



ARDUINO İLE SENSOR UYGULAMALARI

Nisan, 2015

İÇİNDEKİLER

ARDUİNO İLE SENSOR UYGULAMALARI.....	i
İÇİNDEKİLER.....	ii
ÖZET	iii
ŞEKİLLER	iv
Giriş	1
Avantajları	1
1. Hazır Kütüphaneler	1
2. Hazır modül veya kitler	2
3. Döküman	3
Dezavantajları.....	3
Arduino Arayüzü.....	3
File Sekmesi.....	5
Edit Sekmesi	5
Sketch Sekmesi.....	6
Tools Sekmesi	6
Arduino Modülünü Bilgisayara Tanıtmak	7
Arduino Programlamanın Temel Yapısı	10
Örnek 1. Led Yak-Söndür Uygulaması	12
Örnek 2. Buton İle Led Kontrolü Uygulaması	13
Örnek 3. Analog Çıkış Uygulaması	14
Örnek 4. Analog çıkış uygulaması	16
Örnek 5. Arduino İle PIR Sensörü Kontrolü.....	18
Örnek 6. Arduino İle HC-SRO4 Uzaklık Sensörü Kontrolü.....	20
Örnek 7. Kızılötesi Sensör Uygulaması	24
Örnek 8. DHT11 Sensörü kullanılarak ortamın nem ve sıcaklığının ölçülmesi	26
Örnek 9. Arduino kullanılarak alev sensörünün kullanılması	28
Örnek 10. Sıcaklık sensörü uygulaması	29
Örnek 11. Arduino ile titreşim sensörü uygulaması.....	30
Örnek 12. Engel algılayan sensör	31
Örnek 13. Işığa duyarlı sensör uygulaması	32
Örnek 14. Kızıl ötesi sensör kullanılarak çoklu verinin gönderilmesi	33
Örnek 15. Manyetik alan (Hall etkili) sensör uygulaması	37
Örnek 15. Reed Anahtar kullanılarak manyetik alan sensör uygulaması	38
Örnek 16. DS18820 kullanılarak sıcaklık ölçümü	39
Örnek 17. Nabız Sensörü	40
KAYNAKLAR.....	42
EKLER.....	43
EK-1 – Mevcut Arduino modülleri	43
EK-2. Arduino Mimarisi	47

ÖZET

Arduino açık kaynak donanıma sahip (kopyalanabilir, üretilebilir, satılabilir yada değişiklik yaparak geliştirilebilir) ATMEL firmasının ürettiği mikrodenetleyici tabanlı bir elektronik geliştirme havuzudur.

Arduino havuzu da diğer mikrodenetleyicilerde olduğu gibi C dili ile programlanabilmektedir. C dili yeni başlayanlar için zorlayıcı bir faktör olsa da program ve uygulama geliştirilmesi konusunda kullanıcıya büyük kolaylık sağlayan en verimli (sistem kaynaklarını minimum kullanarak daha az zamanda daha çok iş yapan) bir programa platformudur. Arduino altyapısında da C Dili kullanılmakla beraber uygulama geliştirilmesi oldukça kolaydır.

Arduino modüler bir havuz olmakla beraber Atmel işlemciler kullanılmış bir elektronik karttan ibarettir. Arduino ile uygulama geliştirmek isteyen her kullanıcı kendine ait bir arduino kartı yapabilmekte yada hazır devresini temin edebilmektedir.

Arduino' nun sağladığı faydalardan biride daha az hata ile daha hızlı bir şekilde uygulama geliştirilmesine olanak sağlamaktadır. Birçok defa sınanmış bir kontrol devresi olması nedeni ile bu çalışmada da üzerinde durulan sensör kontrol uygulamalarında yoğun olarak kullanılmaktadır.

Yapılan bu çalışmada arduino tabanlı birçok tipte sensör ile uygulama geliştirilmesi konusu üzerinde durulmuştur. Yapılan örnek uygulamalarda birçok farklı sensörün çalışmasına kısaca değinilmiş ve arduino mimarisinde sensörü çalıştırmak için (yada sensörden gelen bilgiyi okuman için) gereken uygulama kodları uygulamalara eklenmiştir. Uygulamalarda analog giriş- analog çıkış, sensörden gelen dijital verinin alınması, anlamlı hale getirilmesi ve seri iletişim yolu ile gönderilmesi konularına değinilmiştir.

Arduino havuzu ölçme ve kontrol uygulamaların giderek daha fazla kullanılmaya başlanmıştır. Buna paralel olarak arduino çatısında kullanılan sensör modülleri de giderek artmaktadır. Yapılan çalışma ile giriş seviyesinde birçok sensör modülü incelenmiştir yinede uygulamalar geniş arduino havuzunun çok küçük bir kısmını oluşturmaktadır. Yapılan uygulamalarda edinilen tecrübe ile burada sözü geçmeyen birçok sensörün de kolaylıkla kullanılabileceği muhakkaktır.

ŞEKİLLER

Şekil 1. Arduino Uno ve Arduino Nano modülleri	1
Şekil 2. Pic Mikrodenetleyicisi ile 2×16 LCD Ekranı Yazı Yazdırma	2
Şekil 3. Arduino ile 2×16 LCD Ekranı Yazı Yazdırma	2
Şekil 4. Arduino Modüler Yapısı	2
Şekil 5. Arduino Sensör Modülleri	3
Şekil 6. Arduino programı indirme ekranı	4
Şekil 7. Arduino Programlama Arayüzü	4
Şekil 8. Arduino sık kullanılan butonlar ve görevleri	4
Şekil 9. File Menüsü	5
Şekil 10. Edit Menüsü	5
Şekil 11. Sketch Menüsü	6
Şekil 12. Tools Menüsü	6
Şekil 13. Arduino Örnek Uygulamalar Menüsü	7
Şekil 14. Arduino COM Port Ayarları	7
Şekil 15. Arduino Modülünün Tanıtılması	8
Şekil 16. Sücünün Yüklenmesi	8
Şekil 17. Sücünün Yüklenmesi	9
Şekil 18. Sücünün Yüklenmesi	9
Şekil 19. Sücünün Yüklenmesi	10
Şekil 20. Led Yak-Söndür Uygulaması	12
Şekil 21. Led Yak Söndür Akış Diagramı	12
Şekil 22. Buton İle Led Kontrolü Uygulaması	13
Şekil 23. Buton ile Led Kontrolü Akış Diagramı	13
Şekil 24. Arduino Uno PWM Çıkışları	14
Şekil 25. PWM Çıkış Aralıkları	15
Şekil 26. PWM çıkış sinyali uygulaması akış diyagramı	16
Şekil 27. Analog giriş uygulaması devre bağlantısı	17
Şekil 28. Analog giriş uygulaması akış diyagramı	17
Şekil 29. PIR Sensörü	18
Şekil 30. PIR sensör modülü	19
Şekil 31. PIR modülü ve arduino bağlantıları	19
Şekil 32. Arduino ile PIR Sensörü Kontrolü Akış Diagramı	19
Şekil 33. HC-SR04 ultrasonik sensor modülü	20
Şekil 34. HC-SR04 ultrasonik sensor modülü ve LED devre bağlantıları	21
Şekil 35. Arduino ile HC-SR04 Uzaklık Sensörü Kontrolü Akış Diagramı	21
Şekil 36. Arduino programı dosya kütüphanesi	22
Şekil 37. HC-SR04 Uzaklık Sensörü Örnek uygulaması	22
Şekil 38. Kızıl Ötesi Kumanda	24
Şekil 39. Kızılötesi Alıcı – Verici Bağlantıları	24
Şekil 40. Arduino ile Kızılötesi Alıcı Kumanda Kontrolü Akış Diagramı	25
Şekil 41. Seriport izleme programı	26
Şekil 42. Arduino ile DHT11 Sensörü Kontrol Devresi	27
Şekil 43. Arduino ile DHT11 Sensörü Kontrol Devresi Akış Diyagramı	27
Şekil 44. Alev sensörü devre bağlantıları	28
Şekil 45. Sıcaklık Sensörü (Termistör Modülü)	29
Şekil 46. Termistör uygulaması bağlantı şeması	30
Şekil 47. Titreşim sensörü	31
Şekil 48. Kızılötesi Engel Algılayan Sensör	32

Şekil 49. Işığa duyarlı sensör (LDR Modülü)	32
Şekil 50. Işık algılayan sensör (LDR) ve Denetleyici bağlantıları.....	33
Şekil 51. Kızıl Ötesi Sensör	34
Şekil 52. Kızılötesi alıcı devresi bağlantıları	36
Şekil 53. Kızılötesi alıcı sensör bağlantıları.....	36
Şekil 54. Manyetik Etki algılayan sensör	37
Şekil 55. Reed Kontak Modülü	38
Şekil 56. Reed kontak modülü bağlantıları	38
Şekil 57. DS18820 sıcaklık sensörü devre bağlantıları.....	39
Şekil 58. Nabız Sensörü	40
Şekil 59. Kalp Atış Sensörü Bağlantıları	40

GİRİŞ

Arduino yüksek derecede elektronik ve mikrodnetleyici bilgisi gerektirmeden, çeşitli projeleri geliştireceğiniz açık kaynaklı bir geliştirme platformudur. Arduino ile birlikte kullanılacak devre elemanlarının çalışma mantığını ve bağlantı yapısını bilmemiz gerekmektedir. Ayrıca belirli bir mikrodnetleyici bilgisine sahip olanlar arduino'yu öğrenmek ve kullanmak açısından daha hızlı ve büyük adımlar atacaktır. Farklı projeler için farklı arduino geliştirme kartı modelleri bulunmaktadır. Aşağıda gördüğünüz arduino uno başlangıç için ideal görülürken temel projeleri yapmak için yeterli donanımına da sahiptir. Daha az donanımın yeterli olduğu projelerde arduino nano gibi modeller kullanılırken, çok sayıda pin' e ihtiyaç duyulduğunda Arduino Mega gibi modeller tercih edilir.



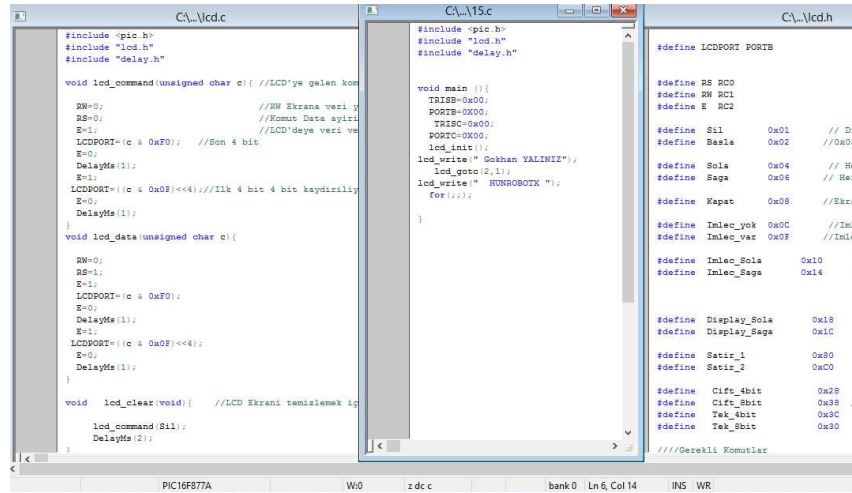
Şekil 1. Arduino Uno ve Arduino Nano modülleri

Avantajları

Arduino' yu diğer geliştirme platformlarından ayıran bazı avantajları vardır. Bu avantajları 3 başlıkta sıralayabiliriz.

1. Hazır Kütüphaneler

Kütüphaneler program tarafından bazı proje elemanlarının, haberleşme protokollerinin vb. daha kolay ve sistematik kontrol edilebilmesi için bir araya getirilmiş komut satırları ve fonksiyonlardan oluşmaktadır. Aşağıda kütüphane oluşturularak bir mikrodnetleyici ile lcd sürülmesi ile arduino ile LCD sürülmesi arasındaki farkı net bir şekilde görebilirsiniz.



Şekil 2. Pic Mikrodenetleyicisi ile 2×16 LCD Ekranı Yazı Yazdırma

Arduino kullanıcılarına, bu kütüphaneleri hazırlayabilecek bilgiye sahip olmadan yalnızca birkaç satırda istediklerine ulaşmasını sağlar.



Şekil 3. Arduino ile 2×16 LCD Ekranı Yazı Yazdırma

2. Hazır modül veya kitle

Arduino ile kolay bir şekilde ileri seviye bilgi gerektiren devreleri kontrol edebilirsiniz.



Şekil 4. Arduino Modüler Yapısı

Aynı görevi benzer görev için tasarlanmış modüller ile de yapabilirsiniz.



Şekil 5.Arduino Sensör Modülleri

3. Döküman

Arduino doküman açısından en fazla kaynağa sahip geliştirme platformudur.

Dezavantajları

Arduino'yu yüksek hız gerektiren uygulamalarda istenileni verememektedir. Bu nedenden ötürü hız gerektiren uygulamalarda farklı yöntemlere başvurulmaktadır. Ekler kısmında sunulmuş olan Mevcut arduino modelleri tablosunda gösterilmekte olan arduino modellerinin büyük bir kısmının çalışma frekansının 16Mhz olduğu görülmektedir. Bu çalışma frekansı sıkça kullanılan haberleşme, veri edinimi ve motor kontrol uygulamalarında yetersiz kalmaktadır. Bu başlık altında bahsedilmesi gereken bir diğer konuda arduino platformunun birçok şeyin kolaylıkla yapılabildiği ve teknik bilgiye ihtiyaç duyulmadığı yanılgısıdır. Arduino platformu her ne kadar programlama ve modüler yapısı ile birçok avantaj sağlasa da sağladığı kolaylıkların yanında mikrodenetleyici tabanlı çalışan bir platformdur. Uygulama geliştirilmesinde temel elektronik ve programlama (özellikle C Dili) altyapısına sahip olunması çok daha işlevsel uygulamalar geliştirilmesine olanak sağlamaktadır. Arduino platformunda işlevsel ve sorunsuz uygulamaların geliştirilmesi de diğer mikrodenetleyici uygulamalarında olduğu gibi zaman alan bir süreçtir.

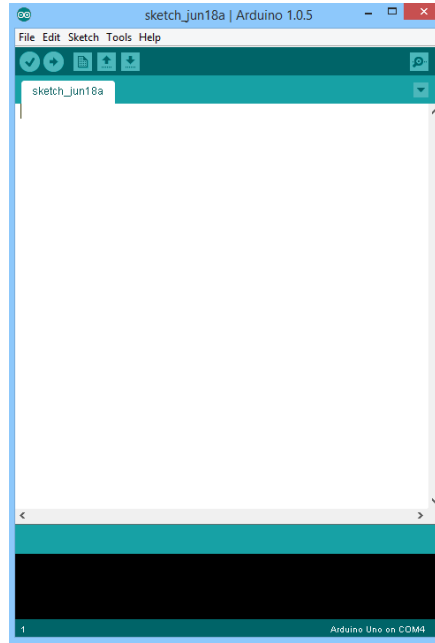
Arduino Ara yüzü

Arduino arayüzü kullanım açısından çok rahattır. Arduino yazılımını <http://arduino.cc/en/main/software> sitesinden, kullanılabilecek işletim sistemine uygun versiyonunu ücretsiz olarak indirilebilmektedir.



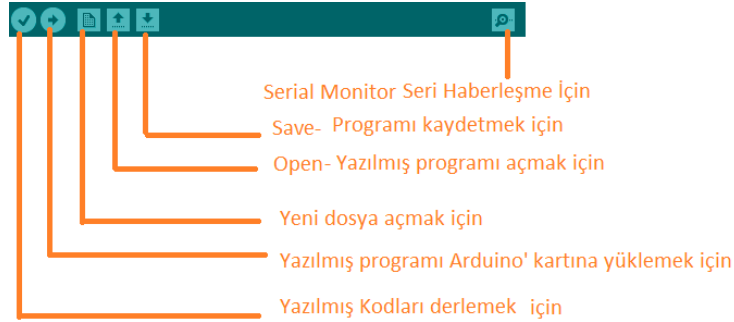
Şekil 6.Arduino programı indirme ekranı

Windows işletim sistemi için zip dosyasını indirdikten sonra ayıklayıp arduino. Exe'ye tıklayarak programı kurmanız yeterli olacaktır. Kurulum işlemi tamamlandıktan sonra Şekil... da görüldüğü gibi sade bir ekran ile karşılayacaktır.



Şekil 7. Arduino Programlama Arayüzü

Karşımıza gelen birkaç buton ve basit bir menüden oluşan arayüzde sıkça kullanılan butonlar ve almaları aşağıdaki şekilde gösterildiği gibidir.



Şekil 8. Arduino sık kullanılan butonlar ve görevleri

Arayüzün üst kısmında Word, Excel programlarda da yer alan temel bir menü çubuğu mevcuttur. Bu bölümde yer alan komutlarla önceki şekilde anlatılan sık kullanılan butonların yaptığı işlevler yerine getirilebilmekte, buna ek olarak modül ayarlamaları, iletişim seçenekleri, ileri program ayarlamaları gibi işlevlerde yerine getirilebilmektedir.

File Sekmesi

Aşağıdaki şekilde “File” sekmesi görülmektedir. Burada birçok programda standart olarak yer alan dosya aç, yeni dosya..vs gibi komutlar yer almaktadır. Buna ek olarak “Examples” [örnekler] komutu ile program kütüphanesinde yer alan kodlar incelenebilmektedir. Çok sayıda örneğin bulunduğu bölümde arduino ile birlikte kullanılacak gerekli bağlantıları yaparak görsel olarak örnekleri inceleyebilir, küçük değişiklikler yapıp etkisini görebilirsiniz. Bu bölüm yeni başlayanların için programa alışılması açısından büyük kolaylık sağlamaktadır.

File	Edit	Sketch	Tools	Help
New				Ctrl+N
Open...				Ctrl+O
Sketchbook				
Examples				
Close				Ctrl+W
Save				Ctrl+S
Save As...				Ctrl+Shift+S
Upload				Ctrl+U
Upload Using Programmer				Ctrl+Shift+U
Page Setup				Ctrl+Shift+P
Print				Ctrl+P
Preferences				Ctrl+Comma
Quit				Ctrl+Q

Şekil 9.File Menüsü

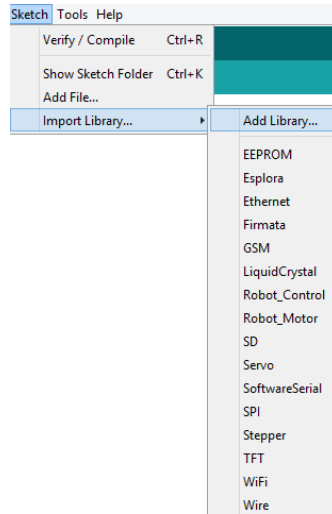
Edit Sekmesi

Standard işlemlerin dışında Edit sekmesinde Copy for Forum ve Copy as HTML seçenekleri kullanılarak internet ortamında, arduino arayüzünde yazdığınız biçimde (renk-yazı tipi) paylaşabilirsiniz.

Edit	Sketch	Tools	Help
Undo			Ctrl+Z
Redo			Ctrl+Y
Cut			Ctrl+X
Copy			Ctrl+C
Copy for Forum			Ctrl+Shift+C
Copy as HTML			Ctrl+Alt+C
Paste			Ctrl+V
Select All			Ctrl+A
Comment/Uncomment			Ctrl+Slash
Increase Indent			Ctrl+Close Bracket
Decrease Indent			Ctrl+Open Bracket
Find...			Ctrl+F
Find Next			Ctrl+G
Find Previous			Ctrl+Shift+G
Use Selection For Find			Ctrl+E

Şekil 10. Edit Menüsü

Sketch Sekmesi

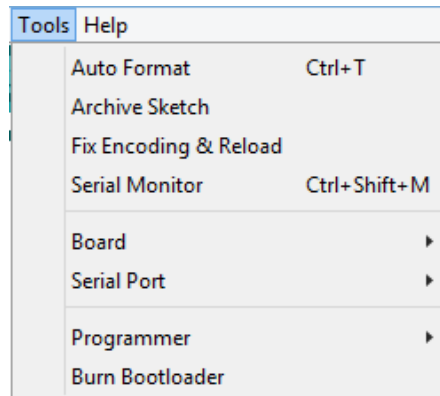


Şekil 11. Sketch Menüsü

Standard işlemlerin dışında Sketch sekmesini kullanarak istediğiniz kütüphaneyi otomatik olarak kodunuza ekleyebilirsiniz. Ayrıca başka bir kütüphane eklemek isterseniz Add library'i kullanabilirsiniz.

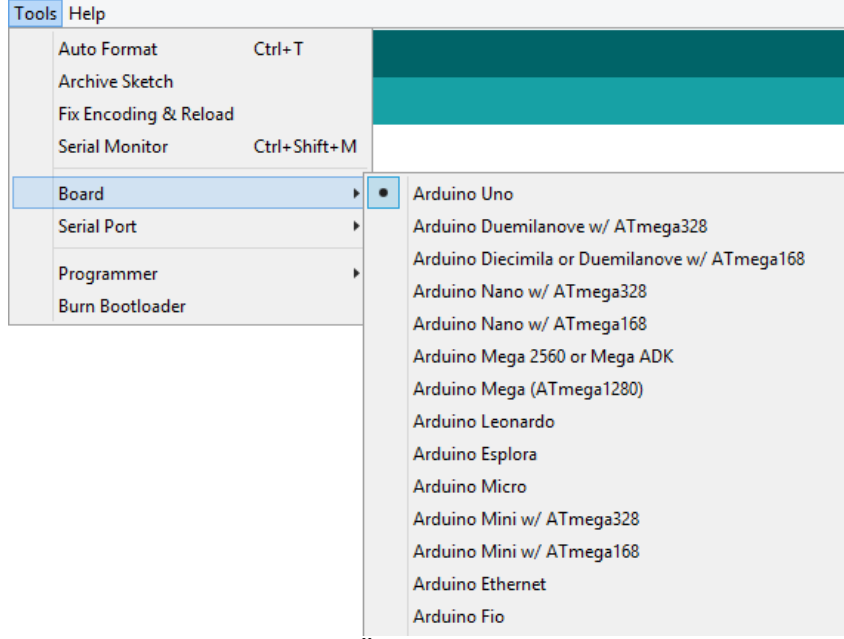
Tools Sekmesi

Standard işlemlerin dışında Tools sekmesini kullanarak Board kısmından programlayacağınız arduino modelini seçimi, Serial Port kısmından kullandığınız arduino'yu bilgisayarınızın hangi portuna bağladığınız seçmelisiniz. Arayüzde en çok kullanılan kısımlar bunlardır.



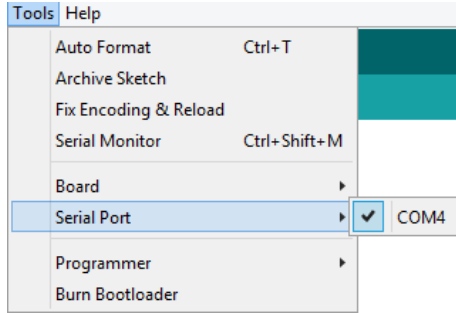
Şekil 12. Tools Menüsü

Türkiye'de en çok satan ve uygulaması bulunan modül Arduino Uno'dur. Eğer arduino uno'yu kullanıyorsanız Board seçiminiz aşağıdaki gibi olmalıdır.



Şekil 13. Arduino Örnek Uygulamalar Menüsü

Arduino temelde bilgisayarın USB arayüzünden iletişim kuran bir yapıdır. Yukarıda gösterilmiş olan şekildeki SerialPort bölümünden modülün bilgisayar ile olan bağlantı ayarı yapılmaktadır. Modülün bilgisayar ile bağlantıya geçebilmesi için arduino'nuzun bağlı olduğu port ile arayüzde seçili olan portun aynı olduğundan emin olunmalıdır.



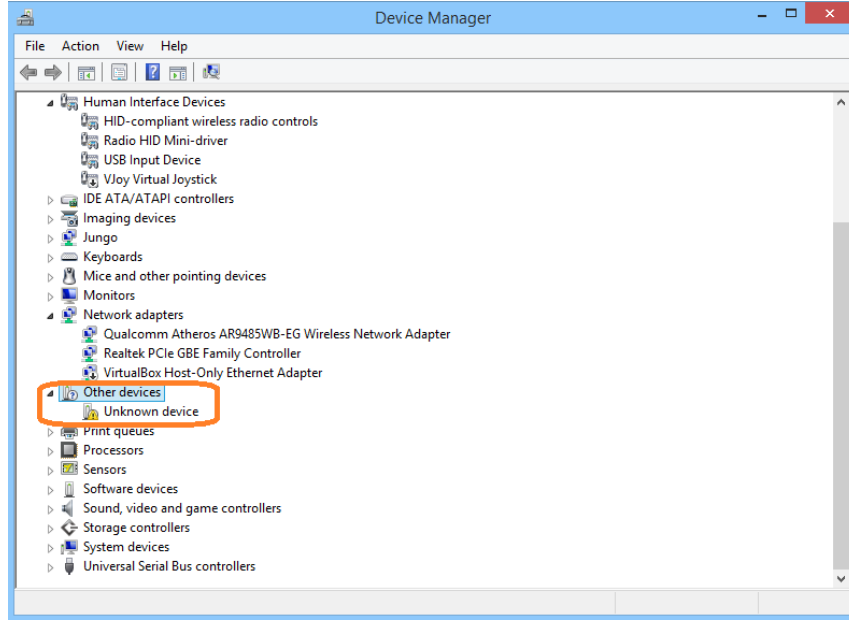
Şekil 14. Arduino COM Port Ayarları

Arduino Modülünü Bilgisayara Tanıtmak

Arduino'yu bilgisayarınıza ilk kez taktığınızda doğal olarak bilgisayar arduino'nuzu tanımayacaktır, arduino'yu bilgisayara tanıtmak için daha önceden indirdiğimiz arduino dosyasını içinde bulunan driver dosyasını yüklememiz gerekmektedir. Bu işlem için aşağıdaki adımları inceleyeceğiz.

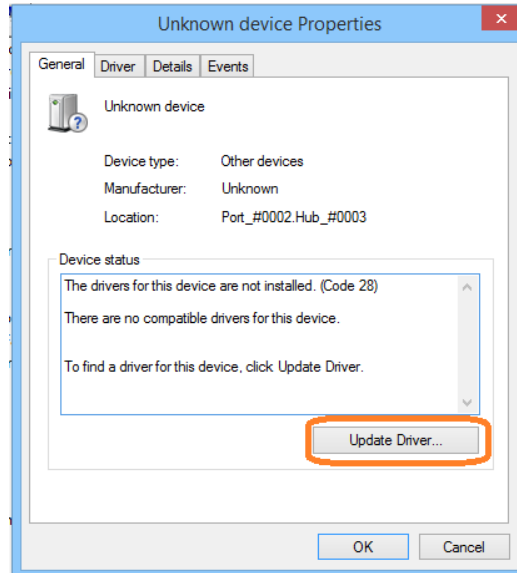
Windows işletim sistemi için, **Control Panel → Hardware and Sound → Devices and Printers → Devices Manager** kısmını açınız (Türkçe olan işletim sistemlerinde Aygıt Yöneticisini bölümü açılacak)

Aşağıdaki şekilde de görüldüğü gibi unknown device-bilinmeyen aygıt olduğunun farkına varacaksınız.



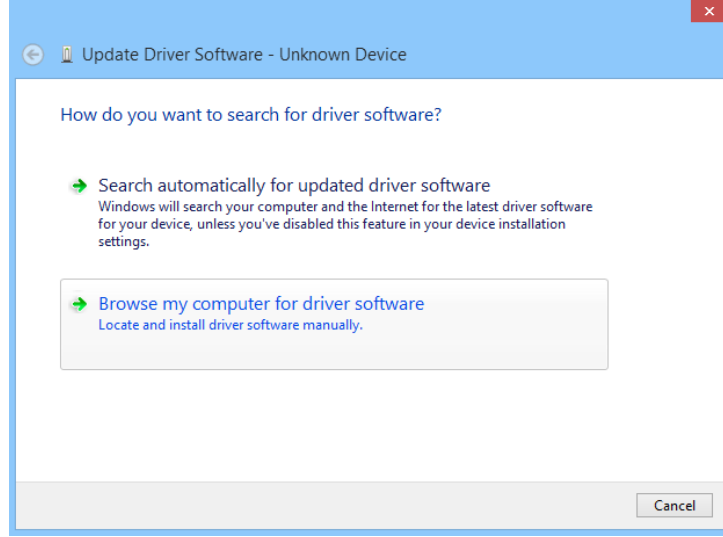
Şekil 15. Arduino Modülünün Tanıtılması

Unknown device'ın üzerine tıklayıp, açılan pencereden update drivers butonuna basacağız.



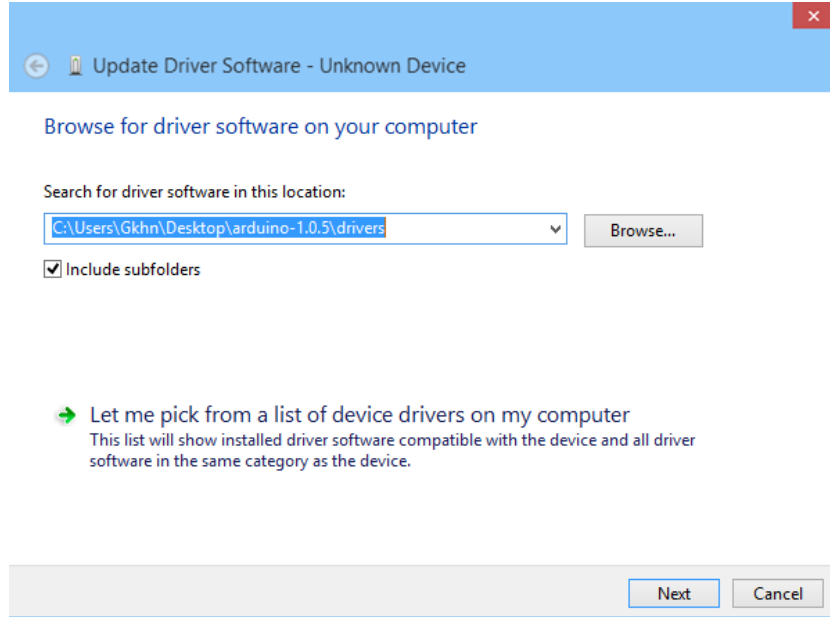
Şekil 16. Sürücünün Yüklenmesi

Daha sonra gelen pencereden browse my computer for driver software seçeneğinin üzerine tıklayacağız.



Şekil 17. Sürücünün Yüklenmesi

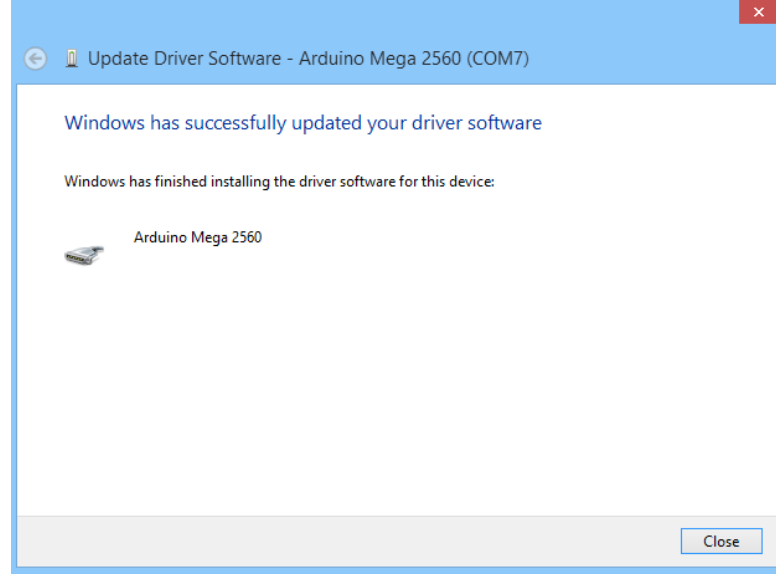
Ardından browse kısmı için arduino'yu indirdiğimiz klasörün içindeki drivers klasörünü seçip Next butonuna basacağız.



Şekil 18. Sürücünün Yüklenmesi

Bu adımlardan sonra arduino'nun başarılı bir şekilde yüklenmesi beklenmektedir.

Eğer arduino'nuz aşağıdaki gibi başarılı bir şekilde yüklenmez ise işletim sisteminize göre "sürücü imzalama zorlamasını" devre dışı bırakıp tekrar deneyiniz.



Şekil 19. Sürücünün Yüklenmesi

Gördüğünüz gibi arduino modeli ve portu yukarıda anlattığıma göre farklı, arayüzde hem modeli hem de port numarasını yeniden ayarlamamız gerekmektedir.

Arduino Programlamanın Temel Yapısı

Bir mikrodeneleyiciyi veya bir geliştirme kartının programlarken temel olarak 2 yapıtaşına ihtiyacımız bulunmaktadır. Birincisi başlangıç ayarlarının yapıldığı, ikincisi mikrodeneleyici veya kartınızın sürekli yapacağı görevin bulunduğu kısımdır. Burada üzerinde durulması önemli bir nokta da tüm program geliştirme uygulamaların gerekli olan temel seviye de olsa bir programlama dilinin bilinmesi (tercihen C dili) ve akış şemalarına aşina olmanın gerekliliğidir. Aşağıda ki program prosedüründe mikrodeneleyiciye ait temel ayarlar yapılmaktadır. “void setup()” ile başlayan satır, takip eden tırnak içindeki bölümde temel ayarların yapılacağını belirtmektedir. Burada yapılan ayarlamalarda hangi mikrodeneleyici pininin (veri bacağının) giriş=Input (veri çekilen port) yada çıkış=Output (veri gönderilen port) olduğu belirtilmektedir.

```
void setup()
{ // Arduino'nun başlangıç ayarlarının
  // yapıldığı kısımdır. (setup fonksiyonu)
}
```

Aşağıda gösterilmiş olan “Loop” [Döngü] prosedür ise setup prosedüründen sonra eklenen ve mikrodeneleyici yada arduinomuzun beslemesi devam ettiği sürece tekrarlanan komutlar yazılmaktadır. Buraya yazılan komutlar ile arduino pinleri arasından muayese, ilişkilendirme, matematiksel işlemler vs... yapılmaktadır. Daha açık bir ifade ile bu bölüme yazılan komutlar arduino ile yapılacak işlemlerin çatısını oluşturmaktadır.

```
void loop()
{ // Tekrarlana ana programın yazıldığı bölüm...
}
```

Aşağıda gösterilmiş olan örnek komut satırlarında arduino pinlerinin Input (veri okulanan giriş) ve Output (veri yazılan çıkış) olarak ayarlanması gösteriştir. Birinci program satırından ilgili pin, Buton olarak isimlendirilmiş ve Input olarak ayarlanmıştır. İkinci program satırından aynı yolla, pin LED olarak isimlendirilmiş ve Output olarak ayarlanmıştır.

```
pinMode(BUTTON, INPUT); // pin input olarak ayarlanıyor
pinMode(LED, OUTPUT); // pin output olarak ayarlanıyor
```

Aşağıdaki komut satırı ile mikrodnetleyici uygulamalarında önemli bir yeri olan iletişi konusu ile ilgilidir. Burada gösterilmiş olan komut satırları ile Arduino'nun seri haberleşmeye başlaması sağlamaktadır. Burada seri haberleşmeden kasıt Rx ve Tx olarak isimlendirilen iki iletişim portu aracılığı ile modüller arası, yada modül ile bilgisayar arasından seri iletişim yolu ile yapılan bilgi alışverişidir. Komut satırından gösterilmiş olan “Serial.begin” komutu seri iletişim prosedürü başlatmakta, parantez içinde gösterilmiş olan “9600” değeri de, saniyede iletilen veri sayısı cinsinden (Bit Per Second=Boundraid) veri iletim hızını belirtmektedir. Burada dikkat edilmesi gereken önemli nokta seri iletişim işleminde iletişimin sağlandığı iki modül yada modül-PC ikilisinde her iki tarafından aynı veri iletişim hızına ayarlanmış olması gerektiğidir. Aksi bir durumda alıcı ile verici aynı frekansta olmaması nedeni iletişim doğru şekilde gerçekleşmemiş olacaktır.

```
Serial.begin(9600); // Seri haberleşme başlatılıyor
```

Mikrodnetleyici dünyasında bir pinin çıkış olarak ayarlanması, ilgili pin yolu ile denetleyicinin dış ortama elektriksel bir bilgi göndermesi anlamına gelmektedir. Gönderilen veri ise analog yada dijital olarak ayarlanabilmektedir. Burada üzerinde durulması gereken konu dijital bir verinin 1 yada 0'lardan oluştuğudur. Mikrodnetleyiciler için ise bir pinde ki yaklaşık +5 V' luk bir çıkış aynı zamanda 1 yada “on” anlamına gelmektedir. Benzer şekilde çıkışta gerilim olmaması yada çıkışın 1.5 V' dan daha az olması 0, “off” anlamına gelmektedir. Analog veri tipinde ise iletişim tipinde ise çıkış 0 ile 256 arasında istenen değere göre değeri alabilmektedir.

Aşağıdaki gösterilmiş olan komut satırlarında bir veri pininin dijital olarak 1 yada sıfır olarak ayarlanması üzerinde durulmuştur. Burada HIGH sözcüğü ile 1 yani 5V, LOW sözcüğü ile ise 0 diğer bir deyişle 0V ayarlaması yapılmıştır. Burada yapılan işlem ile PIN'e bağlanmış olan bir LED yakılıp söndürülebilmekte yada komplike bir proses, sistem yada süreç durdurulup başlatılabilmektedir (Örneğin bir motorun 10 dk süre ile çalıştırılıp kapatılması).

```
digitalWrite(LED, HIGH); // pin'e dijital 1 gönderiliyor (5V)
digitalWrite(LED, LOW); // pin'e dijital 0 gönderiliyor (0V)
```

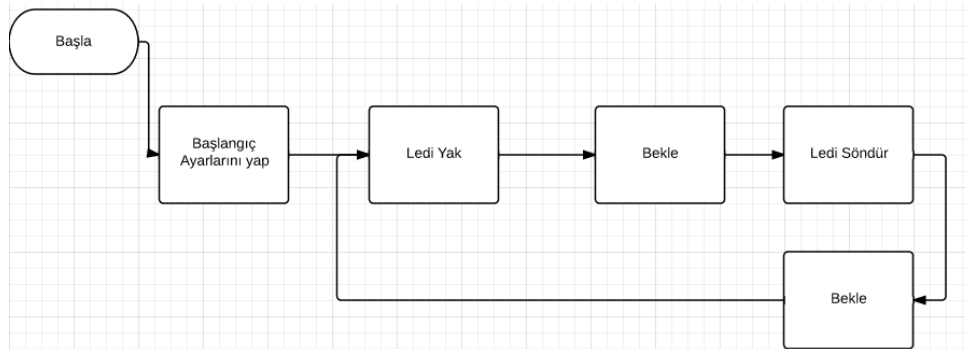

Aşağıda yapılmış olan örnek uygulama ile Arduino' nun çıkış Pin' ine bağlı bir LED'in yakılıp söndürülmesi üzerinde durulmuştur.

Örnek 1. Led Yak-Söndür Uygulaması



Şekil 20. Led Yak-Söndür Uygulaması

Arduino temel öğrenim kolaylığı ile ön plana çıkan bir modüler altyapıdır. Donanım yapısı itibari ile birçok programın sınanması konusunda programcıya büyük kolaylık sağlamaktadır. Yapılan bu uygulamada yine uygulamaların sınanmasında büyük kolaylık sağlayan karta entegre LED'in yakılıp söndürülmesi üzerinde durulmuştur. Modüle entegre olan bu LED'in arduino tasarımı incelenirse 13. PIN' e bağlı olduğu görülmektedir.



Şekil 21. Led Yak Söndür Akış Diyagramı

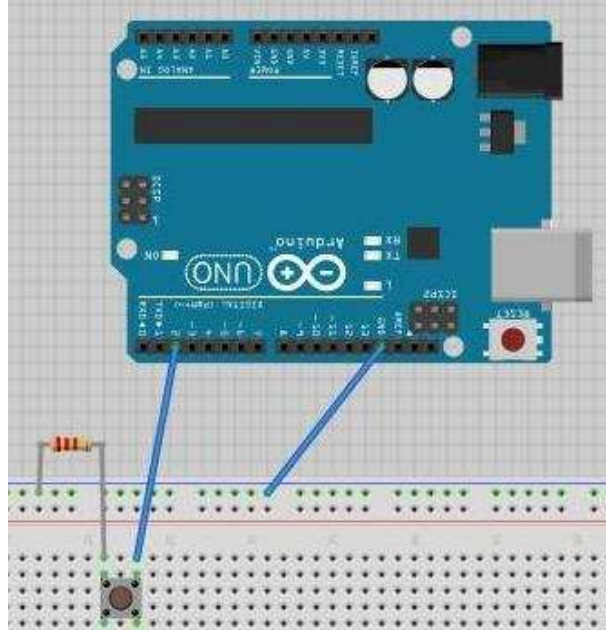
Uygulamaya ait akış kontrol diyagramı yukarıda gösterildiği gibidir. Akış kontrol şemasında görüldüğü gibi uygulama bir adet döngü ve biri başta diğer sonda olmak üzere iki adet bekleme kullanılmıştır. Burada Başlangıç ayarlarını yap kısmında 13 Nolu pin çıkış olarak ayarlanmaktadır. Bekleme bölümünde led'in yanması ve sönme süreleri belirtilmektedir. Programa ait uygulama kodları aşağıdaki gibidir.

```
#define LED 13 // Arduino Uno Kartının üzerinde 13. Pin de
bulunan
// void setup() // Başlangıç Ayarları yapılıyor.
{pinMode(LED, OUTPUT); // Led OUTPUT, çıktı olarak ayarlanıyor.
digitalWrite(LED,LOW); // Ledin başlangıç durumu LOW,dijital 0 olarak
ayarlanıyor. Program ile çalıştırıldığın LED // yanarak uygulamaya başlanıyor.
}
void loop() // Döngüye giriliyor.
{ digitalWrite(LED,HIGH); // LED'E Dijital 1 veriliyor LED yanıyor
delay(500); // 500 ms bekleme için Delay fonksiyonu kullanılıyor
digitalWrite(LED,LOW); // LED'E Dijital 0 veriliyor LED sönüyor
```

```
delay(500); // 500 ms beklemesi için Delay fonksiyonu kullanılıyor, ve daha sonra program tekrarlanıyor.
}
```

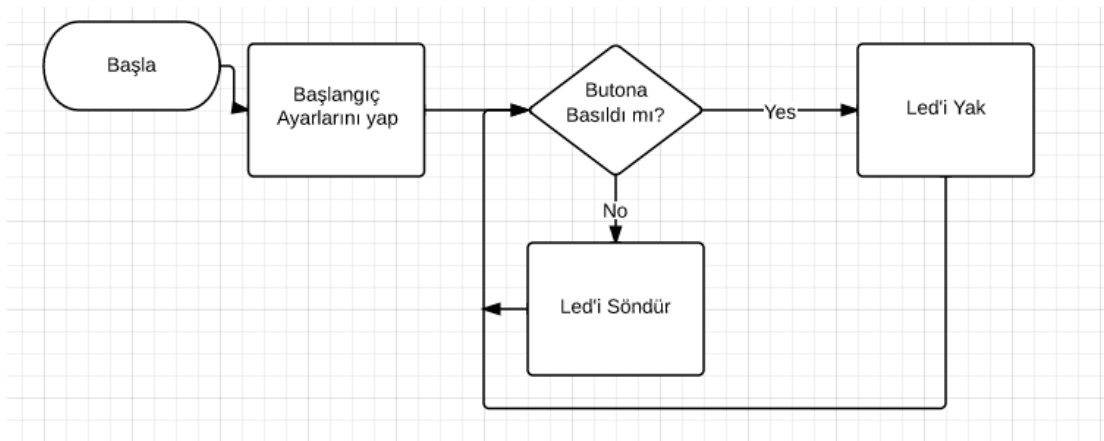
Örnek 2. Buton İle Led Kontrolü Uygulaması

Yapılan bu uygulamada da önceki uygulamada olduğu gibi 13 nolu PIN'e bağlı bir LED yakılıp söndürülmeye çalışılmıştır. Fakat önceki programa ek olarak LED yak sön bilgisi bir zamanlama fonksiyonu yerine butona bağlı olarak belirlenmiştir.



Şekil 22. Buton İle Led Kontrolü Uygulaması

Burada anlaşılması zor olabilecek nokta 10k direncin olduğu kısımdır. Arduino' ya butona basıldığında 5V, basılmadığında ise 0V gönderilmesi gerekmektedir. Bu veriler butonun arduino' ya bağlı olduğu pin aracılığı ile yapılmaktadır. Eğer 5V ve 0V 'u direk olarak birbirine bağlar isek kısa devre oluşacak ve arduino' ya zarar gelecektir. Bunun için 10k bir nevi koruyucu görev görmektedir.



Şekil 23. Buton ile Led Kontrolü Akış Diyagramı

```

#define LED 13 // Arduino Uno Kartının üzerinde 13. pinde bulunan LED
#define BUTTON 12 // Arduino Uno Kartının üzerinde 12. pine bağlanan Bulunan BUTON
void setup() // Başlangıç Ayarları yapılıyor.
{
  pinMode(LED, OUTPUT); // Led OUTPUT, çıktı olarak ayarlanıyor.
  pinMode(BUTTON, INPUT); // BUTTON INPUT, girdi olarak ayarlanıyor.
  digitalWrite(LED, LOW);
} void loop()
{
  if( digitalRead(BUTTON) == HIGH) // Eğer Buton'a basıldıysa
    digitalWrite(LED, HIGH); // Led Yansın
  else // basılmadıysa
    digitalWrite(LED, LOW); // Led Sönsün
}

```

Örnek 3. Analog Çıkış Uygulaması

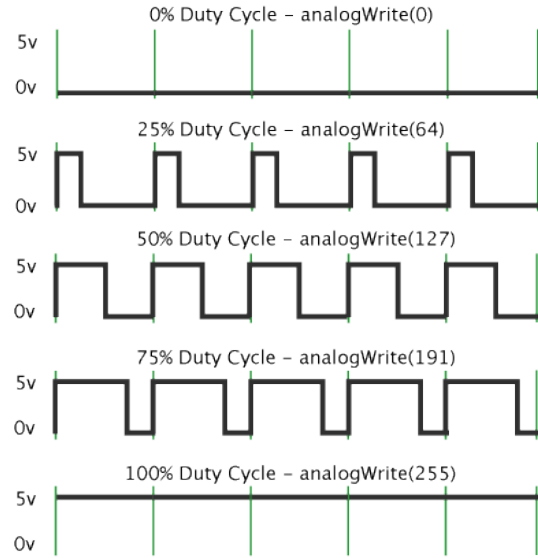
Veri alışverişi uygulamaları analog ve dijital olarak iki kısımdan oluşmaktadır. Önceki yapılan uygulama ile 1 ve 0'dan oluşan veri iletişimi ve verinin kullanımı üzerinde kısaca durulmuştur. Dijital veriler yapılan uygulamalarda 1 ve 0 ve kombinasyonları kullanılmaktadır. Fakat analog verilerde böyle keskin bir farkı yoktur. Gönderilen veri 0 ile 256 arasında değer (8 Bitlik bilgi) alabilmektedir. Dijital verilerde 0 ve 1'in teoride karşılığı 0V ve 5V iken, pratikte 0V 'a yakın değerler ve 5V'a yakın değerlerden oluşur. Analog verileri ise 0V ve 5V arasında aklınıza gelebilecek bütün değerler olabilirler.(Genelde çoğu entegre işlemlerini 0-5V aralığında yapar, bununla beraber PLC gibi endüstriyel uygulamalarda yada CMOS entegrelerde 0-12V, 0-24V gibi standartlarda kullanılmaktadır. Sensör uygulamalarında ise sensörün ve ölçülen bilginin tipine göre analog veya dijital iletişime ihtiyaç duyulabilmektedir.

Yapılan bu çalışmada donanım olarak arduino Uno modülünden yararlanılmıştır. Arduino Uno modülünde analog çıkış olarak PWM modülünden yararlanılmıştır. Aşağıda gösterilmiş olan şekilde bazı PINlerin üzerinde “~” işaretinin olduğu görülmektedir. Bu işaret PIN aracılığı ile PWM çıkışı alınabileceğini göstermektedir.



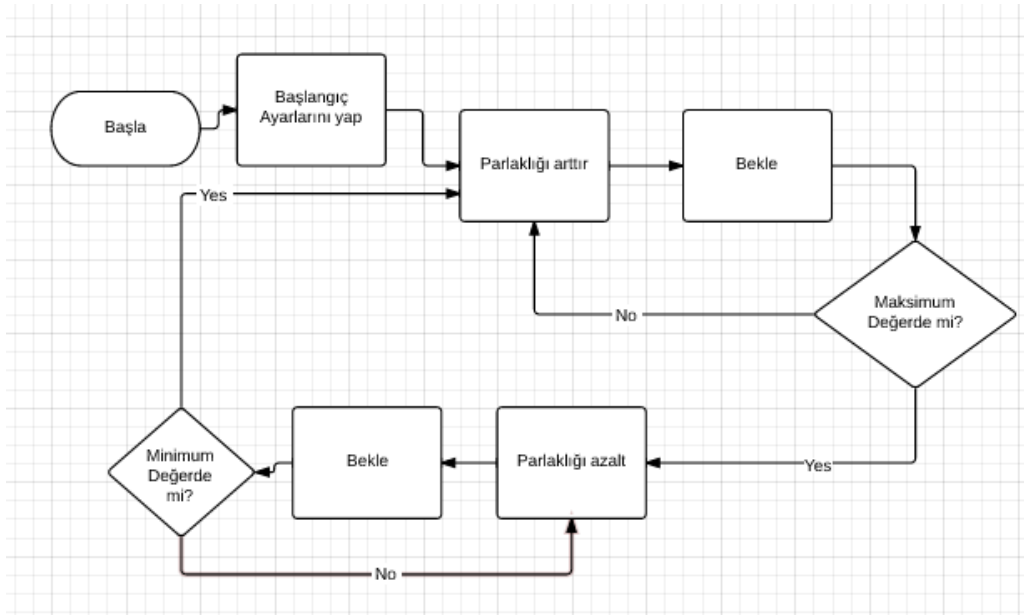
Şekil 24. Arduino Uno PWM Çıkışları

Bu konunun anlaşılması için öncelikle PWM (Pulse Width Modulation =darbe-sinyal genişliği modülü) konusuna açıklık getirilmesinde yarar vardır. Dijital kontrol uygulamalarında mikrodnetleyici elemanımız veri gönderme işlemini genel yapı itibarı ile 0 V yada 5 V (0/1) olarak göndermektedir. Fakat bu veri iletim hızını oldukça düşürmekte ve geliştirilen uygulamaları ağırlaştırmaktadır. Bunun temel nedeni işlemcinin iş yükü ve işlem sırası düşünüldüğünde veri gönderme işlemi yaparken başka bir iş yerine getirilememekte ve yazılan program ağırlaşarak modülün cevap süresi uzamaktadır. Bu durum arduino modülleri içinde söz konusudur. Problemin aşılmasında ise temelde Aç-Kapa kontrol uygulamalarında kullanılan teknikten yararlanılmaktadır. Çıkış pin' in belirli bir aralıkta aç kapa yapılmasına dayanan bu yöntemde pin' den alınan gerilim bilgisi 0V ile 5V arasında değiştirilebilmektedir. Burada bilmemiz gereken önemli nokta aç kapa işleminde pin' in açık(5V) kaldığı sürenin artması ile çıkış geriliminin de 5V'a yaklaştığıdır. Böylelikle PWM sinyali kullanılarak istenen 0V-5V aralığı elde edilebilmektedir. Arduino modüllerinde 0V ile 5V arasında 256 (28) farklı değer üretilebilmektedir. PWM çıkışında 0V elde etmek istediğimizde çıkışa 0 (20-1), PWM' den 5V çıkış elde etmek istediğimizde ise 255 (28-1) göndermemiz gerekmektedir. Aralıkta elde edilmek istenen farklı bir değer içinde ara değerlerin verilmesi yeterli olacaktır. Aşağıda gösterilmiş olan şekilde %0(0V), %25(1,25V), %50(2,5V), %75(3,75V) ve %100 (5V) çıkışları için üretilen PWM çıkış sinyalleri gösterilmektedir.



Şekil 25. PWM Çıkış Aralıkları

Aşağıda şekilde PWM çıkış aracılığı ile, LED'den elde edilen ışık şiddeti ayarlanan bir uygulama yapılmıştır. Akış diyagramı inlenirse LED'in parlaklığı artırılırken değer in Maksimum bir seviyeye ulaşması amaçlanmaktadır. Değer in maksimum değer ulaşmaması durumunda artırılan değer PWM çıkışıdır. Maksimum değer ise programcı tarafından daha önce girilmiş olan ve Led' in istenen parlaklık değerini ifade eden 0 ile 255 arasından bir değerdir.



Şekil 26. PWM çıkış sinyali uygulaması akış diyagramı

Uygulamaya ait program kodları aşağıdaki gibidir.

```

#define LED 11 // LED 11. pinde
int i=0; // parlaklık değişkeni
void setup(){
  pinMode(LED, OUTPUT); // LED output
}
void loop(){
  for(i=0;i<256;i++){ // pwm 0 ile 255 değerinde
    analogWrite(LED,i); // parlaklık değişkeni i ile
    // Analog olarak outputa gönderiliyor
    delay(7); // parlaklığın değiştiğini
    // rahatlıkla görebilmek için
    if(i==255){ // pwm değeri maksimum olduğunda
      for(;i>0;i--){ // bu kez parlaklık değişkeni azaltılıyor
        if(i==255){ // pwm değeri maksimum olduğunda
          for(;i>0;i--){ // bu kez parlaklık değişkeni azaltılıyor
            analogWrite(LED,i); // analog olarak outputa gönderiliyor
            delay(7); // parlaklığın değiştiğini
            // rahatlıkla görebilmek için
            if(i==1) // parlaklık değişkeni minimum olduğunda
              break; // parlaklığı yeniden arttırılabilmek için
          } // break komutu ile ikinci for döngüsünden çıkıp
          // ilk for döngüsüne gidiyoruz.
        }
      }
    }
  }
}

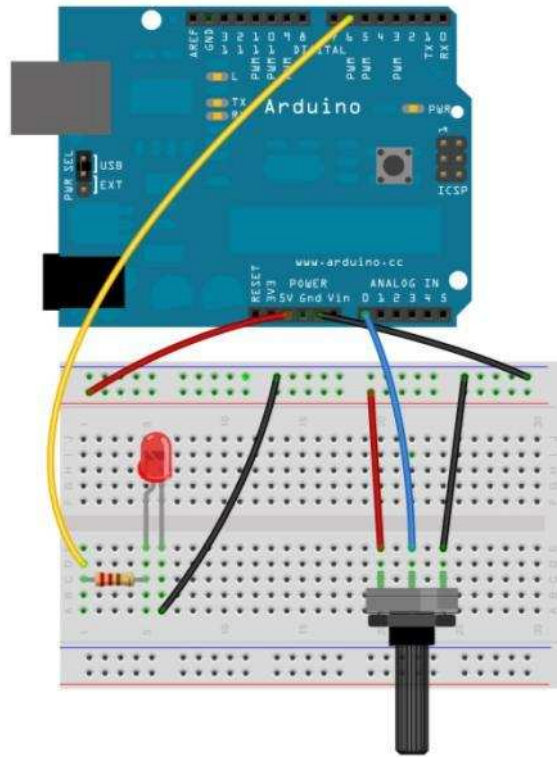
```

Örnek 4. Analog çıkış uygulaması

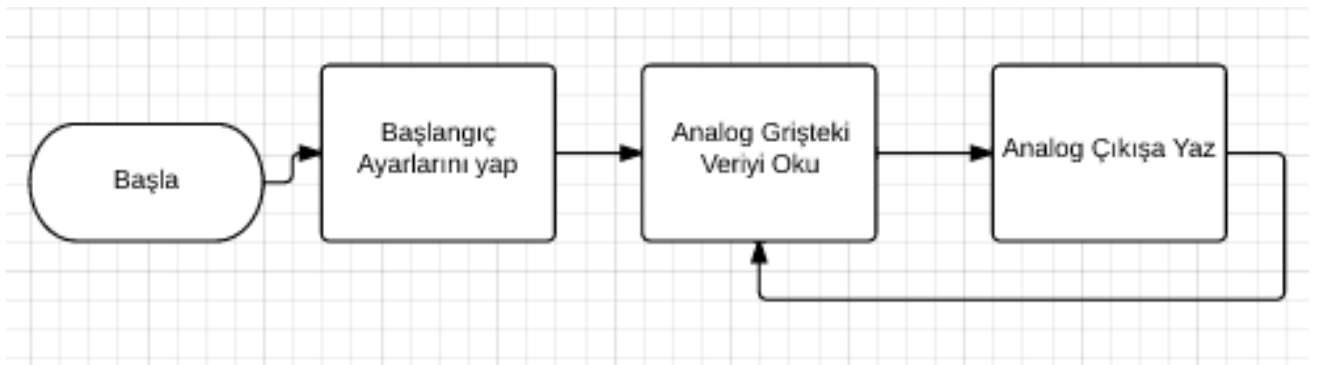
Mikro denetleyicilerde analog çıkış sinyalinin üretilmesinin yanında analog sinyalin okunarak anlamlı hale getirilmesi de önemli bir konudur. Sensör uygulamalarında, sensörden elde edilen çıkış sinyali pek çok uygulamada analog niteliktedir. 1/0 yani Var/yok şeklinde çıkışlar ise pek çok uygulama da yetersiz kalmaktadır. Bu nedenden analog input konusuna da açıklık getirmekte yarar vardır. Bu uygulamada analog girişin okunması konusunda bilgiler sunulmuş ve konu ile ilgili bir örnek verilmiştir. Analog verinin gönderilmesi işleminde hatırlanacağı gibi gönderilen veri (0-255) aralığında değer alabilmekte idi. Analog giriş

uygulamalarında ise verinin girdiği port' a bağlı bir ADC (Analog-Dijital Dönüştürücü) aracılığı ile bilgi dijitalleştirilmektedir. Analog veri girişte 0-5V aralığında değişmektedir. Bu değer çevirme işlemine 0 ((20-1) ile 1024 (210-1) aralığına da yeniden skalalandırılmaktadır.

Aşağıda gösterilmiş olan uygulamada arduino modülüne bağlı bir potansiyometre ile LED'in parlaklık değeri değiştirilmeye çalışılmaktadır. Burada potansiyometrenin ürettiği çıkış Arduino modülü için analog olarak algılanmakta ve değer 0-1024 aralığında tekrar skalalandırılmaktadır. Daha sonra elde edilen sayı tekrar düzenlenerek 0-255 değer aralığında ifade edilmekte ve LED'i yakan PWM sinyali üretilmektedir. Uygulama da kullanılan potansiyometre modülün A0 pinine bağlanmaktadır. Potansiyometreden elde edilen bilgi 9.Pin'e bağlı LED'in ışık şiddetini değiştirmektedir.



Şekil 27. Analog giriş uygulaması devre bağlantısı



Şekil 28. Analog giriş uygulaması akış diyagramı

Uygulamaya ait program kodları aşağıdaki gibidir.

```
#define LED 11 // LED 11. pinde
int i=0; // parlaklık değişkeni
int analog_in=0;
void setup(){
  pinMode(LED, OUTPUT); // LED output
}
void loop(){
  analog_in=analogRead(0); // Analog veri okunuyor
  analogWrite(LED,analog_in/4); // Analog input 10 bit 1024 olduğu için yaklaşık
  olarak // değerlere doğru ulaşmak için input 4'e bölünerek
  outputa gönderiliyor...
}
```

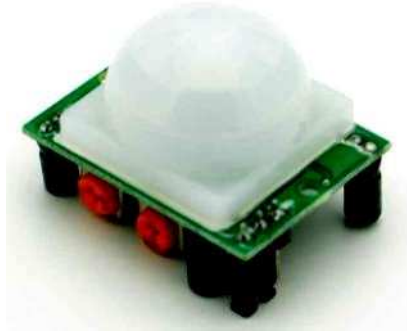
Örnek 5. Arduino İle PIR Sensörü Kontrolü

Yapılan bu uygulama da arduino modülü kullanılarak hareket sensörünün kontrolü yapılacaktır. Uygulamada piyasada PIR (Passive Infrared) olarak bilinen harekete karşı duyarlı bir sensör kullanılacaktır. Sensörden gelen bilgiye göre bir LED'in yanması kontrol edilecektir.



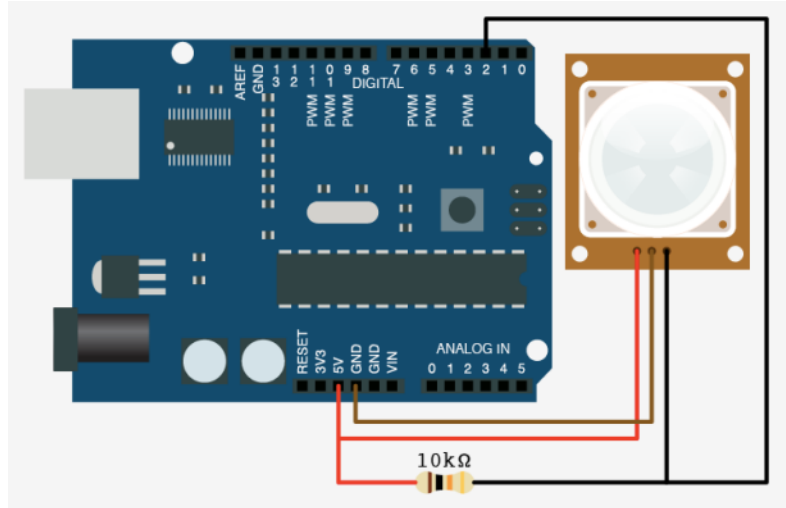
Şekil 29. PIR Sensörü

PIR sensörü kızıl ötesi ışınlar, çıkışındaki elektriksel yük miktarının değiştirerek tepki göstermektedir. PIR sensöründen alınan yük miktarı bilgisi Arduino modülünü içerisinde bulunan dahili FET (Alan Etkili Transistor) yardımı ölçülerek mikrodenetleyicilerin anlayacağı dile çevrilecektir. Yapılan uygulamada Arduino için tasarlanmış PIR sensör modülü kullanılmıştır. Bu modülün kullanılması bağlantıların yapılması açısından büyük kolaylık sağlamaktadır.



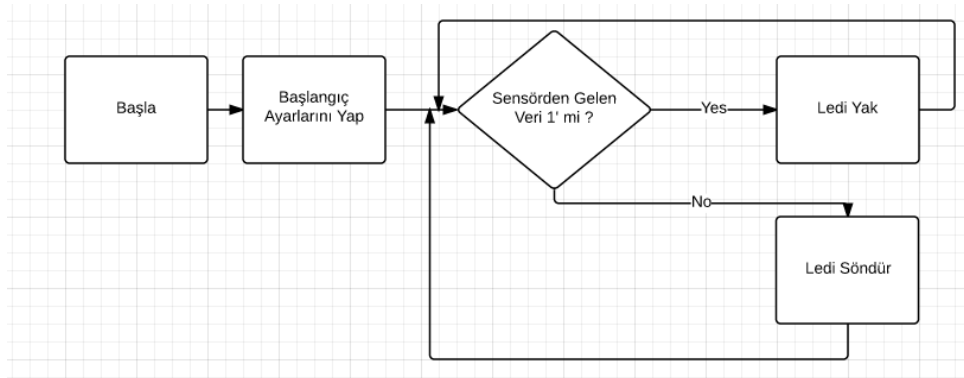
Şekil 30. PIR sensör modülü

PIR sensörü modülü dijital olarak kontrol edilir. Bu gibi devre elemanları ile ilgili ayrıntılı bilginin modülün teknik dökümanlarına (datasheet) bakılarak öğrenilmesinde yarar vardır. Aşağıdaki gösterilmiş olan uygulamada PIR modülü ve arduino bağlantıları gösterilmektedir.



Şekil 31. PIR modülü ve arduino bağlantıları

Uygulamaya ait akış diyagramı aşağıda verildiği gibidir.



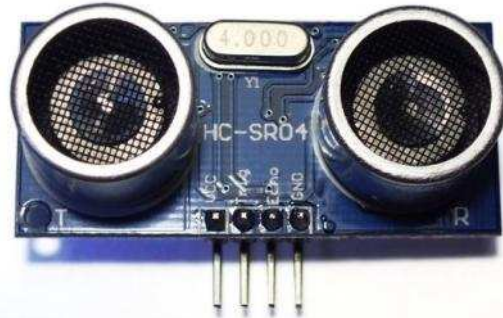
Şekil 32. Arduino ile PIR Sensörü Kontrolü Akış Diyagramı

Arduino uygulamasına ait uygulama kodları aşağıdaki gibidir.

```
#define PIR 12          // Sensör modülü data çıkışı
#define LED 13          // ledin bağlı olduğu pin
int input = 0;          // 1 veya 0 değerinin saklandığı değişken
void setup() {
    pinMode(PIR, INPUT);    // Sensör input
    pinMode(LED, OUTPUT);   // Led output
    digitalWrite(LED, LOW); // başlangıç olarak led 0
}
void loop(){
    input = digitalRead(PIR); // veri okunuyor
    if (input == HIGH)        // input digital 1 ise
        digitalWrite(LED, HIGH); // led yansın
    else
        digitalWrite(LED, LOW);  // 0 ise sönsün
}
```

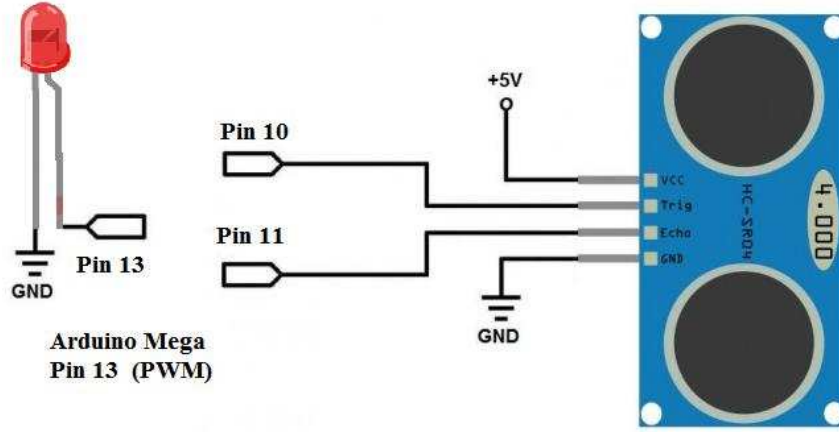
Örnek 6. Arduino İle HC-SRO4 Uzaklık Sensörü Kontrolü

Uzaklık sensörleri genellikle engel algılamak için kullanılırlar. Örneğin; yapacağınız robotun bazı engellere takılmaması için bir uzaklık ihtiyaç duyarsınız. En çok bilinen bir diğer uygulama da su seviyesinin ölçümüdür. Yapılan bu uygulamada HC-SRO4 uzaklık sensörü, engeli algılamak için ultrasonik ses dalgalarını kullanır. Üzerinde bulunan verici ultrasonik ses dalgası yayar. Bu dalga engele çarpıp yine HC-SRO4'ün üzerinde bulunan alıcıya ulaşır. Sinyalin vericiden çıkıp, alıcıya ulaşması için geçen süreye göre engele olan mesafe hesaplanır.

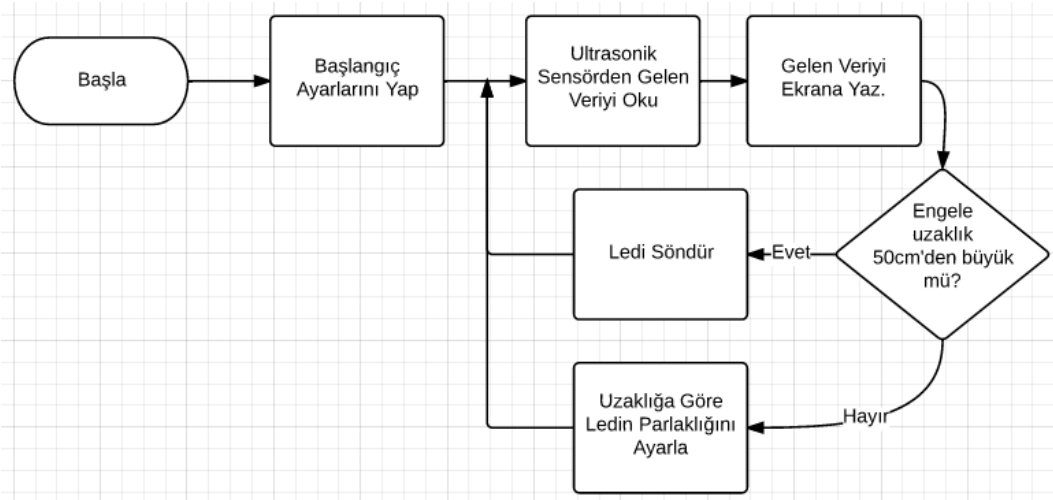


Şekil 33. HC-SRO4 ultrasonik sensör modülü

HC-SRO4 için hazırlanmış kütaphanenin örnek programını kullanarak, arduino'nun 13 nolu pinine bağladığımız ledin uzaklığa göre parlaklığını kontrol edeceğiz. Yapılan uygulamada engel yaklaştıkça ledin parlaklığının artmasını sağlanmaktadır. Kullanılacak ultrasonik sensöre ve LED'e ait bağlantılar aşağıda gösterilmektedir. Aşağıdaki şekil incelendiğinde LED'in 13 nolu pine bağlandığı görülmektedir. Buna ek olarak sensör modülü bağlantılarında Trig ve Echo uçlarının veri iletişimi için kullanıldığı görülmektedir. Modül ile devre iletişimde 10 ve 11.pinler kullanılmıştır.

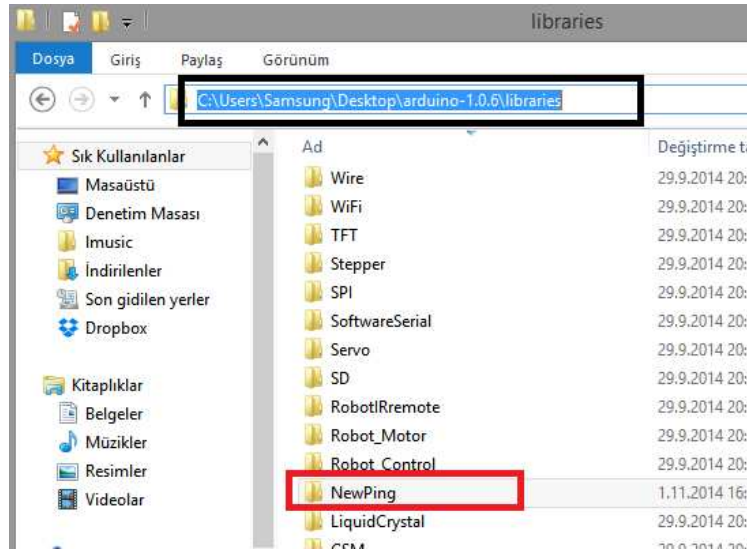


Şekil 34. HC-SR04 ultrasonik sensör modülü ve LED devre bağlantıları



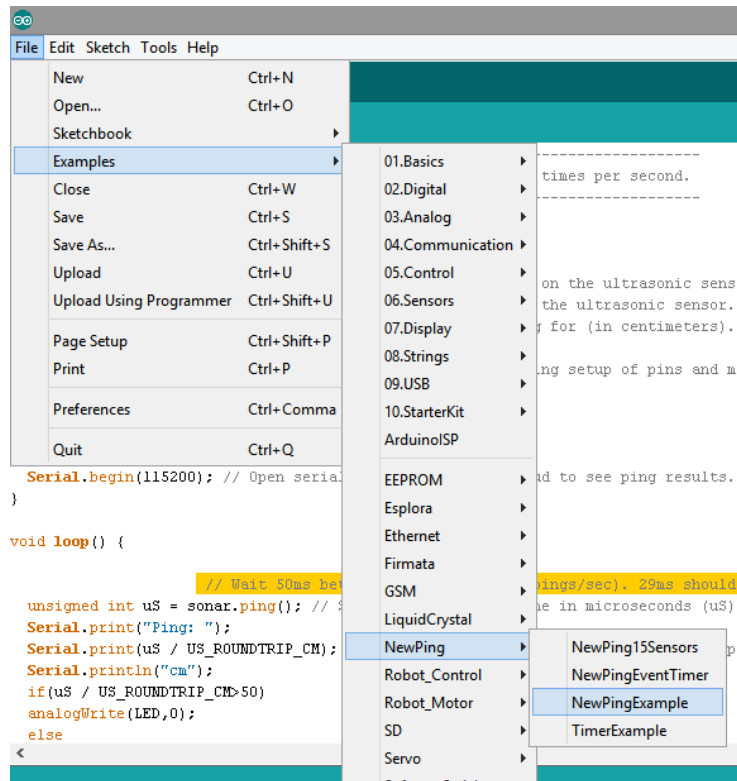
Şekil 35. Arduino ile HC-SR04 Uzaklık Sensörü Kontrolü Akış Diyagramı

Burada üzerinde durulması gereken bir diğer noktada kullanılan her modül için arduino ile haberleşmede uygun kütüphanenin projeye eklenmesinin gerektiğidir. Eklenen kütüphane dosyası sensörden gelen bilginin okunması görevini üstlenmektedir. Bunun temel nedeni kullan her sensörden gelen bilginin farklı bir yapıda olmasıdır. Kullanılan kütüphaneler uzmanlar tarafından hazırlanan ve sensörden gelen bilgiyi anlamlı hale getiren kod parçacıklarından meydana gelmektedir. HC-SR04 sensörüne ait arduino kütüphane dosyasına internette yapılan kısa bir arama ile yada <https://code.google.com/p/arduino-new-ping/> adresinden ulaşılabilir. İndirilen kütüphane dosyasının Arduino programına ait program dosya kütüphanesine (libraries) kopyalanması gerekmektedir.



Şekil 36. Arduino programı dosya kütüphanesi

Yapılan uygulamada arduino programı ile gelen örneklerden yararlanılmıştır, ilgili örneğe ulaşmak için file/examples/NewPing yolu takip edilmelidir.



Şekil 37. HC-SR04 Uzaklık Sensörü Örnek uygulaması

Uzaklık sensörü uygulamasına ait uygulama kodları aşağıda gösterildiği gibidir. Aynı uygulama kodlarına Örnek programlar aracılığı ile de ulaşılabilir. Açılan uygulama kodunda sensörden gelen bilginin seri iletişim yolu ile dışarı verildiği görülmektedir.

```

#include <NewPing.h>
#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 11 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

void setup() {
  Serial.begin(115200); // Open serial monitor at 115200 baud to see ping results.
}

void loop() {
  delay(50); // wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.
  unsigned int uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).
  Serial.print("Ping: ");
  Serial.print(uS / US_ROUNDTRIP_CM); // Convert ping time to distance in cm and print result (0 = outside set distance range)
  Serial.println("cm");
}

```

Örnekle hazır uygulamaya birkaç satır ekleyerek sensörden gelen bilgiye göre 13. Pin'e bağlı led'inde kolaylıkla yakılıp sökülmesi sağlanabilmektedir. Yeniden düzenlenen LED ile çalışan uygulama örneği aşağıda gösterildiği gibidir.

```

#include <NewPing.h>
#define TRIGGER_PIN 10 //Ultrasonik Sensörün Trigger Pini
#define ECHO_PIN 11 // Ultrasonik Sensörün Echo Pini
#define MAX_DISTANCE 255 // Engel algılama için Maksimum Mesage
#define LED 13
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
// Ultrasonik Sensör değişkeni oluşturuldu.

void setup() {
  pinMode(LED,OUTPUT); // Led Output Olarak ayarlandı

  Serial.begin(115200); // Seri haberleşme 'Baudrate' ayarlandı.
  /***** Serial Monitörde de Baudrate'i 115200 ayarlamayı unutmayın *****/
}

void loop() {

  unsigned int uS = sonar.ping();
  // Gönderilen ultrasonik dalganın ger dönüş süresinin değişkeni

  Serial.print("Ping: "); // Ekrana yazılıyor

  Serial.print(uS / US_ROUNDTRIP_CM); // Süre mesafe'ye çeviriliyor.
  Serial.println("cm");

  if(uS / US_ROUNDTRIP_CM>50) // Mesafe 50 cm fazla ise Ledi söndür
    analogWrite(LED,0);
  else // Eger değilse Ledin parlaklığının mesafeye göre değiştir.
    analogWrite(LED,255-5*uS / US_ROUNDTRIP_CM);
}

```

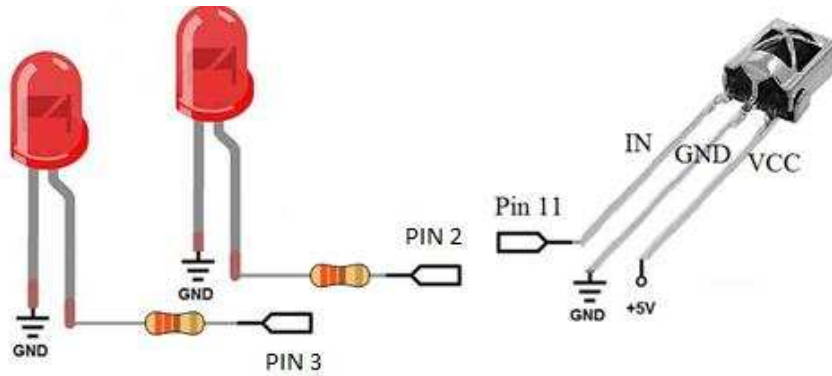
Örnek 7. Kızılötesi Sensör Uygulaması

Kızılötesi ışınlar insan gözünün algılayabileceği dalga boyu aralığı dışında çalışan elektromanyetik dalgalardır. Kızılötesi ışınların yoğun olarak kullanıldığı 2 alan vardır. Askeri olarak gece görüşünün sağlanması için kullanılırlar. İnsan da vücut sıcaklığından ötürü belirli bir kızılötesi ışın yayar. Diğer bir uygulaması da kısa mesafe haberleşmedir. Uzaktan kumandalar da kızılötesi ışın kullanılır. Kızılötesi ışınları algılamak için genellikle piyasada kolaylıkla bulunan CNY70, VS1838B gibi modellerdir. CNY70 sensörü daha çok robotik uygulamalarında, VS1838B sensörü ise arduino’ da hazır kütüphanesi olması için arduino uygulamalarında yaygın olarak kullanılmaktadır. Burada verici olarak basit bir oto teyp kumandasından yararlanılmıştır. Uygulamada kullanılan kumanda aşağıdaki şekilde görülmektedir. Buna ek olarak basit bir deneme ile evdeki TV kumandası ile de uygulama yapılabilmektedir.

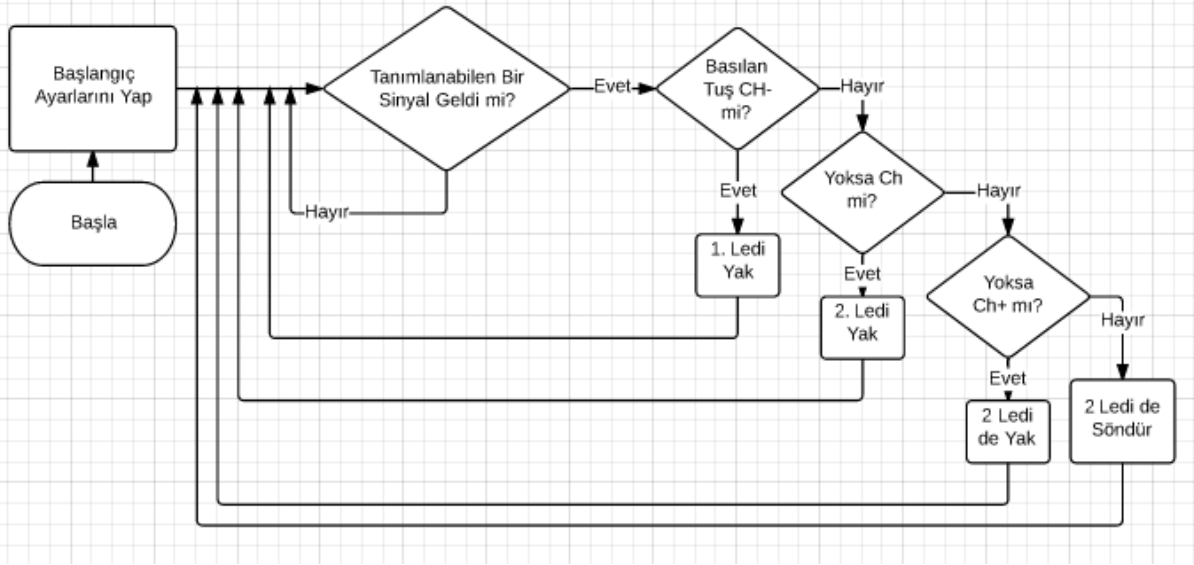


Şekil 38. Kızıl Ötesi Kumanda

Kullanılacak kızıl ötesi sensöre ve LED’e ait bağlantıları, akış diyagramı ve program kodları aşağıda gösterilmektedir.



Şekil 39. Kızılötesi Alıcı – Verici Bağlantıları



Şekil 40. Arduino ile Kızılötesi Alıcı Kumanda Kontrolü Akış Diyagramı

Arduino ile Kızılötesi Alıcı Kumanda Kontrolüne ait uygulama kodları aşağıdaki gibidir. Burada üzerinde durulması gereken bir diğer nokta her uygulama da olduğu gibi sensöre ait kütüphanenin uygulamaya eklenmesinin gerekliliğidir. Uygulamada kullanılan örneğe, arduino hazır uygulamalarından, “IRrcvDemo” ismi ile ulaşılabilir.

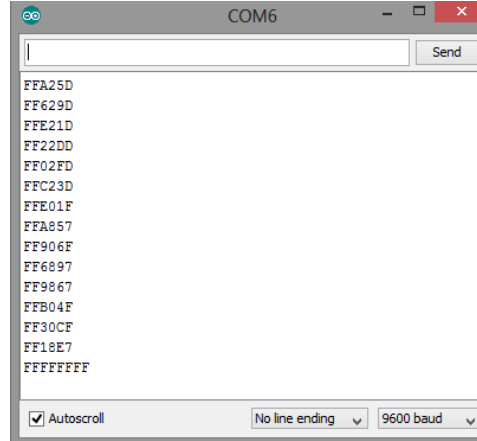
```

* IRremote: IRrecvDemo - demonstrates receiving IR codes with IRrecv
* An IR detector/demodulator must be connected to the input RECV_PIN.
* Version 0.1 July, 2009
* Copyright 2009 Ken Shirriff
* http://arcfn.com
*/
#include <IRremote.h>
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
  delay(100);
}

```

Bu uygulamada sensörden gelen bilgi seri haberleşme yoluyla bilgisayara gönderilmektedir. Aşağıda ki şekilde bir seriport izleyici program aracılığı ile arduino’dan gelen bilgi okunabilmektedir.



Şekil 41. Seriport izleme programı

Okunan bilgiyi seri haberleşme yolu ile gönderen uygulama kodu aşağıdaki gibidir.

```
#include <IRremote.h> // Kütüphane Eklendi
int RECV_PIN = 11; // 11 pin VS1838B IN pini
IRrecv irrecv(RECV_PIN); // Alıcı Tanımlandı
decode_results results; // alıcıya gelen sinyalin dijital karşılığı
void setup()
{
  Serial.begin(9600); // Ekrana yazmak için
  // Seri haberleşme başlatıldı
  irrecv.enableIRIn(); // IR haberleşme başlatıldı.
  pinMode(2,OUTPUT); // 2. ve 3. pindeki Led ayarlandı
  pinMode(3,OUTPUT);
}
void loop() {
  if (irrecv.decode(&results)) { // sinyal algılanırsa
    Serial.println(results.value, HEX); // ekrana hex karşılığını yaz

    if(results.value==0xFFA25D) // CH- ise 2. Led yansın
      digitalWrite(2,HIGH);
    else if(results.value==0xFF629D) // CH ise 3. Led yansın
      digitalWrite(3,HIGH);

    else if(results.value==0xFFE21D){ // CH ise 2. ve 3. Led yansın
      digitalWrite(2,HIGH);
      digitalWrite(3,HIGH);
    }

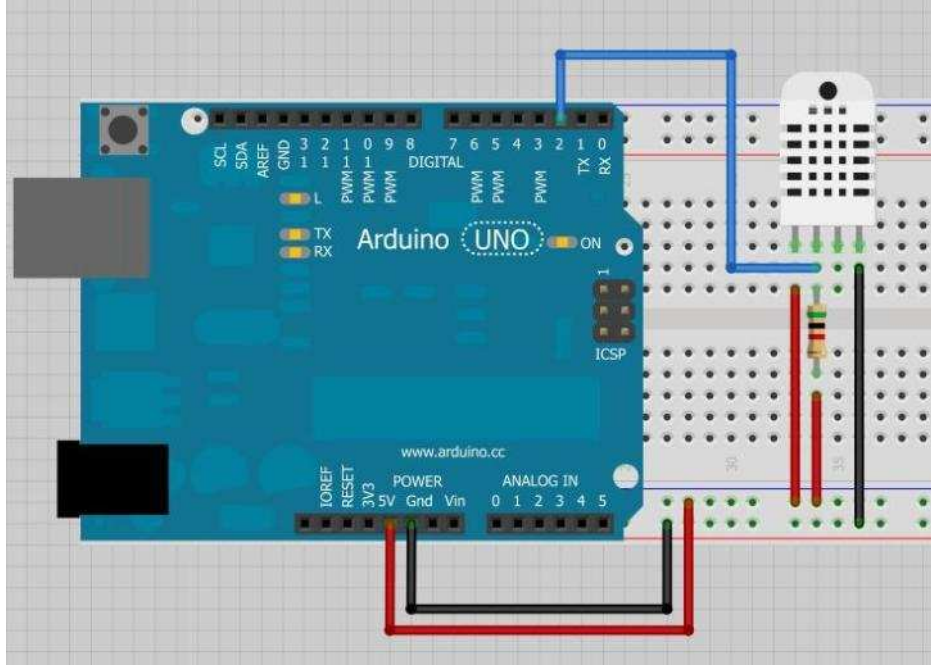
    else { //Diğer butonlar ise Ledler sönsün
      digitalWrite(2,LOW);
      digitalWrite(3,LOW);
    }

    irrecv.resume(); // Yeni bir değer yeniden veri alınacak
  }
  delay(100); // Gelen sinyaller arasında karışıklık olmaması
               // için 100ms bekleniyor
}
```

Örnek 8. DHT11 Sensörü kullanılarak ortamın nem ve sıcaklığının ölçülmesi

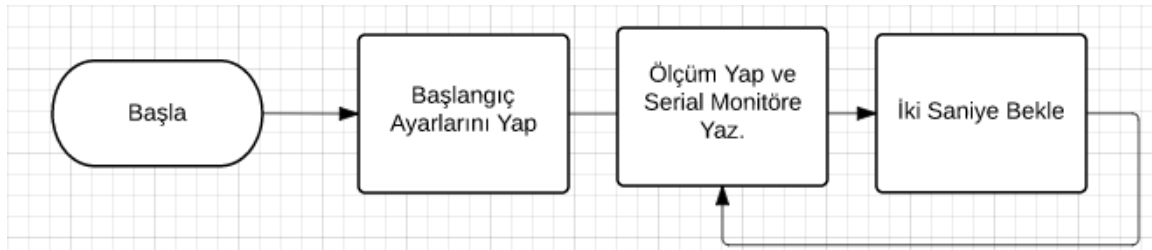
Yapılan bu uygulamada arduino ile DHT11 sıcaklık-nem sensörünü kullanarak bulunduğumuz odanın sıcaklık ve nemini ölçeceğiz. DHT11 sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimdir. Yüksek güvenilirliktedir ve

uzun dönem çalışmalarda dengelidir. 8 bit mikroişlemci içerir, hızlı ve kaliteli tepki verir. 0 ile 50°C arasında 2°C hata payı ile sıcaklık ölçen birim, 20-90% RH arasında 5% RH hata payı ile nem ölçmektedir. Sensör ve çıkış bilgisi ile ilgili detaylı bilginin internetten araştırılmasında yarar vardır.



Şekil 42. Arduino ile DHT11 Sensörü Kontrol Devresi

DHT11 doğrudan dijital bilgi üreten kompleks ileri seviye bir sensördür. Uygulamada sensörün doğru şekilde çalışması için 4.7k'lık bir direncin gösterildiği gibi bağlanmasında yarar vardır. Bu bağlantı tipi literatüre Pull-Up olarak geçmiştir. Kullanımına ait geniş bilgiye internet üzerinden ulaşılabilir. Bağlanan direnç ile sensör çıkışındaki gürültünün azaltılması amaçlanmaktadır. DHT11 sensörü bağlantısı yapılırken şekilde de görüldüğü gibi 3 nolu uç (NC) boş bırakılmalıdır. Eğer uygulamada sensöre modül olarak sahipseniz, modülün NC çıkışa sahip olmadığını görebilirsiniz. Uygulamaya ait akış diyagramı şekilde görüldüğü gibidir.



Şekil 43. Arduino ile DHT11 Sensörü Kontrol Devresi Akış Diyagramı

Uygulamada kullanılan program kodları aşağıda gösterildiği gibidir. Uygulama kodları incelendiğinde DHT11 sensörüne ait kütüphanenin de uygulamaya eklendiği görülmektedir. Uygulamada kullanılan kodlara program örnek kodlarından da ulaşılabilir. Örnek kodlara seri haberleşme bölümü de eklenerek nem sıcaklık bilgisinin bilgisayara da gönderilmesi sağlanmıştır.


```

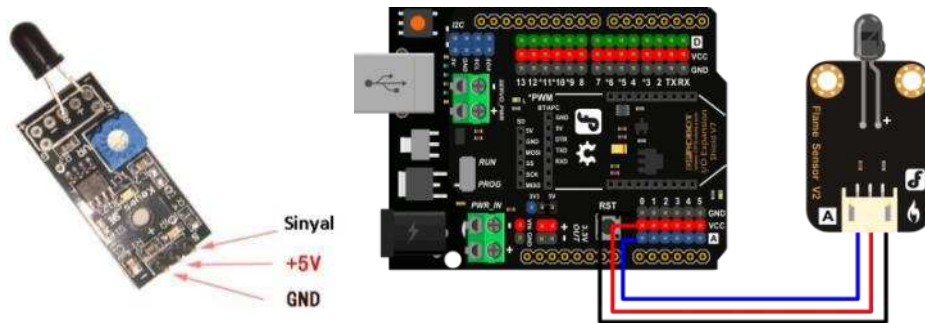
#include "DHT.h" // DHT sensörü için yazılan kütüphane
#define DHTPIN 2 // DHT'nin data pinini arduino'ya bağladığımız pin
#define DHTTYPE DHT11 //DHT21 //DHT22
DHT dht(DHTPIN, DHTTYPE); // normal hızda(16Mhz) çalışan arduinolara için
// sensörümüz tanımlandı
//DHT dht(DHTPIN, DHTTYPE, 30); // yüksek hızda arduino'lar için
void setup() {
  Serial.begin(9600); // Seri haberleşme
  Serial.println("DHT11 Haberleşme Başladı!");
  dht.begin(); // sensörden veri alımı başlatıldı
}
void loop() {
  delay(2000); // Her iki saniyede bir ölçüm yapılacaktır
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  // Veri okumada sıkıntı var mı kontrol ediliyor
  if (isnan(h) || isnan(t)) { // eğer h, ve t değişkenleri sayı ifade etmiyorsa
    Serial.println("Sensörden veri okunamadı!");
    return; // Veri okunamadığı için başa dönüldü.
  }
  Serial.print("Nem: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Sıcaklık: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print("\n");
}

```

Örnek 9. Arduino kullanılarak alev sensörünün kullanılması

Yangın sensörü, alev oluşması durumunda ortama yayınlan ultraviyole ışınları algılayarak sinyal üreten bir sensördür. Dijital çıkış üreten sensörümüz 40-50 cm kadar kaynaklardaki alevi algılayabilmektedir. Yaygın olarak yangın söndüren robotlarda kullanılır. Alev algılanması durumunda çıkış 0V (0), yoksa 5V (1) çıkış verir.

Sensöre ait devre bağlantıları aşağıda gösterildiği gibidir.



Şekil 44. Alev sensörü devre bağlantıları

Uygulamaya ait uygulama kodları aşağıdaki gibidir. Uygulama kodları incelenirse sensör için ekstra bir kütüphane eklenmediği dahası programın buton ile LED yakma uygulaması ile neredeyse aynı görülmektedir. Bunu nedeni sensörün çıkışının bir butonunki gibi basit var/yok bilgisinin gelmesidir.

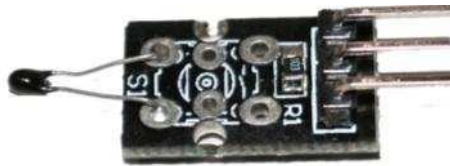
```

const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;      // the number of the LED pin
// variables will change:
int buttonState = 0;        // variable for reading the pushbutton status
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}

```

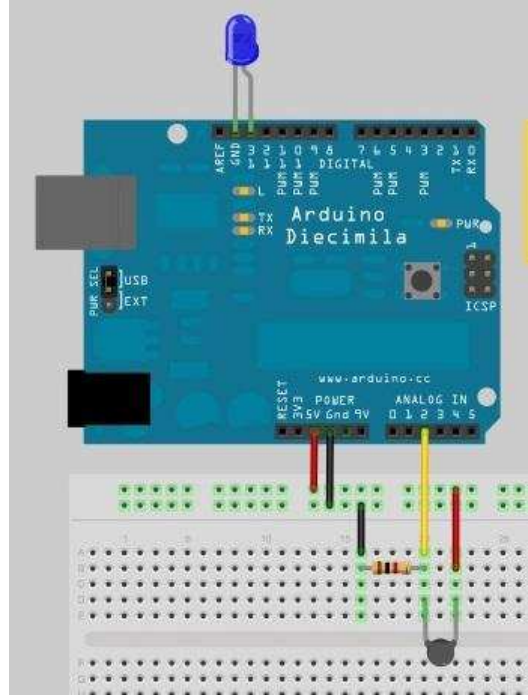
Örnek 10. Sıcaklık sensörü uygulaması

Sıcaklık basit sıcaklığa tepki gösteren dirençler ile kolaylıkla algılanabilmektedir. Bu dirençler ile yapılan uygulamalarda, direncin ortamında sıcaklığına göre değişim göstermesi ile direncin bağlı olduğu arduino pinindeki gerilimde $V=I.R$ bağıntısına bir miktar değişmektedir. Buda pin'den analog bilgi okunması ile sayısal bir değere dönüştürülmektedir. Burada üzerinde durulması gereken önemli bir nokta uygulamada pindeki bilgiyi okumak için bir ADC' nin yeterli olmasıdır. ADC konusu Analog giriş uygulamasında detaylı olarak incelenmiştir. Sıcaklık ile direnci değişen sensörlerin yanında sıcaklık bilgisine göre kompleks dijital çıkış üreten sensörde mevcuttur. Piyasada en çok kullanılan tipleri D1621 ve DHT11 sensörleridir ve uygulamalarda uygun kütüphaneye sahip olmak gerekir. DHT11 sensörüne ait bir uygulama önceki bölümlerde incelenmiştir. İncelenen bu uygulamada ise sıcaklık ile direnci değişen bir eleman olan termistör aracılığı uygulama geliştirilmiştir.



Şekil 45. Sıcaklık Sensörü (Termistör Modülü)

Uygulamada sensör ortam sıcaklığını gerçek zamanlı olarak algılamakta ve arduino kartın analog girişlerine bir gerilim bilgisi göndermektedir. Yapılan uygulama ile sensörden gelen bilgi ile bahçe otomasyonu gibi uygulamalar geliştirilebilmektedir. Buradaki örnekte sıcaklık bilgisine göre bir LED'in yakılıp söndürülmesi amaçlanmıştır. Uygulamada kullanılan sensör modülünün bağlantıları aşağıdaki şekilde gibidir.



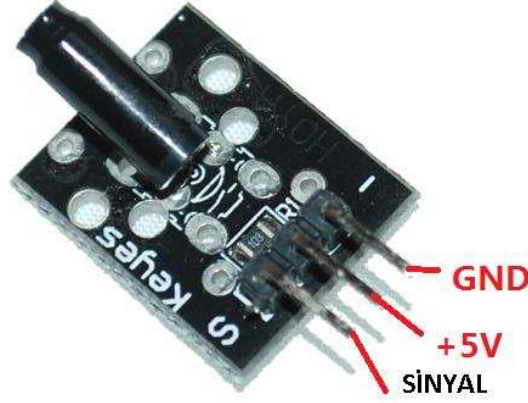
Şekil 46. Termistör uygulaması bağlantı şeması

Uygulamaya ait program kodları aşağıdaki gibidir.

```
int sensorPin = A2;    // select the input pin for the potentiometer(or termistor)
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
  Serial.println(sensorValue, DEC);
}
```

Örnek 11. Arduino ile titreşim sensörü uygulaması

Ortamda (sensörün bağlandığı gövdede) meydana gelebilecek ani hareketlerin algılanmasında kullanılan titreşim sensörleri piezo elektrik malzeme kullanılarak, yada iletken bir sıvının elektriksel kutuplar içeren bir tüp içindeki hareketi incelenerek yapılabilmektedir. Buna ek olarak sensörlere fazladan bir rezonans kütlesi eklenerek duyarlılığının arttırıldığı sensör modüllerinde görülmektedir. Bu prensip ile çalışan jiroskop sensörleride (açısal hareketin algılanması) uygun fiyatlara piyasadan bulunmaya başlanmıştır.



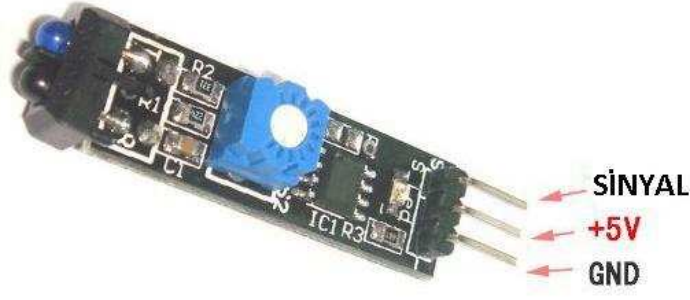
Şekil 47. Titreşim sensörü

Burada yapılan uygulamamızda iletken bir sıvının titreşim etkisine göre elektriksel kutuplar içeren bir tüp içindeki hareketine göre çıkış üreten bir sensör kullanılmaktadır. Modül halinde olan bu sensör piyasadan uygun fiyatlara kolaylıkla temin edilebilmektedir. Uygulama kodları aşağıdaki gibidir.

```
int Led=13;//define LED interface
int Shock=3//define vibration sensor interface
int val;//define digital variable val
void setup()
{
  pinMode(Led,OUTPUT);//define LED as output
  pinMode(Shock,INPUT);//define shock as input
}
void loop()
{
  val=digitalRead(Shock);//
  if(val==HIGH)//
  {
    digitalWrite(Led,LOW);
  }
  else
  {
    digitalWrite(Led,HIGH);
  }
}
```

Örnek 12. Engel algılayan sensör

Engel algılayan sensör aslında yaklaşım sensörlerinin bir çeşididir. Endüstriyel kontrol uygulamalarında daha çok endüktif ve kapasitif tip sensörler ile bu işlem yapılmaktadır. Fakat bu tip sensörlerin maliyeti yüksek ve yapısı daha karmaşıktır. Oysa bunun yanında aynı işlem kızıl ötesi ışık kullanılarak ta yapılabilir. Bu tip sensörlerde de diğer tiplerde olduğu bir alıcı ve bir verici eleman mevcuttur.



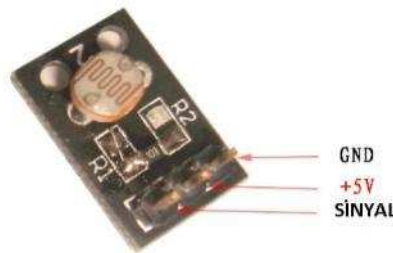
Şekil 48. Kızılötesi Engel Algılayan Sensör

Kızıl ötesi sensörün önüne bir engel geçtiğinde alıcı ile verici arasında iletişim kesildiği için sensörün önünden geçen cisim algılanabilmektedir. Yukarıdaki çalışma prensibi incelendiğinde sensör çıkışında var/yok (1/0) şeklinde bir sinyal elde edildiği görülmektedir. Bu nedenle sensörün kullanılması için özel bir kütüphaneye ihtiyaç duyulmamaktadır. Sensöre ait bağlantılar yukarıda gösterildiği gibidir. Uygulama kodları aşağıda gösterilmiştir.

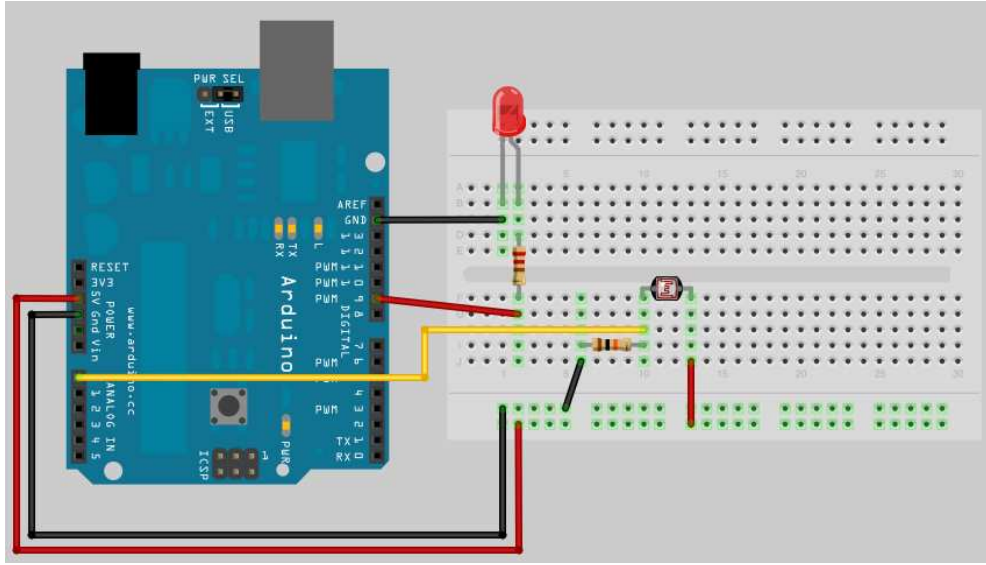
```
int Led=13;
int Shock=3
int val;
void setup()
{
  pinMode(Led,OUTPUT);
  pinMode(Shock,INPUT);//
}
void loop()
{
  val=digitalRead(Shock);
  if(val==HIGH)/
  {
    digitalWrite(Led,LOW);
  }
  else
  {
    digitalWrite(Led,HIGH);
  }
}
```

Örnek 13. Işığa duyarlı sensör uygulaması

Işık algılayan sensörler (LDR) de çoğunlukla ışığın etkisi ile direnci değişen bir eleman kullanılmaktadır. İlgili sensörün bir bacağı denetleyicinin analog giriş pinlerinden birine bağlanmaktadır. Sensör ve denetleyici bağlantıları aşağıdaki şekillerde gösterilmektedir.



Şekil 49. Işığa duyarlı sensör (LDR Modülü)



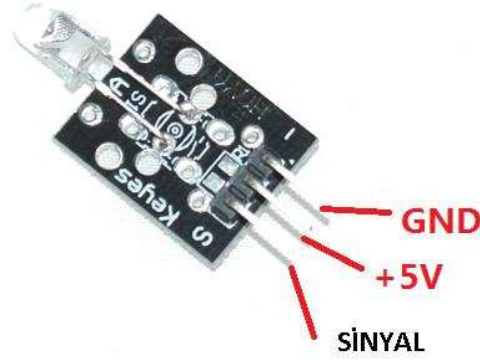
Şekil 50. Işık algılayan sensör (LDR) ve Denetleyici bağlantıları

Diğeri ise +5V beslemeye bağlanmaktadır. Sensörünün ışığın şiddetine göre direncinin değişmesi ile bir pin üzerine değilken bir gerilim akmaktadır. Gerilim miktarının ölçülmesi ile ışığın şiddetinin bir göstergesi olan bilgi elde edilmektedir. Sensörün çalışmasında kullanılan örnek bir uygulama aşağıdaki gibidir.

```
int sensorPin = A5;    // select the input pin for the potentiometer
int ledPin = 13;       // select the pin for the LED
int sensorValue = 0;   // variable to store the value coming from the sensor
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
  Serial.println(sensorValue, DEC);
}
```

Örnek 14. Kızıl ötesi sensör kullanılarak çoklu verinin gönderilmesi

Kızılötesi verici ile arduino üzerinde yazılacak programa bağlı olarak sinyal gönderilebilmektedir. Bugün bu teknoloji elektronik cihazların uzaktan kontrol edilmesinde sıkça kullanılmaktadır. Modüler olarak temin edilebilen bu sensör yardımı ile birden çok verinin kodlanıp gönderilmesi mümkündür. Gönderilen RF teknolojilerine göre daha yavaş kalırken diğer bir açıdan maliyet açısından çok daha avantajlıdır. Uygulamada kullanılan modül ve bağlantıları aşağıdaki şekilde gösterilmektedir. Sensör ile gönderilen veri çoğaldıkça program yükü de artmaktadır. İncelenen uygulamada bir kumandaya ait temel fonksiyon tuşları kodlanmaya çalışılmıştır.



Şekil 51. Kızıl Ötesi Sensör

```
const int analogInPin = A0;
int sensorValue = 0;
int KEY ;
int NEWKEY ;
int OLDKEY = 0;
int OK = 0 ;
int FLAG = 0;
int coun = 0;
int coun1 = 0;
int look = 0;
int chk[16]
{0,100,260,385,482,530,587,630,660,688
,714,735,753,769,784,800};//
set array comparision value
//*****
*****
#include
long irKeyCodes[16] = {
0x00000000, 0x40BFF807, 0x40BF7887,
0xC03FC03F, // correspond
to Keypad button '1', '2', '3', '4'
0xC03F40BF, 0x40BF00FF, 0x00000000,
0x00000000, // correspond
to Keypad button '5', '6', '7', '8'
0x00000000, 0x00000000, 0x00000000,
0x00000000, // correspond to
Keypad
button '9', '10', '11', '12'
0x80FFC13E, 0x80FFE11E, 0x80FFD12E,
0x80FFF10E // correspond
to Keypad button '13', '14',
'15', '16'
};
IRsend irsend;
//*****
*****
void setup() {
Serial.begin(9600); //Set Baud Rate:
}
void loop() {
sensorValue = analogRead(analogInPin);
delay(15);
aaa:
if (FLAG == 0)
{
if (chk[0]<=sensorValue && sensorValue
{
KEY = 1 ;
}
else if (chk[1]<=sensorValue &&
sensorValue
{
```

```
KEY = 2 ;
}
else if (chk[2]<=sensorValue &&
sensorValue
{
KEY = 3 ;
}
else if (chk[3]<=sensorValue &&
sensorValue
{
KEY = 4 ;
}
else if (chk[4]<=sensorValue &&
sensorValue
{
KEY = 5 ;
}
else if (chk[5]<=sensorValue &&
sensorValue
{
KEY = 6 ;
}
else if (chk[6]<=sensorValue &&
sensorValue
{
KEY = 7 ;
}
else if (chk[7]<=sensorValue &&
sensorValue
{
KEY = 8 ;
}
else if (chk[8]<=sensorValue &&
sensorValue
{
KEY = 9 ;
}
else if (chk[9]<=sensorValue &&
sensorValue
{
KEY = 10 ;
}
else if (chk[10]<=sensorValue &&
sensorValue
{
KEY = 11 ;
}
else if (chk[11]<=sensorValue &&
sensorValue
{
KEY = 12 ;
}
else if (chk[12]<=sensorValue &&
sensorValue
{
```



```

{
KEY = 13 ;
}
else if (chk[13]<=sensorValue &&
sensorValue
{
KEY = 14 ;
}
else if (chk[14]<=sensorValue &&
sensorValue
{
KEY = 15 ;
}
else if (chk[15]
{
KEY = 16 ;
}
else if (sensorValue>1000 )
//
{
KEY = 0;
}
}
NEWKEY = KEY; //save
value of KEY to NEWKEY
if ( NEWKEY == OLDKEY) // if
NEWKEY == OLDKEY then OK + 1
{
OK++;
}
else
{
OK = 0; // if not,
then OK=0
}
OLDKEY = NEWKEY;
if(OK>=5) // if OK
=5 print the value
{
Serial.print("KEY = ");
Serial.println(NEWKEY);
Serial.println(sensorValue);
Serial.print("look = ");
Serial.println(look);
if (NEWKEY !=0)
{
switch(NEWKEY){
case 1:
FLAG =1;
break;
case 2:
FLAG =1;
break;
case 3:
FLAG =1;
break;
case 4:
FLAG =1;
break;
case 5:

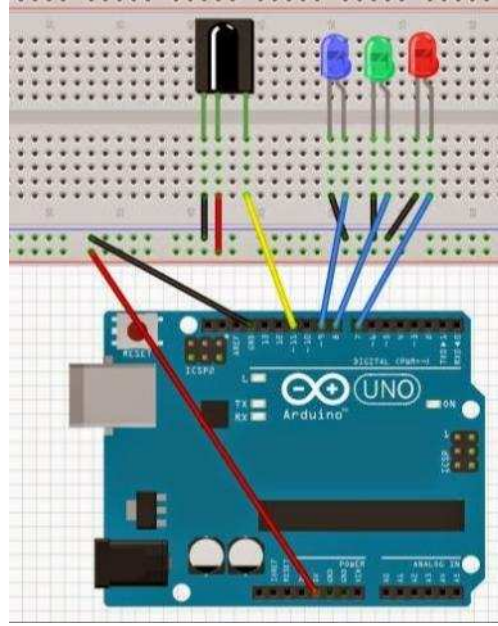
```

```

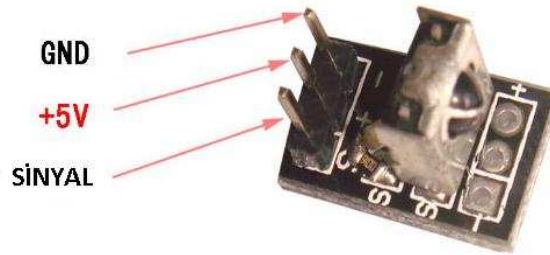
FLAG =1;
break;
case 6:
FLAG =1;
break;
case 7:
FLAG =1;
break;
case 8:
FLAG =1;
break;
case 9:
FLAG =1;
break;
case 10:
FLAG =1;
break;
case 11:
FLAG =1;
break;
case 12:
FLAG =1;
break;
case 13:
FLAG =1;
break;
case 14:
FLAG =1;
break;
case 15:
FLAG =1;
break;
case 16:
FLAG =1;
break;
}
if(NEWKEY>0)
{
coun1++;
}
if(coun1>5)
{
coun1=0;
irsend.sendNEC(irKeyCodes[NEWKEY],
32);
look++;
}
if (sensorValue>950)
{
coun++;
}
if(coun>1)
{
FLAG =0;
coun = 0;
goto aaa;
}
}
}

```

Yukarıdaki örnek kodlar ile kızıl ötesi çoklu bilginin gönderilmesi üzerinde durulmuştur bu bölümde ise alıcı kısmında kullanılan kızıl ötesi alıcı sensör devresi üzerinde durulmuştur. Devreye ait bağlantılar ve alıcı sensör bağlantıları aşağıdaki gibidir.



Şekil 52. Kızılötesi alıcı devresi bağlantıları



Şekil 53. Kızılötesi alıcı sensör bağlantıları

Verici devresi uygulama kodları aşağıdaki gibidir.

```
#include <IRremote.h>
int RECV_PIN = 11;
int RELAY_PIN = 4;
IRrecv irrecv(RECV_PIN);
decode_results results;
// Dumps out the decode_results
// structure.
// Call this after IRrecv::decode()
// void * to work around compiler
// issue
//void dump(void *v) {
//    decode_results *results =
//    (decode_results *)v
//}
void dump(decode_results *results) {
    int count = results->rawlen;
    if (results->decode_type == UNKNOWN)
    {
        Serial.println("Could not decode
        message");
    }
    else {
        if (results->decode_type == NEC) {
            Serial.print("Decoded NEC: ");
        }
        else if (results->decode_type ==
        SONY) {
```

```
        Serial.print("Decoded SONY: ");
    }
    else if (results->decode_type ==
    RC5) {
        Serial.print("Decoded RC5: ");
    }
    else if (results->decode_type ==
    RC6) {
        Serial.print("Decoded RC6: ");
    }
    Serial.print(results->value, HEX);
    Serial.print(" (");
    Serial.print(results->bits, DEC);
    Serial.println(" bits)");
}
Serial.print("Raw (");
Serial.print(count, DEC);
Serial.print("): ");

for (int i = 0; i < count; i++) {
    if ((i % 2) == 1) {
        Serial.print(results-
        >rawbuf[i]*USECPERTICK, DEC);
    }
    else {
```

```

        Serial.print(-(int)results-
>rawbuf[i]*USECPERTICK, DEC);
    }
    Serial.print(" ");
}
Serial.println("");
}
void setup()
{
    pinMode(RELAY_PIN, OUTPUT);
    pinMode(13, OUTPUT);
    Serial.begin(9600);
    irrecv.enableIRIn(); // Start the
receiver
}
int on = 0;
unsigned long last = millis();

```

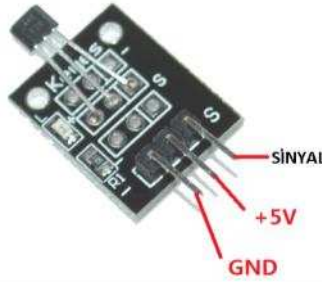
```

void loop() {
    if (irrecv.decode(&results)) {
        // If it's been at least 1/4
second since the last
        // IR received, toggle the relay
        if (millis() - last > 250) {
            on = !on;
            digitalWrite(RELAY_PIN, on ?
HIGH : LOW);
            digitalWrite(13, on ? HIGH :
LOW);
            dump(&results);
        }
        last = millis();
        irrecv.resume(); // Receive the
next value
    }
}

```

Örnek 15. Manyetik alan (Hall etkili) sensör uygulaması

Temelde bir kontak gibi çalışan bu sensör modülü bulunduğu ortamdaki manyetik alandan etkilenecek kontaklarını açıp kapamamaktadır.



Şekil 54. Manyetik Etki algılayan sensör

Yapılan bu tür uygulamalarda yoğun olarak kullanılan bir diğer sensör tipide reed anahtarıdır. Fakat mekanik bir aktivasyon mantığı ile çalışan bu eleman kolaylıkla deforme olabilmektedir. Burada kullanılan modül ise entegre temelli çalıştığı için bu problem ile daha az karşılaşılmaktadır. Uygulamada kullanılan deney kodları aşağıdaki gibidir.

```

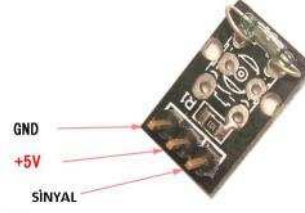
int Led=13;
int SENSOR=3;
int val;
void setup()
{
    pinMode(Led,OUTPUT);
    pinMode SENSOR,INPUT);
}
void loop()
{
    val=digitalRead(SENSOR);
    if(val==HIGH)
    {
        digitalWrite(Led, HIGH);
    }
    Else
    {
        digitalWrite(Led, LOW);
    }
}

```

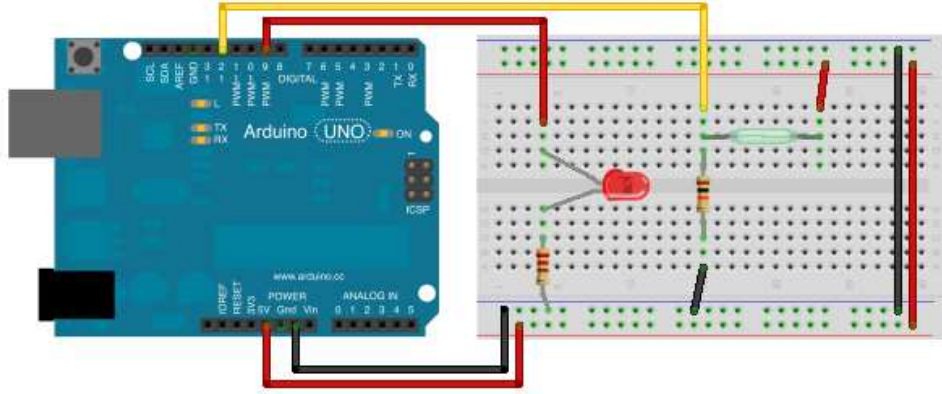
Örnek 15. Reed Anahtar kullanılarak manyetik alan sensör uygulaması

Manyetik alan algılamada kullanılan bu sensör havası alınmış cam tüp içerisine manyetik alandan etkilenen bir anahtar mekanizmasının yerleştirilmesi ile elde edilmektedir. Manyetik alandan etkilenen metal kontak mekanizması sayesinde manyetik etki dijital bir çıkış bilgisi ölçülebilmektedir.

Sensör bağlantıları, devre bağlantıları ve uygulama kodları aşağıdaki gibidir.



Şekil 55. Reed Kontakt Modülü

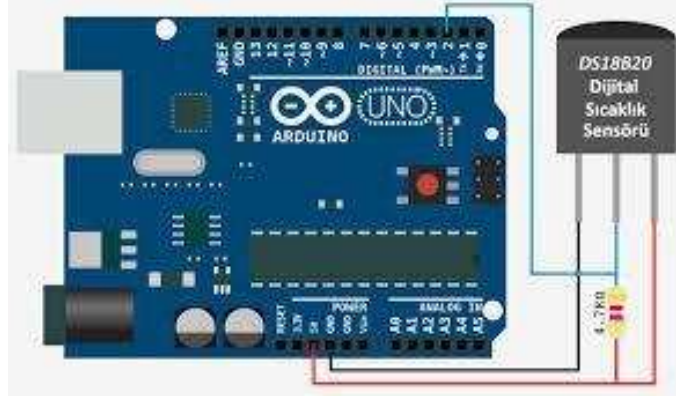


Şekil 56. Reed kontakt modülü bağlantıları

```
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;     // the number of the LED pin
// variables will change:
int buttonState = 0;       // variable for reading the pushbutton status
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Örnek 16. DS18820 kullanılarak sıcaklık ölçümü

Sıcak ölçümünde çoğunlukla sıcaklık etkisi ile direnci değişen elemanlar kullanıldığından daha önce bahsedilmiştir. Bu uygulamamızda ise sıcaklık etkisi ile 8 bitlik dijital bir bilgi çıkışı veren DS18820 sensörünün kullanımı üzerinde durulmuştur. Uygulamada kullanılan sensör ve devre bağlantısı aşağıdaki şekilde gösterilmektedir.



Şekil 57. DS18820 sıcaklık sensörü devre bağlantıları

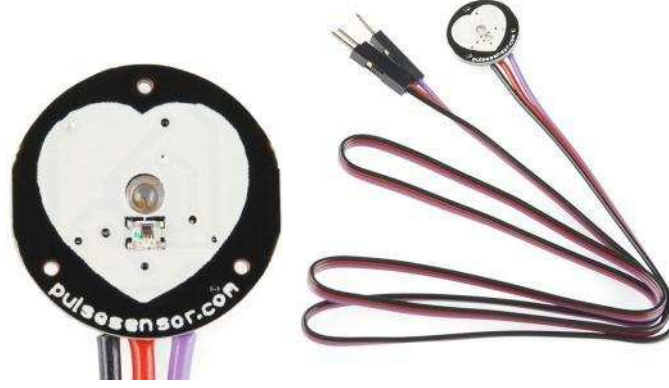
Uygulama kodları aşağıdaki gibidir. Uygulama kodları incelendiğinde elde edilen sensör bilgisinin sensör kütüphanesi olmadan alındığı görülebilmektedir. İlgili sensör kütüphanesinin projeye eklenmesi ile program yükü oldukça hafiflemektedir.

```
#include
/* DS18S20 Temperature chip i/o
*/
OneWire ds(10); // on pin 10
void setup(void) {
  // initialize inputs/outputs
  // start serial port
  Serial.begin(9600);
}
void loop(void) {
  byte i;
  byte present = 0;
  byte data[12];
  byte addr[8];
  int Temp;
  if ( !ds.search(addr)) {
    //Serial.print("No more addresses.n");
    ds.reset_search();
    return;
  }
  Serial.print("R="); //R=28 Not sure
  what this is
  for( i = 0; i < 8; i++) {
    Serial.print(addr[i], HEX);
    Serial.print(" ");
  }
  if ( OneWire::crc8( addr, 7) !=
    addr[7]) {
    Serial.print("CRC is not valid!n");
    return;
  }
  if ( addr[0] != 0x28) {
    Serial.print("Device is not a DS18S20
    family device.n");
    return;
  }
  ds.reset();
  ds.select(addr);
```

```
ds.write(0x44,1); // start
conversion, with parasite power on at
the end
delay(1000); // maybe 750ms is
enough, maybe not
// we might do a ds.depower() here,
but the reset will take care of it.
present = ds.reset();
ds.select(addr);
ds.write(0xBE); // Read Scratchpad
Serial.print("P=");
Serial.print(present, HEX);
Serial.print(" ");
for ( i = 0; i < 9; i++) { //
we need 9 bytes
data[i] = ds.read();
Serial.print(data[i], HEX);
Serial.print(" ");
}
Temp=(data[1]<<8)+data[0]; //take the
two bytes from the response relating
to
temperature
Temp=Temp>>4; //divide by 16 to get
pure celcius readout
//next line is Fahrenheit conversion
Temp=Temp*1.8+32; // comment this line
out to get celcius
Serial.print("T="); //output the
temperature to serial port
Serial.print(Temp);
Serial.print(" ");
Serial.print(" CRC=");
Serial.print( OneWire::crc8( data, 8),
HEX);
Serial.println();
}
```

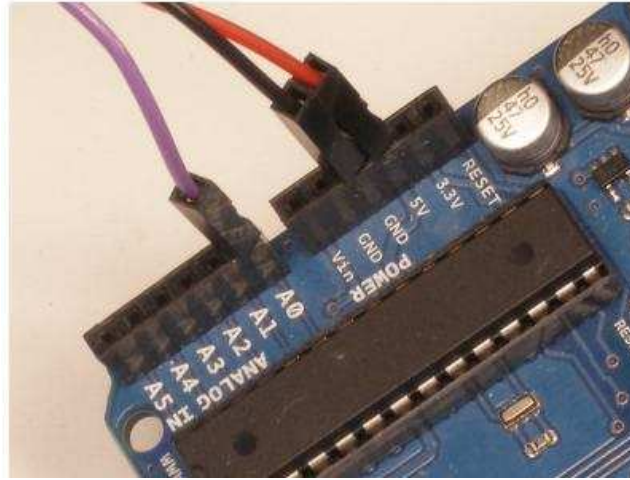
Örnek 17. Nabız Sensörü

Bu nabız sensörü ile insanların kalp aktivitelerini, endişe seviyelerini takip edebilir veya kalp atışınızla yanıp sönen bir tişört yapabilirsiniz. Bu nabız sensörü Arduino modülleri ile doğrudan çalıştırılabilmektedir. Sensör kalp ritmini kulak memesinden, parmağınızdan yada doğrudan göğüs kafesinden alabilmektedir. Uygulamada kullanılan kalp atış sensörü aşağıdaki şekilde gösterilmektedir.



Şekil 58. Nabız Sensörü

Uygulama da kullanılan sensör kalp atış bilgisinin analog bir çıkış üreterek ölçmektedir. Analog bilginin okunmasında A0 Pini kullanılmıştır. Sensörün kalp atışını algılaması için parmak ucuna yada kulak memesine temas etmesi gerekmektedir. Uygulanması ve programlanması ile ilgili detaylı bilgilere, <http://www.robotshop.com/media/files/ZIP/documentation-sen-11574.zip> linkinden ulaşılabilir.



Şekil 59. Kalp Atış Sensörü Bağlantıları

Örnek uygulama kodları aşağıdaki gibidir. Daha detaylı bilgi almak ve işlevsel programlar yazmak için sensör dökümanının incelenmesi faydalı olacaktır.

```
int ledPin=13;
int sensorPin=0;

double alpha=0.75;
int period=20;
double change=0.0;

void setup()
{
  pinMode(ledPin,OUTPUT);
}

void loop()
{
  static double oldValue=0;
  static double oldChange=0;
  int rawValue=analogRead(sensorPin);
  double value=alpha*oldValue+(1-alpha)*rawValue;
  change=value-oldValue;

  digitalWrite(ledPin,(change<0.0&&oldChange>0.0));

  oldValue=value;
  oldChange=change;
  delay(period);
}
```

Aşağıda verilmiş olan bir diğer uygulamada da sensörden gelen bilgiye göre bir LED'in yakılıp söndürülmektedir.

```
// Define these at the top of your sketch.
#define LEDBLINK_PIN 13 // I/O pin connected to the LED.
#define LEDBLINK_MS 1000 // Blink rate (in milliseconds).

void setup()
{
  // For ledBlink(), set LEDBLINK_PIN to output.
  pinMode(LEDBLINK_PIN, OUTPUT);
}

void loop()
{
  // Blink the LED to let the user know we are still alive.
  ledBlink();
  // Do something.
}

//
// LED Heartbeat routine by Allen C. Huffman (www.appleause.com)
//
void ledBlink()
{
  static unsigned int ledStatus = LOW; // Last set LED mode.
  static unsigned long ledBlinkTime = 0; // LED blink time.

  // LED blinking heartbeat. Yes, we are alive.
  // For explanation, see:
  // http://playground.arduino.cc/Code/TimingRollover
  if ( (long)(millis()-ledBlinkTime) >= 0 )
  {
    // Toggle LED.
    ledStatus = (ledStatus==HIGH ? LOW : HIGH);

    // Set LED pin status.
    digitalWrite(LEDBLINK_PIN, ledStatus);

    // Reset "next time to toggle" time.
    ledBlinkTime = millis()+LEDBLINK_MS;
  }
} // End of ledBlink()
```

KAYNAKLAR

1. <http://www.elektrikport.com/teknik-kutuphane/tatilde-baliklar-ac-kalmasin-arduino-bayram-projeleri/9020#ad-image-0>
2. <http://www.arduinouygulamalari.com/category/uygulamalar/>
3. <http://www.roboweb.net/rw-sf-11574.html>
4. <https://www.sparkfun.com/products/11574>
5. <http://playground.arduino.cc/Main/LEDHeartbeat>
6. <http://www.robotistan.com/>
7. <http://mekatronikmuhendisligi.com/dersler/arduino-dersleri>
8. <http://arduinoturkiye.com/arduino-baslangic-uzmanlik-tavsiyeler/>
9. <http://learn.parallax.com/reed-switch-arduino-demo>
10. <http://www.roboweb.net/rw-sf-8642.html>
11. http://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems
12. <https://learn.sparkfun.com/tutorials/arduino-comparison-guide>
13. <http://www.intorobotics.com/compare-arduino-gps-shields-and-modules-to-find-the-right-one-for-you/>
14. <http://store.arduino.cc/category/16>
15. <http://www.arduinom.org/arduino-projeleri/18b20-ve-16x2-lcd-ile-termometre>
16. <http://www.thewebblog.net/icerikgoster.php?yaziID=79>
17. <http://bloglug.com/arduino-projeleri-3-ldr-isik-sensoru-kullanimi/>
18. <http://www.arduinouygulamalari.com/uygulamalar/arduino-ile-ir-alici-verici-uygulamasi/>

EKLER

EK-1 – Mevcut Arduino modülleri

Modül	Besleme Gerilimi	Çalışma frekansı	Dijital I/O	Analog Giriş	PWM	UART	Programlama Arayuzu	Maliyet
ATmega328 Boards — 32kB Programlama Alanı // 1 UART // 6 PWM // 4-8 Analog Giriş // 9-14 Dijital I/O								
 Arduino Uno - R3	5V	16MHz	14	6	6	1	ATMega16U2 USB modülü ile	\$29.95
 Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	ATMega16U2 USB modülü ile	\$29.95
 RedBoard	5V	16MHz	14	6	6	1	FTDI USB modülü ile	\$24.95
 Arduino Pro 3.3V/8MHz	3.3V	8MHz	14	6	6	1	FTDI-Uyumlu soket ile	\$14.95
 Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Uyumlu soket ile	\$14.95
 Arduino Mini 05	5V	16MHz	14	8	6	1	FTDI-Uyumlu soket ile	\$33.95

 Arduino Pro Mini 3.3V/8MHz	3.3V	8MHz	14	8	6	1	FTDI-Uyumlu soket ile	\$9.95
 Arduino Pro Mini 5V/16MHz	5V	16MHz	14	8	6	1	FTDI-Uyumlu soket ile	\$9.95
 Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI-Uyumlu soket ile	\$59.95
 Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI-Uyumlu soket ileyada XBee Wireless modülü ile	\$24.95
 LilyPad Arduino 328 Main Board	3.3V	8MHz	14	6	6	1	FTDI-Uyumlu soket ile	\$21.95
 LilyPad Arduino Simple Board	3.3V	8MHz	9	4	5	0 ⁺	FTDI-Uyumlu soket ile	\$19.95
ATmega32U4 Boards — 32kB Programlama Alanı // 1 UART // 5-7 PWM // 12 Analog Giriş // 9-20								

Dijital I/O								
 Arduino Leonardo	5V	16MHz	20*	12	7	1	Doğrudan USB ile	\$24.95
 Pro Micro 5V/16MHz	5V	16MHz	12	12	5	1	Doğrudan USB ile	\$19.95
 Pro Micro 3.3V/8MHz	3.3V	8MHz	12	12	5	1	Doğrudan USB ile	\$19.95
 LilyPad Arduino USB	5V	16MHz	9	12	5	0	Doğrudan USB ile	\$24.95
ATmega2560 Arduino Mega's — 256kB Programlama Alanı // 4 UARTs // 14 PWM // 16 Analog Giriş // 54 Dijital I/O								
 Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	ATMega16U2 USB modülü ile	\$58.95
 Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Uyumlu soket ile	\$44.95

 Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Uyumlu soket ile	\$44.95
 Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Uyumlu soket ile	\$49.95
AT91SAM3X8E Arduino Due — 512kB Programlama Alanı // 4 UARTs // 12 PWM (2 DAC) // 12 Analog Giriş // 54 Dijital I/O								
 Arduino Due	3.3V	84MHz	54	12	12	4	Doğrudan USB ile	\$49.95

EK-2. Arduino Mimarisi

