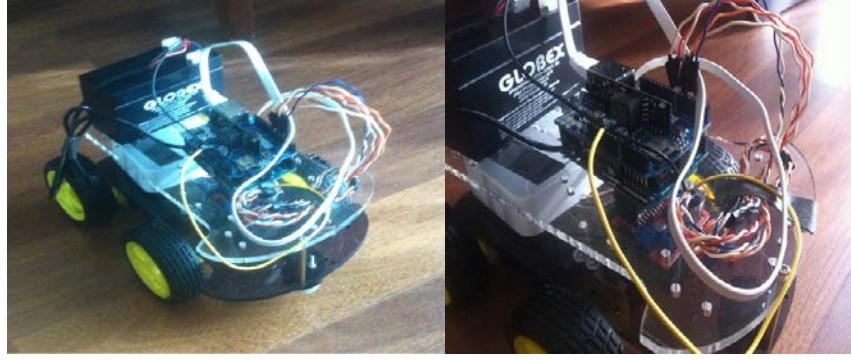


Şekil 4.1 : Visual Studio 2013

## 4. ARDUINO UNO RC CAR

Elektronik cihazlara daha özgür çalışma alanları kazandırma için geliştirmiş olduğum bu projede günümüzün gözdesi haline gelmiş programlanabilir çip olan Arduino Uno modeli kullandım. Ağa bağlı olan her cihazdan kolalıyla kontrol edilebilen bu araba bir çok amaç için kullanılabilir. Aracı programlarken Html ve Arduino Programlama Dili 'ni kullandım. Arduino basitleştirilmiş C++ dilini kullanır.





**Şekil 4.2 : Arduino Uno RC Car**

#### **4.1 Projenin Amacı**

Elektronik devrelere özgür çalışma alanları kazandırmak projenin ilk amacıdır. Bu proje günümüzde birçok amaç için kullanılabilir. Hazırlanmış olduğum RC Car projesi güvenlik amaçlı, tedbir amaçlı ve hobisel faaliyetler için kullanılabilceği gibi askeri amaçla da bir çok yerde kullanıldığı görülmektedir.

#### **4.2 Sistem Nasıl Çalışır**

Aynı ağa bağlı olan bir bilgisayar yada herhangi bir cihaz ile server için oluşturmuş olduğumuz ip adresi girilerek kontrol ekranına HTML tabanlı olarak ulaşılabilir. Açılan bu web sayfasından istenilen komut verilerek Rc Car 'ın kontrolü gerçekleştirilir. Sistem aynı ağ üzerinden çalışan makinelerle çalıştığı gibi server oluştururken gerekli ayarlamalar yapılarak 3G modem yardımı ile interneti olan herhangi bir cihazla da kontrol edilebilir.

##### **4.2.1 Arduino**

Projede Arduino ana rolü üstlenir. Diğer donanımlara verilen emirler ve gerekli yazılım arduino üzerinde yorumlanır bu kısımdan dağıtılır. Arduino Uno modeli için yazılmış kod iki ana fonksiyondan oluşmaktadır.

- Setup
- Loop

Bu iki fonksiyon Arduino'yu oluşturan temel fonksiyonlardır. Setup fonksiyonu genel olarak gerekli donanımların Arduino'ya tanıtıldığı ,pinlerin tanımlamalarının

yapıldığı kısımdır. Loop fonksiyonu arduino çalışırken verilen emirlerin okunduğu kısımdır. Tanımlamalar yapıldıktan sonra asıl loop fonksiyonunda oluşturulur. [24]

#### 4.2.1.1 Yazılan Kod

- **Kullanılan Kütüphaneler**

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

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

Arduino kod kısmında Ethernet Shield'in kullandığı bu iki kütüphane data transferini ve Arduino ile Ethernet Shield in haberleşmesini sağlamaktadır.

- **Ethernet Shield Server Kurulumu**

```
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0F, 0x08, 0xAF };
```

```
byte ip[] = { 192, 168, 0, 199 };
```

```
byte gateway[] = { 192, 168, 0, 100 };
```

```
byte subnet[] = { 255, 255, 255, 0 };
```

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

**Byte mac:** Bu kod Ethernet Shield için tanımlanmış MAC adresini yazdığımız kısımdır. Bu Mac adresi Ethernet Shield'in üzerinde yada Ethernet Shield ile birlikte gelen dökümanlar içerisinde yazılı olarak gelmektedir.

**Byte ip:** Server için ağ üzerinde boş bir ip adresi tanımlamamız gerekiyor. Bu ip adresini daha sonra RC car ın kontrolü için tarayıcıya yazıp işlemleri açılan sayfadan gerçekleştireceğiz.

**Byte gateway:** IPV4 Adresini yazacağımız kısım burasıdır. Ağımızın almış olduğu ip adresidir.

**Byte subnet:** Ağımızın alt ağ maskesi adresini gireceğimiz kısım bu kısımdır.

**EthernetServer server(80):** Serverimizin kullanmış olduğu portu bu kısımda tanımlayacağız.

- **Motor Sürücüsünün Tanımlanması**

```
int ENA = 5;
```

```
int ENB = 6;
```

```
int INA = 7;
```

```
int INB = 8;
```

```
int INC = 9;
```

```
int IND = 3;
```

```
int hiz = 0;
```

Motorları sürebilmek için motor sürücüsü tanımlamaları bu kısımda yapılmaktadır. ENA ve ENB kısımları motorların gidip gitmeyeceğini belirlemek için kullanılır. INA kısımlarını ise motorların hareket yönlerini belirlemek için kullanacağız. Bu kontrolleri Arduino üzerindeki pinleri motor sürücüsüne atayarak yapacağız. Bu kısımda önemli olan, Ethernet Shield ve motor sürücüsü için farklı bin seçimi yapmaktır. 10 , 11 ,12 ve 13. Pinleri Ethernet Shield kullanmaktadır. Bu yüzden motor sürücüsü için kullanacak pinleri bu pinlerin dışında kalan pinlerden seçmemiz gerekmektedir.

- **Setup Fonksiyonu**

```
pinMode(ENA, OUTPUT);
```

```
pinMode(ENB, OUTPUT);
```

```
pinMode(INA, OUTPUT);
```

```
pinMode(INB, OUTPUT);
```

```
pinMode(INC, OUTPUT);
```

```
pinMode(IND, OUTPUT);
```

```
analogWrite(ENA, 0);
```

```
analogWrite(ENB, 0);
```

```
Ethernet.begin(mac, ip, gateway, subnet);
```

```
server.begin();
```

```
Serial.begin(9600);
```

Setup fonksiyonunda motor sürücüsü için seçmiş olduğumuz pinler çıkış (output) olarak ayarlıyoruz. ENA ve ENB fonksiyonlarını 0 yapmalıyız ki proje çalışmaya başladığında motorlar hareket etmesin. Daha sonra ethernet shield i başlatmak için tanımlamış olduğumuz bilgiler yardımı ile ağımızı oluşturuyoruz.

- **LOOP Fonksiyonu**

```
EthernetClient client = server.available();  
  
if (client) {  
  
while (client.connected()) {  
  
if (client.available()) {  
  
char c = client.read();  
  
  
//read char by char HTTP request  
  
if (readString.length() < 100) {  
  
  
  
//store characters to string  
  
readString += c;  
  
//Serial.print(c);  
  
}  
  
  
  
//if HTTP request has ended  
  
if (c == '\n') {  
  
  
  
//////////  
  
Serial.println(readString); //print to serial monitor for debugging  
  
client.println("HTTP/1.1 200 OK"); //send new page  
  
client.println("Content-Type: text/html");  
  
client.println();
```

```
client.println("<HTML>");
client.println("<HEAD>");
client.println("<TITLE>Home Automation</TITLE>");
client.println("</HEAD>");
client.println("<BODY bgcolor='#E6E6E6'>");
client.println("<H1>Home Automation</H1>");
client.println("<hr />");
client.println("<br />");

client.println("<a href='\"/?ileri\"'>Ileri</a>");
client.println("<br />");
client.println("<br />");
client.println("<a href='\"/?geri\"'>Geri</a>");
client.println("<br />");
client.println("<br />");
client.println("<a href='\"/?sol\"'>Sola</a>");
client.println("<br />");
client.println("<br />");
client.println("<a href='\"/?sag\"'>Saga</a>");
client.println("<br />");
client.println("<br />");
client.println("<a href='\"/?dur\"'>Dur</a>");

client.println("</BODY>");
client.println("</HTML>");
```

```
delay(1);

//stopping client

client.stop();

////////// control arduino pin

if(readString.indexOf("?ileri") >0)//checks for on
{
  ileri() ;
}

if(readString.indexOf("?dur") >0)//checks for off
{
  dur();
}

if(readString.indexOf("?geri") >0)//checks for off
{
  geri();
}

if(readString.indexOf("?sol") >0)//checks for on
{
  sol() ;
}
```

```
if(readString.indexOf("?sag") >0)//checks for on
```

```
{  
  sag() ;  
}
```

```
readString=""; //clearing string for next read
```

```
}}}}}
```

Loop fonksiyonu proje çalıştığı sırada alacağı emirlerin tanımlanmış olduğu kodların bulunduğu kısımdır. Serverimizi ağa bağladıktan sonra serverin içerik kısmını yani HTML kısmını bu kısımda oluşturuyoruz. Daha sonra sayfada bulunan ileri , geri, sağa ve sola butonlarına basıldığı zaman hangi kodların çalışacağını tanımlıyoruz. Bu kontrolleri tanımlamış olduğumuz bir string yardımı ile post methodu ile gerçekleştiriyoruz. Yani Server üzerinden ileri butonuna bastığımız zaman html link ile hazırlanmış olan bu buton server a ileri stringini post edecektir. Bizde hazırlamış olduğumuz fonksiyonlar ile motora ileri kabiliyeti kazandıracaktır.

- **Void Fonksiyonları**

```
void ileri()
```

```
{  
  digitalWrite(INA, LOW);  
  digitalWrite(INB, HIGH);  
  digitalWrite(INC, HIGH);  
  digitalWrite(IND, LOW);  
  analogWrite(ENA, 100);  
  analogWrite(ENB, 100);  
}
```

```
void geri()
```

```
{  
  
    digitalWrite(INA, HIGH);  
    digitalWrite(INB, LOW);  
    digitalWrite(INC, LOW);  
    digitalWrite(IND, HIGH);  
    analogWrite(ENA, 100);  
    analogWrite(ENB, 100);  
}
```

```
void dur()
```

```
{  
    digitalWrite(INA, LOW);  
    digitalWrite(INB, LOW);  
    digitalWrite(INC, LOW);  
    digitalWrite(IND, LOW);  
    analogWrite(ENA, 0);  
    analogWrite(ENB, 0);  
}
```

```
void sol()
```

```
{  
    digitalWrite(INA, HIGH);  
    digitalWrite(INB, LOW);  
    digitalWrite(INC, HIGH);  
    digitalWrite(IND, LOW);
```



```
analogWrite(ENA, 100);  
analogWrite(ENB, 100);  
}
```

```
void sag()  
{  
digitalWrite(INA, LOW);  
digitalWrite(INB, HIGH);  
digitalWrite(INC, LOW);  
digitalWrite(IND, HIGH);  
analogWrite(ENA, 100);  
analogWrite(ENB, 100);  
}
```

Loop fonksiyonunun dışında tanımladığımız bu fonksiyonlar kod hammallığından kurtulmak için yazılmış fonksiyonlardır. Dört adet Void fonksiyonumuz bulunmaktadır. Bunlar ileri, ger, sağ, ve sol fonksiyonlarıdır. Motora yön vermek için gereken kodlar bulunmaktadır. IN değerlerini LOW ve HIGH yaparak motorların dönüş yönlerini fonksiyonların içinde ayarlıyoruz. EN değerlerini aktif edip motorlara güç veriyoruz ve bu motorların dönüş hızını EN, hız değeri girerek 0 ile 255 arası bir değer tanımlıyoruz.

Arduino'nun içerisindeki kod yükünü azaltmak ve küçük olan hafızasında daha az alan kaplamak için ASP.Net ile RC Car ın kontrol edilebildiği bir web uygulaması geliştiriyorum. Bu kısımda önemli olan geliştirdiğimiz uygulama ile Arduinonun haberleşme mantığı.

Arduino Ethernet Shield yardımı ile kurmuş olduğumuz web serverın , yaptığımız web uygulaması ile haberleşmesi TCP Client mantığına dayanıyor. ControlService.cs sayfasında yazılan kod aşağıdaki gibidir.

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.IO;
```

```
using System.Linq;
```

```
using System.Net.Sockets;
```

```
using System.Web;
```

```
using System.Web.Script.Services;
```

```
using System.Web.Services;
```

```
[WebService(Namespace = "http://tempuri.org/")]
```

```
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
```

```
[ScriptService]
```

```
public class ControlService : System.Web.Services.WebService {
```

```
    public ControlService () {
```

```
    }
```

```
    private void Forward()
```

```
    {
```

```
        try
```

```
{  
    TcpClient tcpac = new TcpClient("192.168.0.199", 4444);  
    NetworkStream ns = tcpac.GetStream();  
    StreamWriter sw = new StreamWriter(ns);  
    sw.Write("F");  
    sw.Flush();  
    tcpac.Close();  
}  
catch  
{  
}  
}
```

```
private void Back()  
{  
    try  
    {  
        TcpClient tcpac = new TcpClient("192.168.0.199", 4444);  
        NetworkStream ns = tcpac.GetStream();  
        StreamWriter sw = new StreamWriter(ns);  
        sw.Write("B");  
        sw.Flush();  
        tcpac.Close();  
    }  
}
```

```
    catch
    {
    }
}

private void Left()
{
    try
    {
        TcpClient tcpac = new TcpClient("192.168.0.199", 4444);
        NetworkStream ns = tcpac.GetStream();
        StreamWriter sw = new StreamWriter(ns);
        sw.Write("L");
        sw.Flush();
        tcpac.Close();
    }
    catch
    {
    }
}
```

```
private void Right()
{
    try
```

```
{  
    TcpClient tcpac = new TcpClient("192.168.0.199", 4444);  
    NetworkStream ns = tcpac.GetStream();  
    StreamWriter sw = new StreamWriter(ns);  
    sw.Write("R");  
    sw.Flush();  
    tcpac.Close();  
}  
catch  
{  
}  
}
```

**private void ForwardRight()**

```
{  
    try  
    {  
        TcpClient tcpac = new TcpClient("192.168.0.199", 4444);  
        NetworkStream ns = tcpac.GetStream();  
        StreamWriter sw = new StreamWriter(ns);  
        sw.Write("X");  
        sw.Flush();  
        tcpac.Close();  
    }  
}
```

```

    catch
    {
    }
}

private void ForwardLeft()
{
    try
    {
        TcpClient tcpac = new TcpClient("192.168.0.199", 4444);
        NetworkStream ns = tcpac.GetStream();
        StreamWriter sw = new StreamWriter(ns);
        sw.Write("Y");
        sw.Flush();
        tcpac.Close();
    }
    catch
    {
    }
}

private void BackRight()
{
    try
    {
        TcpClient tcpac = new TcpClient("192.168.0.199", 4444);

```

```

        NetworkStream ns = tcpac.GetStream();

        StreamWriter sw = new StreamWriter(ns);

        sw.Write("Z");

        sw.Flush();

        tcpac.Close();
    }

    catch

    {

    }

}

private void BackLeft()
{
    try
    {
        TcpClient tcpac = new TcpClient("192.168.0.199", 4444);

        NetworkStream ns = tcpac.GetStream();

        StreamWriter sw = new StreamWriter(ns);

        sw.Write("V");

        sw.Flush();

        tcpac.Close();
    }

    catch

    {

    }
}

```

```
}
```

```
[WebMethod]
```

```
public void Stop()
```

```
{
```

```
    try
```

```
    {
```

```
        TcpClient tcpac = new TcpClient("192.168.0.199", 4444);
```

```
        NetworkStream ns = tcpac.GetStream();
```

```
        StreamWriter sw = new StreamWriter(ns);
```

```
        sw.Write("S");
```

```
        sw.Flush();
```

```
        tcpac.Close();
```

```
    }
```

```
    catch
```

```
    {
```

```
    }
```

```
}
```

```
[WebMethod]
```

```
public void SmartControl(string controlKey)
```

```
{
```

```
    try
```

```
    {
```

```
        if (controlKey == "F")
```



```
    Forward();  
else if (controlKey == "B")  
    Back();  
else if (controlKey == "L")  
    Left();  
else if (controlKey == "R")  
    Right();  
else if (controlKey == "FR")  
    ForwardRight();  
else if (controlKey == "RF")  
    ForwardRight();  
else if (controlKey == "FL")  
    ForwardLeft();  
else if (controlKey == "LF")  
    ForwardLeft();  
else if (controlKey == "BR")  
    BackRight();  
else if (controlKey == "RB")  
    BackRight();  
else if (controlKey == "BL")  
    BackLeft();  
else if (controlKey == "LB")  
    BackLeft();
```

```
    }  
    catch  
    {  
    }  
}  
}
```

Server ile bağlantı kurmamızı , web sitesinden gelen emirleri Arduinoya aktarmamızı sağlayan kod bloğu,

```
TcpClient tcpac = new TcpClient("192.168.0.199", 4444);  
  
    NetworkStream ns = tcpac.GetStream();  
  
    StreamWriter sw = new StreamWriter(ns);  
  
    sw.Write("F");  
  
    sw.Flush();  
  
    tcpac.Close();
```

kısımındadır. Tanımlamış olduğumuz ip adresi ve port numarası üzerinden yazmış olduğumuz fonksiyonlar yardımı ile Arduino'nun anlayacağı dile çevirip gönderiyoruz. "sw.Write("F");" kısmındaki "F" arduinoda Forward olarak tanımlanmıştır. Dolayısıyla bu "İleri" komutunu veren fonksiyona ait kod parçasıdır.

Bu sayfada birtek dur() ve SmartControl() fonksiyonları web method olarak tanımlanmıştır. Bunun sebebi bu fonksiyonları digger sayfalarda da aktif olarak kullanılıyor olmalarıdır.

SmartControl servisi web uygulamasının ana sayfasından gelen tuş kombinasyonlarını Jquery ve Ajax yardımı ile algılayan stringin içeriğine göre yazmış olduğumuz fonksiyonları çalıştıran servistir.

Uygulamamızın anasayfası olan ControlPage.aspx deki kodlarımız şu şekildedir.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ControlPage.aspx.cs" Inherits="ControlPage" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title>Arduino Control Page</title>
```

```
<link href="Style/Css/StyleSheet.css" rel="stylesheet" />
```

```
<script type="text/javascript" src="Scripts/jquery-1.10.2.js"></script>
```

```
<script>
```

```
var CallSmartControl = function (controlText) {
```

```
    $.ajax({
```

```
        type: "POST",
```

```
        contentType: "application/json; charset=utf-8",
```

```
        url: "http://localhost:16893/ControlService.asmx/SmartControl",
```

```
        data: "{ 'controlKey': " + controlText + " }",
```

```
        dataType: "json",
```

```
        success: function () {
```

```
        },
```

```
        error: function (a) {
```

```
            alert(a.responseText);
```

```
        }
```

```
    });
```

```
}
```

```
var CallControlStop = function () {
```

```
    $.ajax({
```

```
        type: "POST",
```

```
        url: "http://localhost:16893/ControlService.aspx/Stop",
```

```
        data: "",
```

```
        dataType: "text",
```

```
        success: function () {
```

```
            $('#LblResult').val("Stop");
```

```
        },
```

```
        error: function () {
```

```
            $('#LblResult').val("Stop");
```

```
        }
```

```
    });
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <div class="main">
```

```
            <div id="displayCommand">
```

```
                </div>
```

```
<div class="forward">
```

```
    <asp:Button ID="Button1" OnClientClick="CallSmartControl('F')"  
    runat="server" Text="Forward" Width="80" Height="50" />
```

```
</div>
```

```
<div class="left">
```

```
    <asp:Button ID="Button3" OnClientClick="CallSmartControl('L')"  
    runat="server" Text="Left" Width="80" Height="50" />
```

```
</div>
```

```
<div class="stop">
```

```
    <asp:Button ID="Button5" OnClientClick="CallControlStop()"  
    runat="server" Text="Stop" Width="80" Height="50" />
```

```
</div>
```

```
<div class="right">
```

```
    <asp:Button ID="Button4" OnClientClick="CallSmartControl('R')"  
    runat="server" Text="Right" Width="80" Height="50" />
```

```
</div>
```

```
<div class="backward">
```

```
    <asp:Button ID="Button2" OnClientClick="CallSmartControl('B')"  
    runat="server" Text="Backward" Width="80" Height="50" />
```

```
</div>
```

```
    <asp:Label ID="LblResult" runat="server" ClientIDMode="Static"  
    Text=""></asp:Label>
```

**</div>**

**<script>**

```
var TotalCommand = '';  
$(document).ready(function () {  
    $(document).keydown(function (e) {  
        if (e.keyCode == 38) { //forward  
            AddCommand('F');  
            CallSmartControl(TotalCommand)  
        }  
        else if (e.keyCode == 40)//Back  
        {  
            AddCommand('B');  
            CallSmartControl(TotalCommand);  
        }  
        else if (e.keyCode == 37) { //Left  
            AddCommand('L');  
            CallSmartControl(TotalCommand);  
        }  
        else if (e.keyCode == 39) { //Right  
            AddCommand('R');  
            CallSmartControl(TotalCommand);
```

```

    }

    $('#displayCommand').text(TotalCommand);

});

});

var AddCommand = function (commandText) {

    if (commandText == "F"

        && TotalCommand.indexOf("F") < 0

        && TotalCommand.indexOf("B") < 0

        && TotalCommand.length < 2) {

        TotalCommand += "F";

    }

    else if (commandText == "B"

        && TotalCommand.indexOf("B") < 0

        && TotalCommand.indexOf("F") < 0

        && TotalCommand.length < 2) {

        TotalCommand += "B";

    }

    else if (commandText == "R"

        && TotalCommand.indexOf("R") < 0

        && TotalCommand.indexOf("L") < 0

        && TotalCommand.length < 2) {

        TotalCommand += "R";

    }

    else if (commandText == "L"

```

```

    && TotalCommand.indexOf("L") < 0
    && TotalCommand.indexOf("R") < 0
    && TotalCommand.length < 2) {
        TotalCommand += "L";
    }
};

var RemoveCommand = function (commandText) {
    TotalCommand = TotalCommand.replace(commandText, "");
    if (TotalCommand == "") {
        CallControlStop();
    }
};

$(document).ready(function () {
    $(document).keyup(function (e) {

        if (e.keyCode == 38) { //forward
            RemoveCommand("F");
            CallSmartControl(TotalCommand);
        }

        else if (e.keyCode == 40)//Back
        {

```



```
        RemoveCommand("B");

        CallSmartControl(TotalCommand);

    }

    else if (e.keyCode == 37) { //Left

        RemoveCommand("L");

        CallSmartControl(TotalCommand);

    }

    else if (e.keyCode == 39) { //Right

        RemoveCommand("R");

        CallSmartControl(TotalCommand);

    }

    $("#displayCommand").text(TotalCommand);

    });

    });

</script>

</form>

</body>

</html>
```

ControlPage.aspx de yazdığımız kodlar basılan tuşları ve tuş kombinasyonlarını algılar. Tuş kombinasyonlarından dolayı doğacak hataları kontrol eder. Örneğin, RC Car ileri giderken ileri tuşundan elimizi kaldırmadan geri tuşuna bastığımız zaman araç ileriye doğru gitmeye devam eder. Ya da sağ tuşuna basılı iken sol tuşa bastığımız zaman araç mevcut hareketini bozamaz.

Kullanacağımız web uygulamasının tasarımını da ControlPage.aspx sayfasında yapıyoruz. Butonlar ve butonların üstlenmiş olduğu komutlar bu sayfada mevcuttur.

Uygulamamızın tasarımı için kullanmış olduğum StyleSheet.css sayfasının kodları aşağıdaki gibidir.

**html, body, div, span, applet, object, iframe,**

**h1, h2, h3, h4, h5, h6, p, blockquote, pre,**

**a, abbr, acronym, address, big, cite, code,**

**del, dfn, em, font, img, ins, kbd, q, s, samp,**

**small, strike, strong, sub, sup, tt, var,**

**b, u, i, center,**

**dl, dt, dd, ol, ul, li,**

**fieldset, form, label, legend,**

**table, caption, tbody, tfoot, thead, tr, th, td {**

**margin: 0;**

**padding: 0;**

**border: 0;**

**outline: 0;**

**font-size: 100%;**

**vertical-align: baseline;**

**background: transparent;**

**}**

```
body {  
  
    line-height: 1;  
  
    background: url("../Img/bg.jpg") no-repeat center center fixed;  
  
    -moz-background-size: cover;  
  
    -webkit-background-size: cover;  
  
    -o-background-size: cover;  
  
    background-size: cover;  
  
}  
  
ol, ul {  
  
    list-style: none;  
  
}  
  
blockquote, q {  
  
    quotes: none;  
  
}  
  
blockquote:before, blockquote:after,  
q:before, q:after {  
  
    content: " "  
  
    background-color: black;  
  
}  
  
:focus {  
  
    outline: 0;  
  
}  
  
ins {
```

```
text-decoration: none;
}
del {
text-decoration: line-through;
}
table {
border-collapse: collapse;
border-spacing: 0;
}

.main {
width:40%;
height:40%;
padding-top:5%;
margin-left:30%;
margin-top:10%;

background: url("../Img/mainbg.png") no-repeat center center fixed;
-moz-background-size: cover;
-webkit-background-size: cover;
-o-background-size: cover;
background-size: cover;
position:absolute;
```

```
-webkit-border-radius: 40px 10px;  
-moz-border-radius: 40px/10px;  
border-radius: 40px/10px;  
  
}
```

```
.forward {  
float:left;  
width:100%;  
margin-left:40%;  
  
}
```

```
.backward {  
float:left;  
width:100%;  
margin-left:40%;  
margin-top:7%;  
  
}
```

```
.left {  
float:left;  
width:auto;  
margin-left:18%;  
margin-top:5%;
```

```
}
```

```
.right {
```

```
float:left;
```

```
width:auto;
```

```
margin-left:8%;
```

```
margin-top:5%;
```

```
}
```

```
.stop {
```

```
float:left;
```

```
width:auto;
```

```
margin-top:5%;
```

```
margin-left:7%;
```

```
}
```



## SONUÇ

Bu projede Arduino Uno modeli ile bir RC Car tasarımı yaptık. Arduino 'nun çalışma mantığını ve çevre birimleri ile iletişim algoritmalarını gördük.

Bu projenin Arduino uno ile geliştirmek istediğim diğer projeler için yardımcı olacağını düşünüyorum.

Umarım bu tez Arduino ile proje geliştirmek isteyen insanlar için bir kaynak olabilir.



## REFERANSLAR

1. <http://www.instructables.com/id/Internet-Controlled-RC-Car/>
2. [http://en.wikipedia.org/wiki/Remote\\_control](http://en.wikipedia.org/wiki/Remote_control)
3. [http://inventors.about.com/od/rstartinventions/a/remote\\_control.htm](http://inventors.about.com/od/rstartinventions/a/remote_control.htm)
4. <http://science.howstuffworks.com/innovation/repurposed-inventions/history-of-remote-control.htm>
5. <http://cyberneticzoo.com/not-quite-robots/1911-20-chess-playing-machines-leonardo-torres-y-quevedo-spanish/>
6. [http://cnrgzgz.com/arduino-nedir/http://en.wikipedia.org/wiki/Sun\\_Microsystems](http://cnrgzgz.com/arduino-nedir/http://en.wikipedia.org/wiki/Sun_Microsystems)
7. <http://www.arduino.cc/en/Main/ArduinoBoardUno>
8. [http://www.arduino.cc/en/uploads/Main/Arduino\\_Uno\\_Rev3-schematic.pdf](http://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf)
9. <http://www.robotistan.com/Arduino-Ethernet-Shield-Wiznet-W5100-Klon,PR-1793.html>
10. <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Dev/Arduino/Boards/arduino-ethernet-schematic.pdf>
11. <http://www.robotistan.com/L298-Cift-Motor-Surucu-Shield-Dual-Motor-Driver-Yesil-PCB,PR-3051.html>
12. [http://www.datasheetcatalog.com/datasheets\\_pdf/L/2/9/8/L298N.shtml](http://www.datasheetcatalog.com/datasheets_pdf/L/2/9/8/L298N.shtml)
13. [http://www.kliksa.com/urun/tp-link-tl-mr3020-portatif-3g4g-kablosuz-n-router/900379085?gclid=CPXnh\\_XSycUCFePItAod9R8AbQ](http://www.kliksa.com/urun/tp-link-tl-mr3020-portatif-3g4g-kablosuz-n-router/900379085?gclid=CPXnh_XSycUCFePItAod9R8AbQ)
14. <http://www.merterelektronik.com/Mustang-12-Volt-13-Amper-Aku,PR-25324.html>
15. <http://www.sahibinden.com/ilan/alisveris-teknik-elektronik-pil-sarj-cihaz-ataba-at-660-6v-12v-500-mah-aku-sarj-cihaz-adapter-150791819/detay>
16. <http://www.arduino.cc/en/Main/Software>
17. <http://www.w3.org/People/Raggett/book4/ch02.html>

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

**İsim Soyisim:** Özgür BEKTAŞ  
**Doğum Tarihi:** 31.08.1989  
**Doğduğu Şehir:** İstanbul  
**Uyruğu:** Türkiye Cumhuriyeti  
**Ehliyet:** Evet (B Sınıfı)  
**Evlilik Durumu:** Bekar  
**Cep Telefonu:** 0506 270 48 01  
**Mail:** ozgurbektas@outlook.com.tr



### EĞİTİM

**İstanbul Aydın Üniversitesi,** Mühendislik Fakültesi, Yazılım Mühendisliği , 2011 - ..., GNO: 2.10 (Mevcut GNO)

**İstanbul Aydın Üniversitesi,** Meslek Yüksek Okulu, Bilgisayar Programcılığı, Mezun, 2009 - 2011, GNO: 2.80

**Lise:** Bahçelievler Lisesi, Fen-Matematik, 2002 – 2006

### İŞ DENEYİMLERİ VE STAJLAR :

**Tam Zamanlı,** Flexsoft, Web Developers, Öğrenilen Beceriler: Basic, C++, Visual Basic, 01.06.2008 - 01.09.2010

**Tam Zamanlı,** Simternet, Web Developers, Öğrenilen Beceriler: ASP.NET, MS-SQL, 01.06.2011 - 01.09.2012

**Yarı Zamanlı,** Pfizer TURKEY, Organization Team, 14.11.2012 – 01.06.2013

**Tam Zamanlı,** Flexible Graphic, Web Developers, Öğrenilen Beceriler: MVC, MS-SQL, 01.06.2013 - Still Working.

### YABANCI DİL

**İngilizce:** Advanced in Reading, Writing, Speaking  
İngilizce Hazırlık Okulu – İstanbul Aydın Üniversitesi (2011-2012)

