



**T.C.**  
**KARADENİZ TEKNİK ÜNİVERSİTESİ**  
Mühendislik Fakültesi

Elektrik-Elektronik Mühendisliği Bölümü

# ARDUINO İLE EV GÜVENLİK SİSTEMLERİ

220714 Sevcan Rabia ÇİRİŞ  
228519 Yiğitcan KOYUNCU  
228595 Semih ORHAN

## **Danışman**

Yrd.Doç.Dr Gökçe HACIOĞLU

Mayıs, 2013  
TRABZON



**T.C.**  
**KARADENİZ TEKNİK ÜNİVERSİTESİ**  
Mühendislik Fakültesi

Elektrik-Elektronik Mühendisliği Bölümü

# ARDUINO İLE EV GÜVENLİK SİSTEMLERİ

220714 Sevcan Rabia ÇİRİŞ  
228519 Yiğitcan KOYUNCU  
228595 Semih ORHAN

**Danışman**

Yrd.Doç.Dr Gökçe HACIOĞLU

Mayıs, 2013  
TRABZON

## LİSANS BİTİRME PROJESİ ONAY FORMU

228595 Semih Orhan, 220714 Sevcan Rabia ÇİRİŞ, 228519 Yiğitcan KOYUNCU tarafından, Yrd. Doç. Dr. Gökçe HACIOĞLU yönetiminde hazırlanan “Arduino İle Ev Güvenlik Sistemleri” başlıklı lisans bitirme projesi tarafımızdan incelenmiş, kapsamı ve niteliği açısından bir Lisans Bitirme Projesi olarak kabul edilmiştir.

.....

---

Yrd. Doç. Dr. Gökçe HACIOĞLU  
Danışman

.....

Prof. Dr. Temel KAYIKÇIOĞLU  
Jüri Üyesi

.....

Doç. Dr. Ali GANGAL  
Jüri Üyesi

---

Prof. Dr. İsmail H. ALTAŞ  
Bölüm Başkanı

## ÖNSÖZ

Bu kılavuzun ilk taslaklarının hazırlanmasında emeđi geçenlere, kılavuzun son halini almasında yol gösterici olan kıymetli hocamız Sayın Yrd. Doç. Dr. Gökçe HACIOĐLU'na Őükranlarımızı sunmak istiyoruz. Ayrıca bu çalışmayı destekleyen Karadeniz Teknik Üniversitesi Rektörlüğü'ne, Mühendislik Fakültesi Dekanlığına ve Elektrik-Elektronik Mühendisliđi Bölüm Başkanlığına içten teşekkürlerimizi sunarız.

Her Őeyden öte, eğitimimiz süresince bize her konuda tam destek veren ailelerimize ve bize hayatlarıyla örnek olan tüm hocalarımıza saygı ve sevgilerimizi sunarız.

Mayıs, 2013

228595 Semih ORHAN  
220714 Sevcan Rabia ÇİRİŐ  
228519 Yiđitcan KOYUNCU

## İÇİNDEKİLER

LİSANS BİTİRME PROJESİ ONAY FORMU .....	III
ÖNSÖZ .....	IV
ÖZET .....	VI
SEMBOLLER VE KISALTMALAR .....	VII
ŞEKİLLER DİZİNİ .....	VIII
TABLolar DİZİNİ .....	IX
1. GİRİŞ .....	1
2. SİSTEMDE KULLANILAN DEVRE ELEMANLARININ TANITILMASI.....	2
2.1. Kontrol Kartı – Arduino UNO R3 .....	2
2.2. Arduino Ethernet Shield.....	4
2.3 Sensörler.....	5
2.3.1 Sıcaklık Sensörü (LM35) .....	5
2.3.2 Manyetik Sensör (Reed Switch).....	6
2.3.3 Hareket Sensörü ( PIR Motion Sensor ).....	6
2.3.4 Gaz Sensörü (MQ-6) .....	7
2.3.5. Akım Sensörü (SCT-013-000) .....	9
3. SENSÖRLERİN ÇALIŞMA PRENSİPLERİ .....	10
3.1 Sıcaklı Sensörü (LM 35) .....	10
3.2 Gaz Sensörü (MQ-6) .....	12
4. SİSTEMİN ÇALIŞMA MANTIĞI VE UYGULAMALARI.....	13
4.1. Sensör Datalarının Teorik Olarak İnternet Ortamına Aktarılması.....	13
4.2. HTML Kodlarının Tanıtılması.....	14
4.3 Sd Card Web Server.....	15
4.4. Web Server Led Kontrolleri.....	19
4.5 Sistemin Pratik Uygulaması .....	24
5.SONUÇLAR VE ÖNERİLER:.....	34
6.ÇALIŞMALAR .....	35
6.1 İş Zaman Çizelgesi.....	35
7.MALZEME LİSTESİ .....	36
KAYNAKLAR .....	37

## ÖZET

Arduino İle Ev Güvenlik Sistemleri projesindeki amaç; çeşitli sensörler aracılığıyla alınan analog bilgileri Arduino UNO R3 mikrodenetleyicisine aktarmaktır. Bu mikrodenetleyici sayesinde analog bilgiler dijital verilere çevrilir. Ethernet shield aracılığı ile veriler modem üzerinden internet ortamına aktarılır. İnteraktif bir ortamda kontrol imkânı sağlanmış olur.

Projede kullanılan sensörlerden biri gaz sensörüdür. Bu sensör sayesinde evde gaz kaçağı varsa, bu durum interaktif ortamdan tespit edilerek gaz kesimi sağlanır. Bir diğer sensör ise sıcaklık sensörüdür. Yangına karşı önlem için kullanılan bu sensör evdeki olası bir yangın durumunda, yangına daha erken müdahale etme imkanı sağlar. Kapılara ve pencerelere yerleştirilen manyetik sensörler aracılığıyla, kapı ve pencere durumları kontrol edilir ve kaydedilir. Sensörlerden bir diğeri hareket sensörüdür. Evin çeşitli yerlerine veya bahçeye yerleştirilen PIR dedektörü hareketleri algılar ve interaktif ortama aktarır.

İnteraktif erişim ortamı sağlandığı takdirde bu sistemi kullanan kişi, evinin güvenlik durumunu denetleme imkanına sahip olacaktır. Zamanında müdahale etme imkânına sahip olduğu için can ve mal güvenliğinin azami miktarda korunması sağlanacaktır.

## SEMBOLLER VE KISALTMALAR

- MySQL My Sequel
- PHP Personal Home Page
- PCB Printed Circuit Board
- SDRAM Random Access Memory
- EEPROM Electrically Erasable Programmable Read Only Memory
- UDP User Datagram Protocol
- TCP Transmission Control Protocol
- SPI Serial Peripheral Interface
- ADC Analog to Digital Converter
- PPM Parts Per Million
- SD Secure Digital

## ŞEKİLLER DİZİNİ

Şekil 1. Arduino Uno R3 .....	2
Şekil 2. Arduino Ethernet Shield .....	4
Şekil 3. Reed Switch .....	6
Şekil 4. MQ-6 Rs/Ro –Ppm Karakteristiği Tablosu .....	8
Şekil 5. SCT-0013 Akım Sensörü.....	9
Şekil 6. Sıcaklık Sensörü Haberleşme Blog Diyagramı .....	10
Şekil 7. LM 35 Proteus Simülasyonu .....	11
Şekil 8. LM 35 Sıcaklık Çıkış Gerilim Değişimi.....	11
Şekil 9. Gaz Sönsürü Haberleşmesi Blog Diyagramı .....	12
Şekil 10. Web Sitesi Haberleşme Blok Diyagramı .....	13
Şekil 11. Forma Tıklandığındaki Ekran Görüntüsü.....	21
Şekil 12. Led2=2 Durumunda Led'e Emir Verme .....	22
Şekil 13. Led'in Yanma Durumu .....	23
Şekil 14. Led'in Sönme Durumu.....	24
Şekil 15. Button ve Form Komutlarıyla Tasarlanan Butonlar .....	27
Şekil 16. Hareket Algılanmadığı Taktirde Ekran Görüntüsü .....	28
Şekil 17. Oda Sıcaklığı Ekran Görüntüsü .....	28
Şekil 18. Kapı Açık Ekran Görüntüsü .....	29
Şekil 19. Kapı Kapalı Ekran Görüntüsü .....	29
Şekil 20. Gaz Kaçağı Var Ekran Görüntüsü .....	30
Şekil 21. Oda Gaz Yoğunluğu Durumu Normal Ekran Görüntüsü .....	30
Şekil 22. Arduino ile Ev Güvenlik Sistemlerinin Web Tabanlı Kullanıcı Arayüzü Başlangıç Sayfası Ekran Görüntüsü .....	32
Şekil 23. Arduino ile Ev Güvenlik Sistemleri Devresinin Üst Görünüşü .....	33



## **TABLolar DİZİNİ**

Tablo 1. Atmega 328 Mikrodenetleyicisinin Özellikleri Tablosu .....	3
Tablo 2. İş Zaman Çizelgesi Tablosu .....	35

,

## 1. GİRİŞ

Teknolojik gelişmeler hayatı kolaylaştırarak, yaşam verimini arttırmaktadır. Eskiden hayali dahi kurulamayan birçok olgu artık gündelik hayatta yer almaktadır. Son yıllarda internet kullanımı yaygınlaşmasıyla, kullanıcılar internetten erişim ve kontrolün mümkün olduğu sistemlere ilgi göstermektedir. Arduino ile Ev Güvenlik Sistemleri projesinin yapılma amacı; piyasadaki mevcut ev güvenlik sistemlerinin çalışma mantığının anlaşılması ve geliştirilmesine katkı sağlanmasıdır.

Bu projede sensörlerden alınan veriler, Arduino UNO R3 Mikrodenetleyicisi kullanılarak, modem üzerinden internet ortamına aktarılmaktadır. Bu işlem yapılırken mikroişlemcinin bağlı olduğu bir bilgisayara ihtiyaç duyulmamaktadır. Mikroişlemci içinde tasarlanan web sayfası ile modem ilişkilendirilerek, internet ortamına veriler aktarılmaktadır. Bu sistem ile evin güvenlik durumunu arayüz üzerinden takip etmek mümkündür. Mikroişlemci üzerinde sürekli olarak yazma ve okuma istenmektedir. Bu sayede evin mevcut durumunun an be an takip edilmesine olanak sağlamaktadır. Gerektiği takdirde, yazma özelliği ile web sayfasını kullanarak, evin elektriğini kesmek, ışıkları yakıp söndürmek gibi uzaktan kontrol özelliklerine erişilebilmektedir. Bu sayede can ve mal kaybını minimuma indirmek amaçlanmaktadır.

Projede kullanılan, Arduino UNO R3' ün seçilmesinin amacı:

- Diğer mikrodenetleyicilere göre daha geniş kütüphane desteğinin olması,
- Birlikte çalışılabilecek birçok shield bulunması,
- Hemen hemen her sensör tipini desteklemesi,
- Açık kaynak kodlu olması,
- Diğer mikroişlemcilere göre en büyük avantajı; *internet üzerinden veri takibinde, mikroişlemcinin bağlı olması gereken herhangi bir bilgisayar gereksinimine ihtiyaç duymamasıdır.*

Projede web sayfası HTML kodları, mikroişlemci içine yazılarak tasarlandı. Modem üzerinden erişim için; Arduino Ethernet Shield'in Web Server özelliği kullanılmaktadır. Bu sayede herhangi bir bilgisayara ihtiyaç duyulmadan RJ-45 üzerinden modem aracılığıyla internet ortamına data aktarılabilir .

## 2. SİSTEMDE KULLANILAN DEVRE ELEMANLARININ TANITILMASI

### 2.1. Kontrol Kartı – Arduino UNO R3

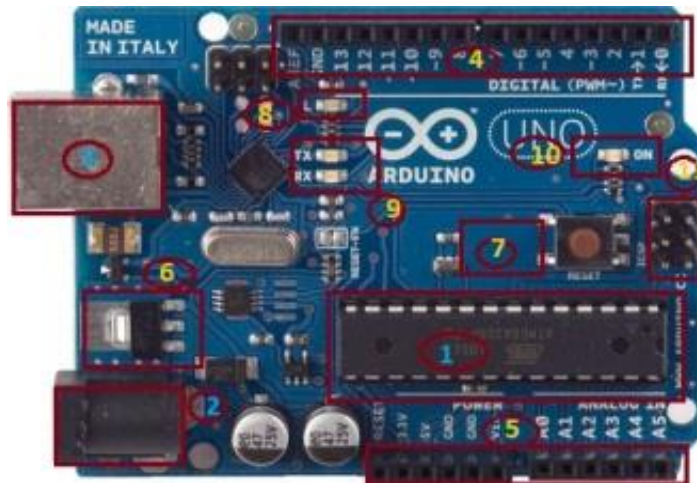
Mikrodenetleyici programlanabilen, verilerin bir program içerisinde depolanıp daha sonra geri çağırılabilirdiği bir chip'ten oluşan elektronik alettir.

Projede kullanılacak olan mikrodenetleyici seçilirken programlanabilme kolaylığı ve kaynak bakımından zenginliği dikkate alınmıştır. Mikrodenetleyicilerin iç yapıları oldukça karmaşık olduğu için daha basit bir yazılım dili aracılığıyla daha verimli bir şekilde kullanmak mümkün olacaktır. Arduino UNO R3 mikrodenetleyiciside bu özelliklerin hepsini barındırmaktadır.

Şekil.1'de görülen Arduino Uno R3, İtalyan mühendislerin geliştirmiş olduğu açık kaynak yazılımlı ve donanımlı bir mikrodenetleyicidir. Açık kaynak donanımı; bir donanıma ait PCB, devre şeması vb. bilgilerin verilmesi ve o donanım üzerinde istenilen değişikliklerin yapılması, üretme hakkını kapsamına anlamına gelmektedir. Açık kaynak yazılımı ise oluşturulan projelerin ve kodların internet ortamında paylaşılmasıdır. Bu sayede önceden yapılmış projeler ve örnekler incelenerek, geliştirilmesine daha çok katkı sağlamaktadır. Kodların paylaşımı yazan kişinin izni doğrultusunda gerçekleştirilebilir. O kodları yazan kişi izni doğrultusunda kodlardaki algoritmalar kullanılabilir, hatta seri üretim dahi yapılabilir. Açık kaynak kodlu ve donanımlı olması büyük bir topluluk desteği sağlamaktadır. Bu sayede geliştirilecek projelerin mevcut örneklerinin incelenebilmesi, kütüphanelerinin kullanılması, projede sorun teşkil edilen bölümlerin çözümünde kolaylık sağlamaktadır.

Arduino İle Ev Güvenlik Sistemleri projesinde Arduino UNO R3 ve Arduino Ethernet Shield kullanılacaktır. [1]

Arduino UNO R3 devre modülü;



Şekil 1. Arduino Uno R3

Kart üzerindeki devre elemanların tanıtımı:

**1)Atmega 328 Mikrodenetleyicisi:** Sensör verilerinin aktarıldığı, bilgisayarda yazılan kodların saklandığı, işlemlerin yapıldığı yani modülün beyni olarak görülebilecek devre elemanıdır. Herhangi bir arızada kolaylıkla değiştirilebilir.

Arduino Uno modülünde Atmega 328 mikrodenetleyicisi 32 KB flash belleğe sahiptir. Flash bellek , üzerine yazılan verilerin silinebildiği ve tekrar yeni verilerin yazılmasına imkan sağlayan bellek tipidir. Tablo 1’de Atmega 328 mikrodenetleyicisinin özellikleri bulunmaktadır.

Tablo 1. Atmega 328 Mikrodenetleyicisinin Özellikleri Tablosu

Dijital Giriş / Çıkış	14 ( 6 tanesi PWM)
Analoj Giriş	6
Her bir G/Ç pini başına akım	40mA
3.3V gerilim çıkışı akımı maksimum	50ma
Flask Bellek	32KB
SRAM	2KB
EEPROM	1KB
Saat Hızı	16 MHz
Bootloader tarafın kullanılan bellek	2KB

**2) 9-12V güç girişi:** Arduino modülünü beslemek için kullanılır. USB bağlantısı olmadan da kart bu girişten beslenebilir.

**3) USB Bağlantı Konektörü:** Arduino üzerinde bulunan mikro denetleyiciye kod yazmak için kullanılan haberleşme birimidir. Aynı zamanda besleme görevi de görmektedir.

**4) Dijital Giriş-Çıkış pinleri:** Dijital giriş çıkış pinlerinin bulunduğu bölümdür. Ayrıca ( ~ ) işareti bulunan pinler analog çıkış (PWM) almak için de kullanılabilir.

**5) Analog giriş Pinleri:** Bu projede kullanılacak analog girişler, gerilim bağlantıları bu bölümde bulunmaktadır.

**6) Güç Regülâtörü:** İşlemci ve diğer birimlerin beslenmesi için gerekli gerilimi üretir.

**7) Reset Butonu:** Arduino’yu yeniden başlatır. RAM bellek üzerindeki veriler silinir.

**8) Led:** Projelerde pin 13 e değer atayarak, programın çalışır olup olmadığı o Led’den takip edilebilir.

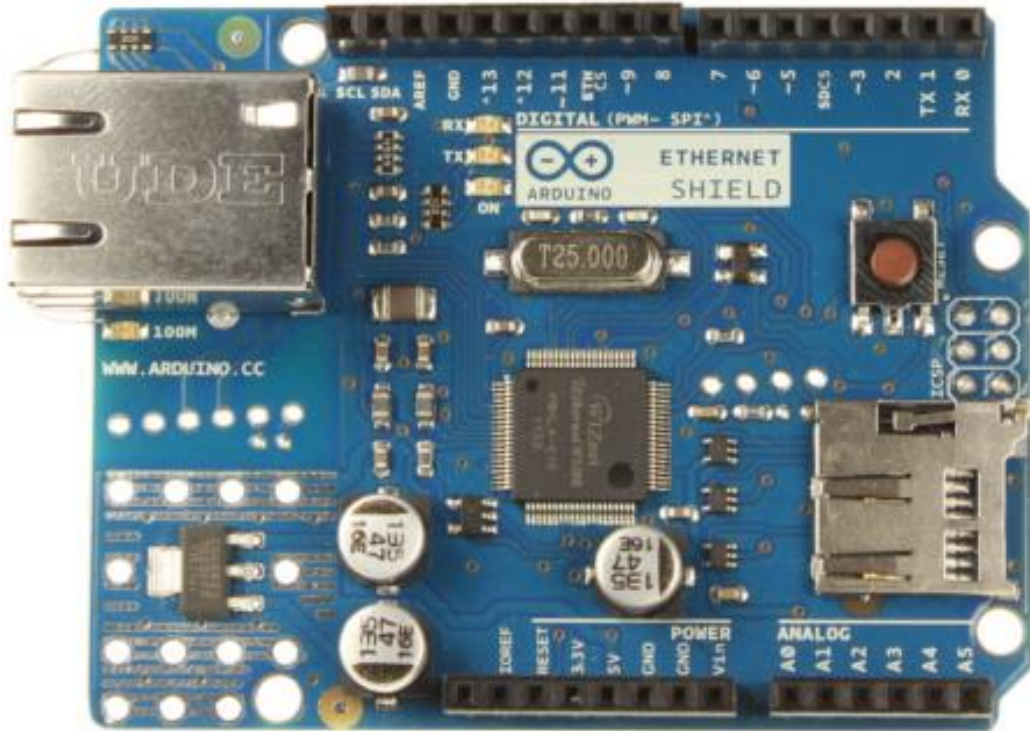
9) Seril İletişim (RX) , çıkış (TX) Led'leri : Seri haberleşme için kullanılan RX ve TX pinlerinin durumunu gösterir. Haberleşme olduğunda bu Led'ler yanacaktır.

10) Güç Led'i: Kartımızın güç gösterge Led'idir. Devremizde bir kısa devre olup olmadığını bu Led'e bakarak görebiliriz.

11) ICSP pinleri: Atmega 328 mikrodenetleyicisini harici bir programlayıcı ile programlayabiliriz. Bu işlem için ICSP pinleri kullanılır. [6]

## 2.2. Arduino Ethernet Shield

Şekil.2'de Projede kullanılan Arduino ETHERNET SHIELD görülmektedir.



Şekil 2. Arduino Ethernet Shield

Arduino ETHERNET SHIELD internet bağlantılı projeler yapabilmek için üzerinde ethernet çipi ve Ethernet portu bulunmaktadır. Bu ethernet çipi Wiznet W5100'dur.

Wiznet W5100 ağ(IP), TCP ve UDP protokollerine sahiptir.b Atmega 328 mikro denetleyicisine sahip Ethernet shield, SD kartı yuvasını da bünyesinde barındırmaktadır. Üzerinde standart bir RJ-45 soketini barındırmaktadır.

Arduino UNO R3, hem W5100 hem SD kart ile, SPI( ICSP) aracılığıyla haberleşir. Dijital sinyal pinleri 11, 12, 13'tür. Pin 10, W5100'u seçmek için kullanılır ve 4 pin de SD kart için ayrılır. Bu pinler giriş/çıkış için kullanılmaz. Mikrodenetleyici içine yazılan HTML kodları ETHERNET SHIELD arayıcılığıyla RJ-45 soketi üzerinden modeme bağlanmakta ve internet erişimini sağlamaktadır . [5]

## 2.3 Sensörler

Sensörler; o an ki çevre koşullarındaki bilgileri toplamaya yarayan elektriksel aygıtlardır. Sensörlere gelen veriler Transducer'lar aracılığıyla elektrik sinyaline dönüşür. Sensörler analog ve dijital olarak iki grupta sınıflandırılabilir. Analog Sensörler: sensör çıkışından gelen işaretin zaman, genlik ekseninin her yerinde değer alabilen sensörlerdir. Dijital sensörler ise binary taban kullanan, çıkışında 1 ve 0 üreten sensörlerdir.

Projede kullanılan sensörler; LM35 analog sıcaklık sensörü, manyetik sensör olarak Reed Switch, hareket sensörü olarak SE-10 (PIR motion sensör) kullanıldı.

Projede analog sensörlerin seçilme amacı; dijital sensörlere göre kullanımının daha kolay olmasıdır. Bu projede kullanım alanına göre çok hassas ölçümlerin yapılması gerekmemektedir. Bütün bu kriterler göz önüne alındığında dijital ve analog sensörler arasındaki fiyat /performansı oranı analog sensör kullanımının daha makul olduğunu göstermektedir.

Arduino mikroişlemcisi, dijital tabanlı bir donanımdır. Mikroişlemcinin bu verileri algılaması için, sensörlerden gelen analog verilerin dijital veri formuna çevrilmesi gerekir. Bu işlem için Arduino kendi bünyesinde ADC bulundurmaktadır. Bu sayede gelen analog sinyal 10 bitlik kuantalayıcıdan geçerek dijital sinyale dönüştürülmektedir. Örnekleme frekansı ise 10 KHz olarak Arduino kütüphanesinde sabitlenmiştir. Ses ve görüntü işleme gibi işlemlerde yüksek frekanslarda örnekleme, sinyal verimi için dijital sinyale çevirim için önemlidir. İstendiği takdirde, Arduino kütüphanesine girerek örnekleme frekansı artırılabilir veya azaltılabilir. Arduinonun analog giriş portlarını kullanarak maksimum ölçülebilecek olan giriş sinyali 5V'dur. 10 bitlik kuantalayıcının adım aralığını hesaplamak gerekirse,  $2^{10}=1024$  adım bulunur. Kuantalayıcı genişliği adım sayısına bölünerek hesaplanabilir.  $5V \div 1024=0,0048828125V$  şeklinde bulunur.

ADC den çıkan decimal sayı 0,0048828125V ile çarpıldığında ilgili birimin volt değerinden tekrar hesaplanmış olur. Bu neredeyse tüm analog çıkışlı sensör verilerinin değerlendirilmesinde kullanılır.

### 2.3.1 Sıcaklık Sensörü (LM35)

Projede sıcaklık sensörü kullanılma amacı; evin veya herhangi bir ortamın sıcaklığını ölçmektir. Bu sıcaklık değerlerinin, hazırlanan web sayfası üzerinden takibinin mümkün olması istenmektedir. Bu sayede sıcaklık istenmeyen bir seviyeye ulaştığı takdirde, örneğin bir yangın halinde erken müdahale edilmesine olanak sağlamaktır. Erken müdahaleye imkân vermesi halinde can ve mal kaybının en aza indirilmesi hedeflenmektedir.

Projemizde LM35 sıcaklık sensörü kullanılmaktadır. Kısaca LM35 sensörünü tanımlamak gerekirse;

- LM35 analog bir sensördür .
- Her bir 1 °C değişiminde çıkış gerilimi 10mV değişir.
- -55 °C ve 150°C ölçüm aralığına sahiptir.
- Ölçüm hassasiyeti 0.5°C dir [3].

### 2.3.2 Manyetik Sensör (Reed Switch)

Projede manyetik sensörlerin kullanılma amacı; evin, kapı, pencere, gibi açılıp kapanabilen ve güvenlik açısından sorun teşkil edebilecek bölgelerine, metal bir parça ve manyetik sensörler konularak evde olunmadığı durumlarda, evin güvenliği hakkında bilgi edinmektir.

Şekil.3’de görülen Reed Switch digital bir sensördür. Bu sensörün seçilmesindeki amaç piyasadaki manyetik sensörlere göre fiyatının daha makul bir durumda olması ve kullanım kolaylığıdır.



Şekil 3. Reed Switch

Reed Switch anahtarlama devresi gibi çalışmaktadır. Paralel olarak birbirine temas eden iki parça metalden oluşmaktadır. Herhangi bir mıknatıs yaklaştırıldığında bu metal parçalar birbirinden ayrılmakta ve çıkış lojik 0 olmaktadır. Anahtarlama devresi mantığıyla çalışmaktadır. Reed switch çıkışı Arduino’nun dijital pinine bağlanarak, “digitalread” komutu ile Reed Switch’in çıkışı okunmaktadır. Bu işlemlerle evin odasının veya penceresinin açık ya da kapalı olduğu denetlenebilmektedir.

### 2.3.3 Hareket Sensörü ( PIR Motion Sensor )

Hareket sensörünün kullanılmasında ki amaç; evin herhangi bir bölümünde, hareketlenme olup olmadığını denetlemektir.

PIR, Pasif kızılötesi sensör objelerden yansıyan kızılötesi ışınımı ölçen elektronik bir sensördür. En çok hareket algılayıcı detektörlerde kullanılır. Küçük, ucuz, düşük güç tüketen, kullanımı kolay ve yıpranmayan cihazlardır. Bu nedenle ev ve işyerlerinde sıklıkla kullanılmaktadır.

PIR temel olarak; yuvarlak bir metalin üstünde görülen, dikdörtgen bir kristal olan pyroelectric sensörden yapılır. Bu sensörün görevi kızılötesi ışımalarının seviyesini belirler. Her şey az da olsa bir miktar ışımaya yapar. Daha sıcak olan şeyler daha fazla ışımaya yapar. Hareket detektöründeki sensör aslında iki yarıma bölünmüştür. Bu nedenle ortalama kızılötesi seviyesi yerine hareket değişimine bakılır. Bu iki yarımdan biri diğerine bağlı olduğu için birbirlerini dengelerler. Eğer yarımlardan biri diğerinden daha az ya da daha fazla kızılötesi ışımaya görse dahi çıkış az ve çok olmak üzere sadece iki değere sahip olur.

Kızılötesi ışınlar insan gözüyle görülemezler. Sadece bu amaç için tasarlanmış cihazlar yardımıyla algılanabilirler. PIR cihazlarına pasif denmesinin sebebi bu cihazların kızılötesi ışın üretmemesi ve yaymamasından kaynaklanır. Bu cihazlar sadece diğer objelerden gelen ışınları algılayabilme yeteneğine sahiptirler.

#### **2.3.4 Gaz Sensörü (MQ-6)**

Piyasada kullanım alanlarına göre farklı birçok çeşit sensör bulunmaktadır. Bunlara örnek vermek gerekirse; karbonmonoksit sensörleri, metal gazı sensörü, Lpg, izobütan, propan gazı sensörleri vb. birçok sensör bulunmaktadır.

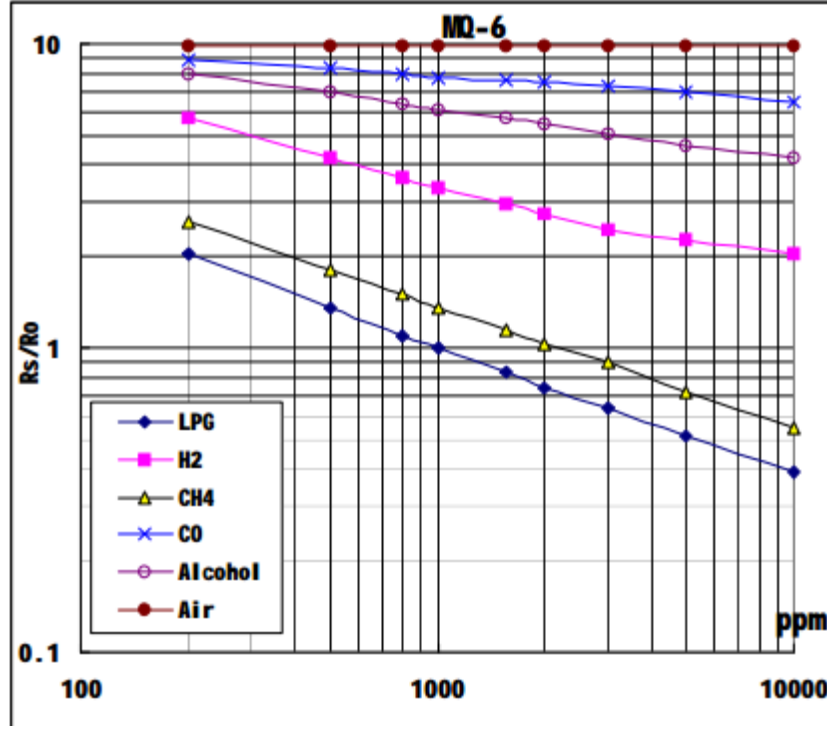
Bu proje ev ortamında gerçekleştirmek üzere tasarlandığı için Lpg, izobütan, propan gazı algılayıcı olan MQ-6 tipi sensör kullanılacaktır.

Gaz sensörü kullanılma amacı; herhangi bir gaz sızıntısı olduğunda can ve mal kaybını en aza indirmektir.

Gaz sensörleri yarı iletken elemanlı çalışma prensibine sahiptir. Yarı iletken malzeme iletken konumdadır, havada bulunan uyarıcı gaz ile etkileşime geçildiğinde elektronlar bağlanır ve iletkenlik azalır. Tekrar temiz havaya çıkarıldığında ise iletkenlik tekrar artmaya başlar.

Temiz ortamdaki iletkenlik / geçirgenlik oranı ile kirli ortamdaki iletkenlik / geçirgenlik oranı karşılaştırılarak ortamın durumu belirlenir.





Şekil 4. MQ-6 Rs/Ro –Ppm Karakteristiği Tablosu

Sparkfun datasheet'ini incelendiğinde sıcaklık 20°C , nem %65 , Oksijen yoğunluğu %21 için  $R_L=20K\Omega$

$R_o$ =Sensör direnci Lpg temiz hava için 1000ppm[

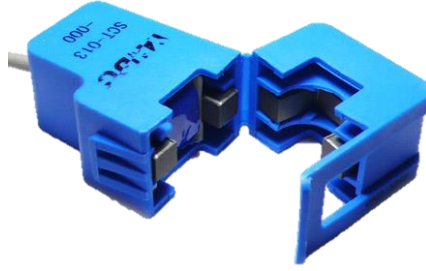
$R_s$ = Hava kirliliğine göre değişen direnç temsil etmektedir.

Şekil.4'te MQ-6 Rs/Ro – Ppm karakteristiği tablosu yer almaktadır. MQ-6 gaz sensörümüz analog çıkışlı bir sensör olduğu için ADC yardımı ile dijital sinyale çevrilecek ve Arduino ile bu veriler bilgisayar ortamına aktarılacaktır. [2]

### 2.3.5. Akım Sensörü (SCT-013-000)

Bir kablodan akım akarken çevresinde gerilim düşümü oluşur. Ayrıca iletkenin çevresinde manyetik alan da oluşur. Akım sensörleri bu devrimler kullanılarak dizayn edilir. Böylece Direk ve dolaylı ölçüm yapan olmak üzere iki tür akım sensörü oluşur. Direk ölçüm yapan sensörler Ohm Yasası'nı baz alırken, dolaylı ölçüm yapan akım sensörleri Faraday ve Amper Yasası'nı baz alır. Direkt ölçüm; pasif devre elemanından akım geçerken oluşan gerilim düşümü ile ilişkilidir. Dolaylı ölçüm ise içinden akım akan iletkeni çevreleyen manyetik alanla ilişkilidir.

Şekil.5'de görülen SCT-0013-000 akım sensörü dolaylı ölçüm yapan, AC gerilimle çalışan bir akım sensörüdür. İletkenin çevresinde oluşan manyetik alandan birinci sargı etkilenerek ikinci sargıda çıkış voltajı endüktler. Çıkış voltajı zener diyotla sınırlandırılır, içinde yük direnci bulundurmaz. Tam yük altında 5V çıkış verebilme yeteneğine sahiptir.



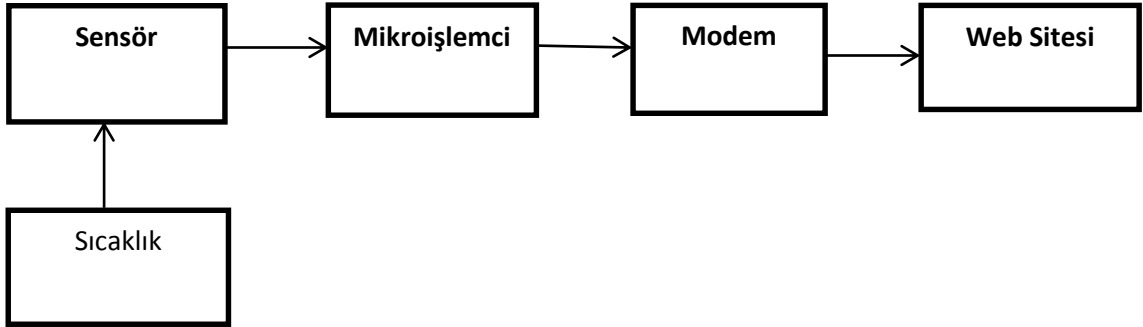
Şekil 5. SCT-0013 Akım Sensörü

### 3. SENSÖRLERİN ÇALIŞMA PRENSİPLERİ

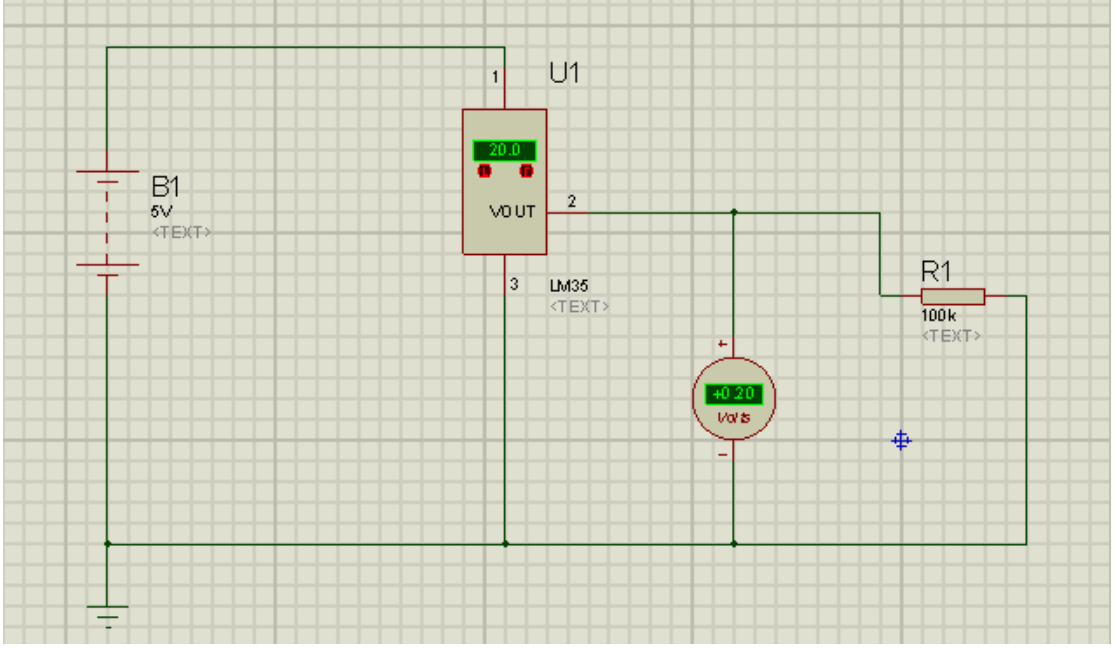
Bu projede sensörden alınan veriler mikroişlemciye iletilmektedir. Mikroişlemciye iletilen veriler dijital sinyale çevrilip Ethernet Shield aracılığı ile RJ-45 soketi üzerinden internet ortamına aktarılmaktadır. Bu sayede uzaktan erişim, internet ortamı ile gerçekleşmektedir.

#### 3.1 Sıcaklı Sensörü (LM 35)

Şekil.6'da sıcaklık sensörü LM35 için hazırlanmış bir blog şema yer almaktadır. LM35 sıcaklık sensörüne etki eden sıcaklıkla gerilim değeri değişmekte ve bu gerilim değeri mikroişlemciye aktarılmaktadır. Mikroişlemciye gelen bu analog veri, ADC ile dijital forma çevrilmekte ve Ethernet Shield aracılığı ile RJ-45 üzerinden modeme gönderilmekte ve internet ortamına aktarılmaktadır. Mikroişlemci içinde tasarlanan web sitesi ile internet ortamında erişimi sağlanmaktadır.

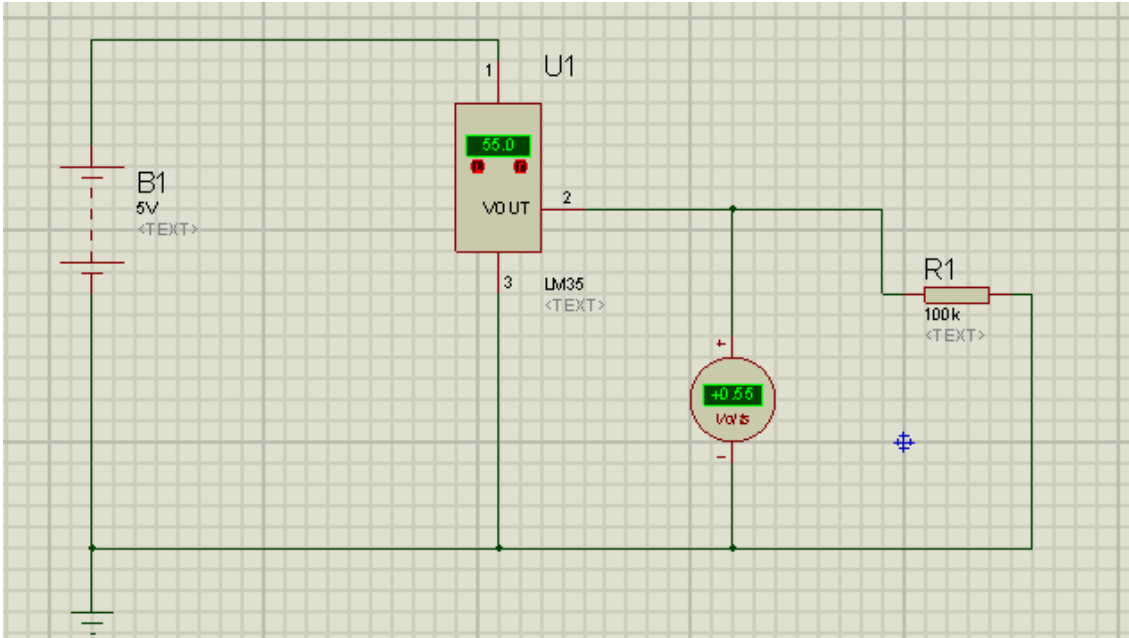


Şekil 6. Sıcaklık Sensörü Haberleşme Blog Diyagramı



Şekil 7. LM 35 Proteus Simülasyonu

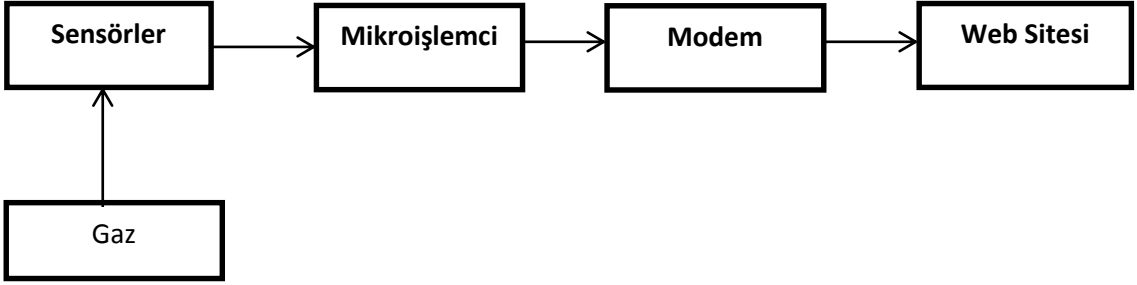
Şekil.7’de LM 35 analog sıcaklık sensörünün Proteus’ta simülasyonu yapılmıştır. Çalışma mantığının anlaşılması amacıyla LM35 datasheet’i incelenerek bağlantısı yapılıp, oda sıcaklığında çıkış gerilimi ölçülmüştür. Şekil.8’de ise LM 35 in sıcaklık değeri artırılıp çıkış gerilim değişimi incelenmiştir.



Şekil 8. LM 35 Sıcaklık Çıkış Gerilim Değişimi

### 3.2 Gaz Sensörü (MQ-6)

Bu projede kullanılan gaz sensörü analog bir gaz sensörüdür. Hava ortamının değişmesiyle gaz sensörünün çıkış gerilimi değişecektir. Bu sinyal mikroişlemciye aktarılacak ve ADC ler yardımı ile dijital (decimal) tabana verilecektir. Dijital veriler usb üzerinden bilgisayar ortamına aktarılacak ve Arduino Ethernet Shield aracılığıyla ethernet üzerinden internete aktarılacaktır.



Şekil 9.Gaz Sönsürü Haberleşmesi Blog Diyagramı

Şekil.9’da Gaz sensörü haberleşmesi blog diyagramı verilmiştir. Projede kullanılan bütün sensörler analog sensör olduğu için blok şemaları ve haberleşme prensipleri birbirine benzemektedir. Diğer sensörlerin simülasyon imkanı zor olması nedeni ve blog şemaların birbirine benzemesi nedeniyle bu blok şemalar yeterli görülmüştür.

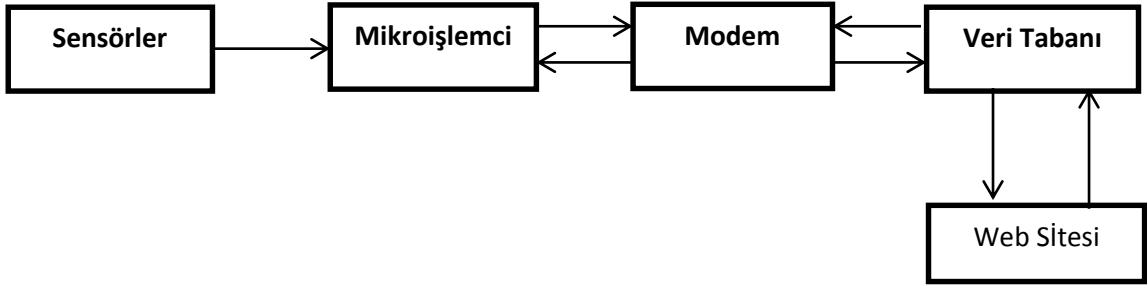
## 4. SİSTEMİN ÇALIŞMA MANTIĞI VE UYGULAMALARI

### 4.1. Sensör Datalarının Teorik Olarak İnternet Ortamına Aktarılması

Mikroişlemciden alınan bilgiler, bilgisayar ortamına aktarıldıktan sonra otomatikman bir veri tabanına kaydedilmesi gerekmektedir. İhtiyaç duyulan veri boyutunun küçük olması nedeniyle Msq1 veri tabanını kullanmakta karar verilmiştir. Msq1 küçük boyutlu verilerde oldukça hızlı çalışmaktadır.

Web sitesi tasarımı PHP kullanılması uygundur. PHP, Msq1 veri tabanı ile birlikte iyi bir performans sergilemektedir. Bugün dünyaca ünlü birçok sitede PHP kullanılmaktadır.

Sensörlerden alınan verilerin sürekli olarak veri tabanından okunması ve yazılması gerekmektedir. Sürekli okuma ve yazma döngüsünde web sitesi sürekli yenilecek şekilde ayarlanıp JAVA yazılım dili ile yapılması uygundur. Sıcaklık değerleri an be an veri tabanına kaydedilecektir. Bu da belirli zaman aralıklarındaki bütün sıcaklık değerlerine erişmeyi mümkün kılar. Şekil.10'da Web sitesi haberleşme blok diyagramı gösterilmektedir.



Şekil 10. Web Sitesi Haberleşme Blok Diyagramı

## 4.2. HTML Kodlarının Tanıtılması

Bu kısımda projede kullanılan HTML kodlarının bir kısmı anlatılmaktadır.

**<html> ,</html>** : Tarayıcının bir html kodu olduğunu algılar ve </html> ile o kodun bittiğini anlaması sağlanır.

**<head> ,</head>** : Bu kod başlık açma ve başlık kapatma kodudur.

**<title> ,</title>** : Başlık bilgilerinin içerildiği kısımdır.

**<body> ,</body>** : Etiketler arasında kalan kısmın tarayıcıda görülecek bölümüdür.

**<h1> ,</h1>** : Başlık boyutlarını ayarlar . h1 en büyük başlık boyutu iken h6 en küçük başlık boyutudur.

**<p>,</p>** : Paragraf komutudur. HTML her paragrafta bir satır atlar.

**<br>,</br>** : Satır atlama komutudur.

**<form>,</form>** : Form elemanlarını içinde bulundurur.

### Örnek Bir HTML Kodu

```
<!DOCTYPE html>

<html>

  <head>

    <title>Arduino Web Sayfası</title>

  </head>

  <body>

    <h1>Merhaba </h1>

    <p>Arduino Web Sunucusu</p>

  </body>

</html>
```

### 4.3 Sd Card Web Server

Bu kısımda SD karda web nasıl kurulacağı ve SD CARD'dan nasıl web servera erişileceği anlatılmaktadır.

```
<!DOCTYPE html>

<html>

  <head>

    <title>Arduino SD Card Web Sayfası </title>

  </head>

  <body>

    <h1>Arduino SD Card dan Merhaba</h1>

    <p>A Arduino Sd Card Web Sayfası.</p>

  </body>

</html>
```

Kodu “Geany” adlı program ile index.htm şeklinde kaydedildi. Sonra SD CARD Arduino Ethernet Shieldde takılarak yazılımı yapılmaya başlandı.

#### **SD CARD'dan WEB SERVER'a Erişim**

```
#include <SPI.h>

#include <Ethernet.h>

#include <SD.h>

// MAC Address Ethernet Shield'in Altında Yazan Numaralar

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```



```

IPAddress ip(196, 168, 1, 4); // IP Adresim

EthernetServer server(80); // Özel Bir Port Açılmadığında Http Standart 80

File webFile;

void setup()

{

  Ethernet.begin(mac, ip); // Ethernet Shield Başlatılıyor

  server.begin(); // Client Dinleniyor

  Serial.begin(9600);

  Serial.println("SD card yukleniyor..."); // Sd Card Yükleniyor

  if (!SD.begin(4)) {

    Serial.println("Hata - SD card yuklenemedi!");

    return;

  }

  Serial.println("Basariyla - SD card yuklendi.");

  // SD Carddan Index.Htm Dosyasının Onayı

  if (!SD.exists("index.htm")) {

    Serial.println("Hata - index.htm dosyasi bulunamadi !");

    return; //

  }

  Serial.println("Basarili - Bulundu index.htm dosyasi.");

}

```

```

void loop()

{

  EthernetClient client = server.available(); // Client'a Erişim

  if (client) { // Client Var Mı?

    boolean currentLineIsBlank = true;

    while (client.connected()) {

      if (client.available()) { // Data Okunma Mümkün

        char c = client.read(); // 1 byte (character) client

        // Clienttan İstek İçin Son Satıra Gelindiğinde \n

        // Bütün Satır İşlemleri Tamamlandığı Zaman

        if (c == '\n' && currentLineIsBlank) {

          // Standart Http Protokolü Baslangıcı

          client.println("HTTP/1.1 200 OK");

          client.println("Content-Type: text/html");

          client.println("Connection: close");

          client.println();

          // Web Sayfasına Gönder

          webFile = SD.open("index.htm"); // Sd Card İçindeki Dosya Açıldı

          if (webFile) {

            while(webFile.available()) {

              client.write(webFile.read()); // Web Sayfası Client'a Gönderildi

```

```
    }

    webFile.close();

}

break;

}

// Bütün Satırlar Alındıktan Sonra Clientin Sonuna \r\n

if (c == '\n') {

    // Text'ten Alınan Son Karakter

    // Sonraki Karakterlerle Yeni Bir Satır Başlatmak İçin

    currentLineIsBlank = true;

}

else if (c != '\r') {

    // Clienttan Karakter Alınmışsa

    currentLineIsBlank = false;

}

} // end if (client.uygunsa())

} // end while (client.bağlandı())

delay(1); // Web Sayfasının Datayı Alması İçin Tanınan Süre

client.stop(); // Bütün Bağlantıyı Kes

} // Eğer Client İsteği Bitmişse Sonlandır

}
```

#### 4.4. Web Server Led Kontroleri

Bu kısımda led'i on/off yapmanın web server üzerinden kontrolü sağlanmıştır. Şekil.11'de Forma tıklanıldığında görülen ekran görüntüsü ve Şekil.12'de Led2=2 durumunda led'e emir verilirken ekrana gelen görüntü yer almaktadır. Şekil.13'te Led'in yandığı ve Şekil.14'te sönmüş olduğu resimler yer almaktadır.

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

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

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

```
IPAddress ip(198, 168, 1, 4); // IP Adresim
```

```
EthernetServer server(80); // Özel Bir Port Açılmadığında Http Standart 80
```

```
String HTTP_req; // Http İsteği
```

```
boolean LED_status = 0; // Ledin Durum , Başlangıcının Off Olması İçin
```

```
void setup()
```

```
{
```

```
  Ethernet.begin(mac, ip); // Ethernet Shield Yükleniyor
```

```
  server.begin(); // Client Dinleniyor
```

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

```
  pinMode(2, OUTPUT); // LED Pin 2
```

```
}
```

```

void loop()

{

    EthernetClient client = server.available(); // Cliente Eriş

    if (client) { // Client Var Mı ?

        boolean currentLineIsBlank = true;

        while (client.connected()) {

            if (client.available()) { //

                char c = client.read(); // read 1 byte (character) from client

                HTTP_req += c; // Her Http İsteğinde 1 Karakter Kaydet

                // Son Satır \n

                // Yada Son Satır Alındığında

                if (c == '\n' && currentLineIsBlank) {

                    // Standart Bir Http Yanıt Başlığı

                    client.println("HTTP/1.1 200 OK");

                    client.println("Content-Type: text/html");

                    client.println("Connection: close");

                    client.println();

                    // Web Sayfasının Gönderilmesi

                    client.println("<!DOCTYPE html>");

                    client.println("<html>");

                    client.println("<head>");

                    client.println("<title>Arduino LED Control</title>");

```

```

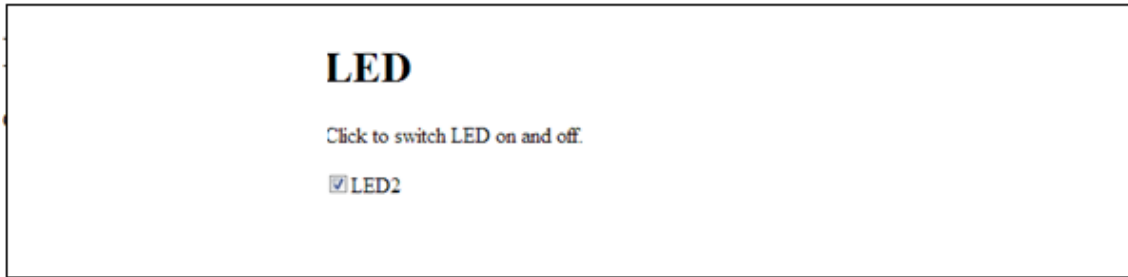
client.println("</head>");

client.println("<body>");

client.println("<h1>LED</h1>");

client.println("<p>Click to switch LED on and off.</p>");

```



Şekil 11. Forma Tıklandığındaki Ekran Görüntüsü

```

client.println("<form method=\"get\">");

ProcessCheckbox(client);

client.println("</form>");

client.println("</body>");

client.println("</html>");

Serial.print(HTTP_req);

HTTP_req = ""; // Talebi Bitir, String Bitince

break;

}

```

// Bütün Satır Ve Dosyalar Alındığı Zaman \r\n

```
if (c == '\n') {
```

```
// Son Karakter Alındığı Zaman
```

```

// Yeni Bir Satır Okunacağında

currentLineIsBlank = true;

}

else if (c != '\r') {

// Bir Karakter Daha Alınırsa

currentLineIsBlank = false;

}

} // Eğer Client Varsa

} // Client'e Bağlı Olduğu Süre Boyunca

delay(1); // Web Sayfasının Dataları Yükleme Zamanı

client.stop(); // Bağlantıyı Kes

} // end if (client)

}

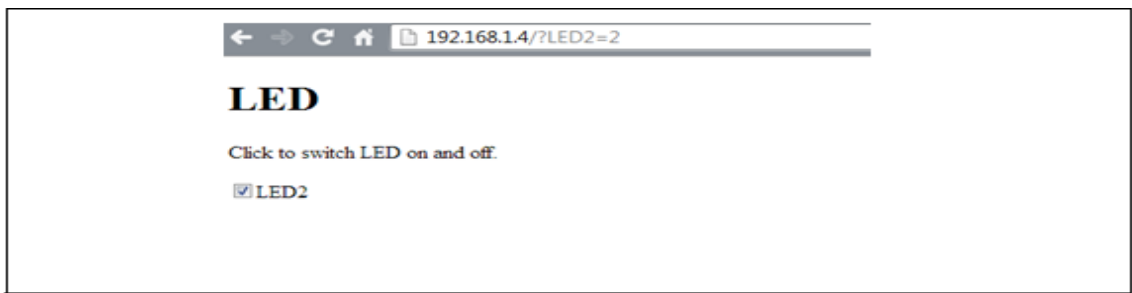
// Led'in Durumunu Değiştir ve Led Checkbox Kısımına Geri Gönder

void ProcessCheckbox(EthernetClient cl)

{

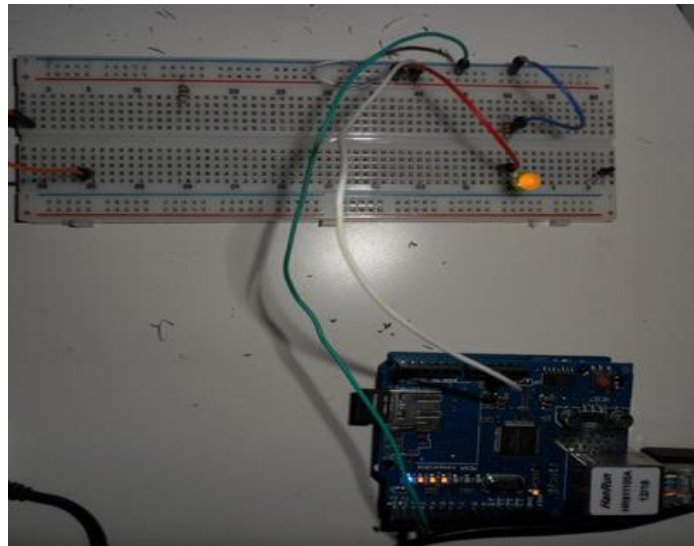
if (HTTP_req.indexOf("LED2=2") > -1) { //

```



Şekil 12. Led2=2 Durumunda Led'e Emir Verme

```
if (LED_status) {  
    LED_status = 0;  
}  
  
else {  
    LED_status = 1;  
}  
  
}  
  
if (LED_status) { // Led i yak  
  
    digitalWrite(2, HIGH);  
  
    // buton kontrol edildi  
  
    cl.println("<input type=\"checkbox\" name=\"LED2\" value=\"2\" \  
    onclick=\"submit();\" checked>LED2"); }  
}
```



Şekil 13. Led'in Yanma Durumu



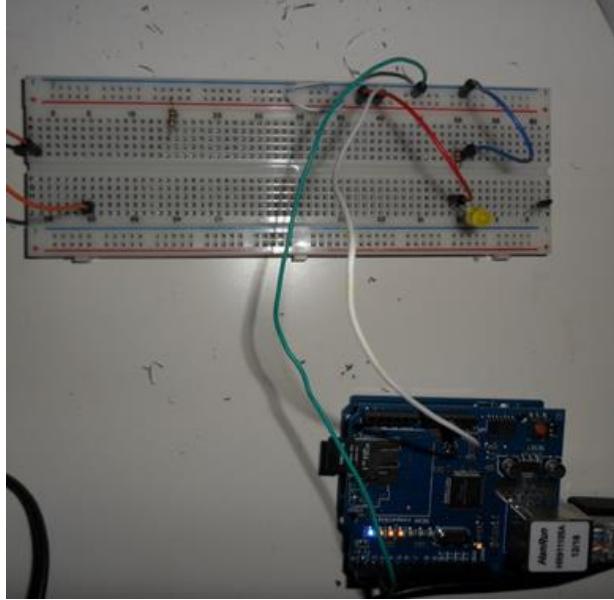
```
else {          //Led i kapat

    digitalWrite(2, LOW);

    cl.println("<input type=\"checkbox\" name=\"LED2\" value=\"2\" \
    onclick=\"submit();\">LED2");

}

}
```



Şekil 14. Led'in Sönme Durumu

#### 4.5 Sistemin Pratik Uygulaması

Arduino Web Server uygulaması geliştirilirken en çok problem olan kısım Led ON/OFF kısmı olmuştur. Dublex bir sensördeki dataları almak için sayfa yenilenmesi gerekmektedir.

Daha önce denenmiş olan Led ON/OFF formülü, sıcaklık sensörü ile birlikte çalışırken hata vermektedir. Sıcaklık, LED kendi kendini ON/OFF yaptığı için başka bir formüle geçilmiştir. Bu formülde sayfa istendiği sıklıkta yenilenmekte ve hiçbir sorun çıkmamaktadır ve ayrıca hareket sensörü PIR ve Reed Switch projeye dahil edilmiş bulunmakta ve oldukça verimli çalışmaktadır.

Şekil.15'te Button ve Form komutlarıyla tasarlanan butonların görüntüleri vardır. Şekil.16'da Hareket algılanmadığı zaman ekrana gelen görüntü, Şekil.17'de Oda sıcaklığı ekran görüntüsü, Şekil 18'de kapının açık olduğu durumdaki ekran görüntüsü, şekil 19'da ise kapının kapalı olduğu durumdaki ekran görüntüsü yer almaktadır. Şekil 20' de gaz sensörü ortamda gaz sızıntısı algıladığında kullanıcı arayüzünde görünen uyarının ekran görüntüsü görülmektedir. Şekil 21'de ise gaz sızıntısı algılanmadığında ekrana gelen görüntü görülmektedir. Şekil 22'de ise Arduino ile ev güvenlik sistemleri kullanıcı arayüzü başlangıç sayfası görülmektedir.

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

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

```
byte mac[] = { 0Xde, 0Xad, 0Xbe, 0Xef, 0Xfe, 0Xed }; //MAC Adresi Ethernet Shieldin
```

```
IPAddress ip(192,168, 1, 4); // Local Ip Adresi
```

```
EthernetServer server(80); // Http Standart Portu
```

```
25oolean25t switchPin = 7;
```

```
25oolean val;
```

```
int gazzsensor;
```

```
String HTTP_req; // Http Isteklerini Atanıyor
```

```
25oolean LED_status = 0; //LED In Durumunu Kontrol Etmek İçin
```

```
int readValue = 0;
```

```
int analogPin = 3;
```

```
float temperature = 0;
```

```
float yy=0;
```

```
int LED=2;
```

```
int PIRstate = 0;
```

```

int PIR = 6;

void setup()
{
  Ethernet.begin(mac, ip); // Ethernet Shield Yükleniyor
  server.begin();        // Client İstedığı Kontrol Ediliyor
  Serial.begin(9600);    // Seri Haberleşme Başlatılıyor
  pinMode(LED, OUTPUT); // Ledpin 2 Den Çıkış Alındı
  pinMode(PIR, INPUT);  // PIR Pin Giriş Olarak Tanımlandı

  pinMode(switchPin, INPUT); // Switchpin Giriş Olarak Tanımlandı
  digitalWrite(switchPin, HIGH); // Başlangıç Değeri Lojik1 Alındı

}

void loop()
{
  EthernetClient client = server.available(); // İsteğe(Client)'e Erişmeye Çalışıyor

  if (client) { // istekvar mı
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) { // Client Datası Okunmaya Hazır
        char c = client.read(); // 1byte Karakter Oku Client'den
        HTTP_req += c; // http_rep eşitliğine kaydet bunu her seferinde

```

```

if (c == '\n' && currentLineIsBlank) {

    client.println("HTTP/1.1 200 OK"); // Http/1.1 200 OK I Gönder
    client.println("Content-Type: text/html");
    client.println("Connection: close");
    client.println();

    // Web Sayfası Kısmı
    client.println("<!DOCTYPE html>");
    client.println("<html>");
    client.println("<head>");
    client.println("<title>Arduino LED Control</title>");
    client.println("</head>");
    client.println("<body>");

    client.println("<center><p><h1>Arduino Ev Otomasyonu
v1.0</h1></p><center><h1><br />");

    client.println("<form method=get name=form>");
    client.println("<button name=LED value=1 type=submit
style=height:80px;width:150px>LED On</button>");

    client.println("<button name=LED value=0 type=submit
style=height:80px;width:150px>LED Off</button>");

```

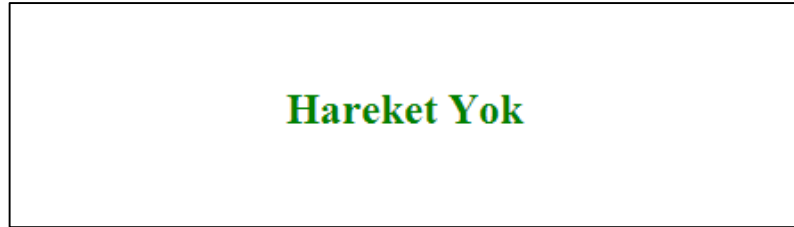


Şekil 15. Button ve Form Komutlarıyla Tasarlanan Butonlar

```

client.println("</form><br />");
PIRstate = digitalRead(6);
if (PIRstate == HIGH) { // PIR sensor hareket algıladı
client.println("<p><h2><font color=red>Hareket Algılandı!</font></h2></p>");
}
else { // Hareket Algılanmadı
client.println("<p><h2><font color=green>Hareket Yok</font></h2></p>");
}

```

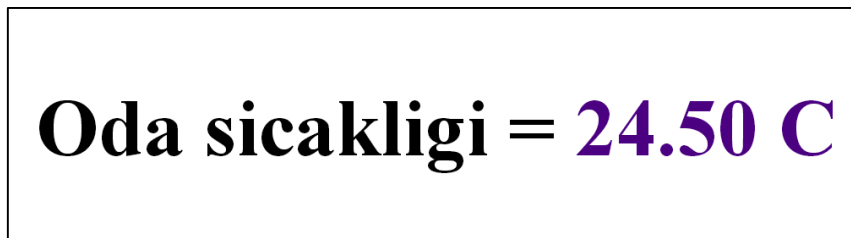


Şekil 16. Hareket Algılanmadığı Taktirde Ekran Görüntüsü

```

client.println("<br /> ");
client.println("<br /> ");
client.println("<p><h2>Oda sıcaklığı = <font color=indigo> ");

```



Şekil 17. Oda Sıcaklığı Ekran Görüntüsü

```

sıcaklık(client);

```

```

client.println("</font><h2></p>"); //Başlık Boyutu H2 Olarak Ayarlandı

```

```
val=digitalRead(7);  
if (val == HIGH) { // Reed Switch Pini High Ise  
client.println(“<p><h2><font color=red>Kapi Acik!</font></h2></p>”);  
}
```



Şekil 18. Kapı Açık Ekran Görüntüsü

```
else { // Mıknatıs Reed Switch'e Yaklaştırıldığında . Kapı Kapalı Mantığı  
client.println(“<p><h2><font color=blue>Kapi Kapali</font></h2></p>”);  
}
```



Şekil 19. Kapı Kapalı Ekran Görüntüsü

```
client.println(“<br /> “);
```

```
client.println(“<br /> “);
```

```
gazsensor = analogRead(4); // Adc Çıkışı Okundu
```

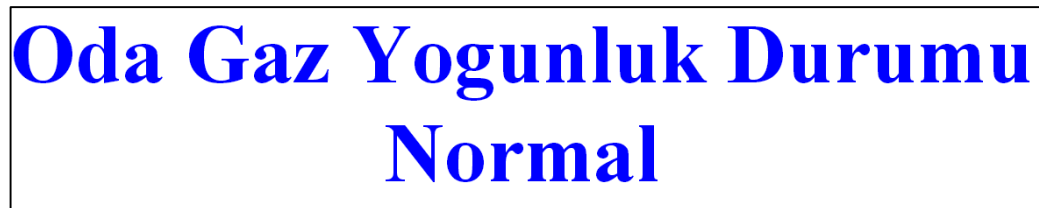
```
if (gazsensor > 800) // Gaz Sensörünün Adc Çıkışı
```

```
{  
client.println("<p><h2><font color=red>Gaz Kacagi Var!</font></h2></p>");  
}
```



Şekil 20. Gaz Kaçağı Var Ekran Görüntüsü

```
else  
{  
client.println("<p><h2><font color=blue>Oda Gaz Yogunluk Durumu Normal  
</font></h2></p>"); }
```



Şekil 21. Oda Gaz Yoğunluğu Durumu Normal Ekran Görüntüsü

```
client.println("<meta http-equiv=\\\"refresh\\\" content=\\\"3\\\">");  
// Her 3 Saniyede Bir Web Sayfasını Yenile. Analog Verilerin Güncellenmesi İçin  
client.println("</body>");  
client.println("</html>");  
Serial.print(HTTP_req);  
HTTP_req = ""; // İstek Bitti  
break;  
}
```

```

    if (c == '\n') {

        currentLineIsBlank = true;
    }

    else if (c != '\r')
    {

        if(HTTP_req.indexOf("GET /?LED=1")>=0)

            digitalWrite(2,HIGH);

        if(HTTP_req.indexOf("GET /?LED=0")>=0)

            digitalWrite(2,LOW);

            currentLineIsBlank = false;
        }
    }

}

delay(1); // Web Sayfasının Datayı Alması İçin 1ms Geçirme Atandı

client.stop(); // Bağlantıyı Kes

}

}

void sicaklik(EthernetClient cl) // Fonksiyon Tanımlandı

{

    float yy=0;

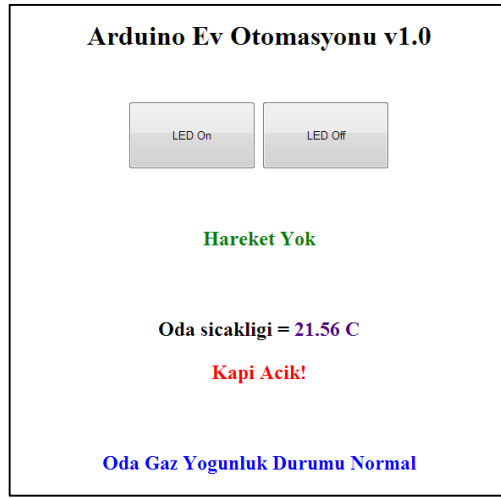
    readValue = analogRead(analogPin); // Analog Data Okundu

    temperature = (readValue * 0.0049); //Arduino ADC İşlemi Yapıldı

```

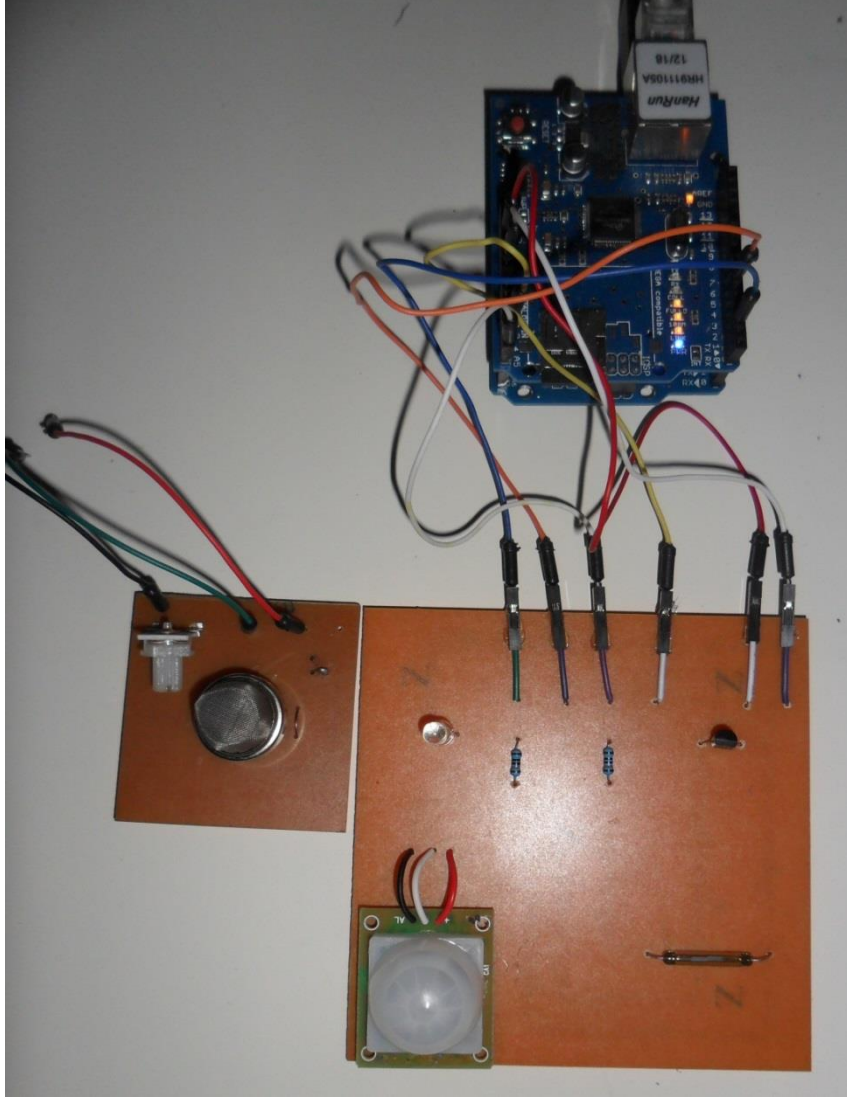


```
temperature = temperature * 100; //Formülüne Edildi  
yy=temperature;  
  
cl.print(yy);  
  
cl.println(" C ");  
}
```



Şekil 22. Arduino ile Ev Güvenlik Sistemlerinin Web Tabanlı Kullanıcı Arayüzü Başlangıç Sayfası Ekran Görüntüsü

Şekil 23'te Arduino ile ev güvenlik sistemleri devresinin üst görünüşü görülmektedir.



Şekil 23. Arduino ile Ev Güvenlik Sistemleri Devresinin Üst Görünüşü

## 5.SONUÇLAR VE ÖNERİLER:

Arduino'yu programlamak için kullanılan yazılım C dili tabanlıdır. Bu çalışmada Arduino'yu kullanmak için C dili kabiliyetimizi geliştirdik. Analog sensörler kullandığımız için sensörlerin çalışma mantığını daha iyi kavradık. Sensörlerden alınan analog verileri dijitale nasıl dönüştürüldüğünü, bu verilerin nasıl okunduğunu öğrendik. Web ortamı kullandığımız için network becerilerimizi geliştirdik. HTML dili hakkında bilgi sahibi olduk. Bu dil sayesinde web ortamına veri aktardık ve web ortamından mikroişlemciyi kontrol edebildik.

Projede yapılabilecek geliştirmeler ise;

Bir adet akım sensörü kullanılarak evin ana hat akımının denetlenilmesi mümkündür.

Basınç sensörü ve Zigbee kullanılarak posta kutusu durumu kontrol edilebilir.

GSM modülü kullanılarak evle ilgili herhangi bir tehlike anında, cep telefonuna bilgilendirme SMS'i gönderilebilir.

## 6.ÇALIŞMALAR

### 6.1 İş Zaman Çizelgesi

Tablo 2 'de projenin hazırlanması için oluşturulan iş zaman çizelgesi yer almaktadır.

Tablo 2. İş Zaman Çizelgesi Tablosu

1.Hafta	Arduino'nun kullanımının öğrenilmesi
2.Hafta	Arduino'nun kullanımının öğrenilmesi
3.Hafta	Arduino'nun kullanımının öğrenilmesi
4.Hafta	Arduino Ethernet Shield'in kullanımının öğrenilmesi
5.Hafta	Arduino Ethernet Shield'in kullanımının öğrenilmesi
6.Hafta	Sıcaklık sensörünün bağlantısı ve çalıştırılması
7.Hafta	Gaz sensörünün bağlantısı ve çalıştırılması
8.Hafta	Hareket sensörünün bağlantısı ve çalıştırılması
9.Hafta	1.Ara Sınavlar
10.Hafta	Sıcaklık sensörünün bağlantısı ve çalıştırılması
11.Hafta	Manyetik sensörün bağlantısı ve çalıştırılması
12.Hafta	Devrenin son halinin gerçekleştirilmesi
13.Hafta	Devrenin son halinin gerçekleştirilmesi
14.Hafta	Son kontroller
15.Hafta	Final Sınavlarına Hazırlık

## 7.MALZEME LİSTESİ

- 1 X Arduino Launchpad
- 1 X Arduino Ethernet Shield
- 2 Led
- 2 Direnç 180 Ohm
- 1 MQ4 Gaz Sensörü
- 1x LM35 Sıcaklık Sensörü
- 1x Reed Switch
- 1x PIR Hareket Sensör Algılayıcısı
- 1 X Atmega 328
- 2 Adet Bakır Plaket
- 10 Adet Bağlantı Kablosu

## KAYNAKLAR

- [1]. M. Riley, *Programming Your Home: Automate with Arduino, Android, and Your Computer (Pragmatic Programmers)*, Amerika,, Haziran 2012
- [2]. “Mq-6 gaz sensörü data sheet”, SparkFun Electronics, United States
- [3]. “Lm 35 temprature sensörü data sheet”, Texas Instruments, Texas, Amerika
- [4]. “SCT-013-000 akım sensörü data sheet”, Beijing YaoHuadechang Electronic Co, Çin
- [5]. Arduino CC web sitesi. Burdan erişim : <http://arduino.cc>
- [6]. C. Taşdemir, *Arduino Analog-Dijital-Sensörler-Haberleşme-Projeler*, Türkiye Dikeyksen Yayıncılık, 2012

## EKLER

### Standartlar Ve Kısıtlar Formu



Karadeniz Teknik Üniversitesi  
Mühendislik Fakültesi  
Elektrik-Elektronik Mühendisliği  
Bölümü

## STANDARTLAR VE KISITLAR FORMU

Tasarım Projesinin hazırlanmasında Standart ve Kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? Açıklayınız.

Projenin tasarım boyutu, bir evin hemen hemen bütün güvenlik açıklarına tedbir alabilmektir. Ev, büro veya küçük çaplı fabrika gibi mekanlara uygulanabilir.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü?

Projede gerek kullanılacak yazılımlarda, gerek sensör haberleşmesindeki değer analizlerinde matematiksel formüller kullanılmıştır.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Bu projede mikroişlemci ve elektronik derslerinde edindiğimiz birçok bilgiyi kullanmış olacağız.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir?

Projemizde TS 9770 standardı kullanılacaktır.

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir?

a) Ekonomi

Projenin üretim maliyeti, gerçek hayata uygulanabilirlik açısından çok önemli bir faktördür. Projede kullanılacak olan elemanların işlevselliği ve fiyatı gözetilerek fiyat/performans elemanları seçilmeye çalışılmıştır

b) Çevre sorunları:

Projenin çevreye herhangi bir zararı olmayacaktır. Düşük enerji tüketimi ile çevre dostu olarak düşünülebilir.

c) Sürdürülebilirlik:

Mikrodenetleyici kontrollü ev güvenlik sistemleri yeni gelişme olan bir sektördür. Mikroişlemci ve sensörlerin gelişmesiyle ilerde çok daha gelişeceği aşikârdır.

d) Üretilebilirlik:

Projede kullanılan elemanların fiyatları oldukça cüzi bir miktardadır. Kullanılabilir fiyat kıyaslamasında, üretilebilirlik öne çıkmaktadır. Seri üretime geçilme imkânı olursa da maliyetler daha da aşağı çekilebilir.

e) Etik:

Mühendislik etik değerlerini gözetebilecektir.

f) Sağlık:

Tasarlanacak projede insan sağlığına kötü yönde etkileyecek herhangi bir şey bulunmamaktadır.

g) Güvenlik:

Web sitesi tasarımında şifreleme arayüzü bulunacaktır. Bu sayede sisteme sadece kullanıcı adı ve şifreyi bilen kullanıcıların erişmesi hedeflenmektedir.

h) Sosyal ve politik sorunlar:

Sosyal ve politik herhangi bir sorun teşkil etmemektedir.