

T.C  
NAMIK KEMAL ÜNİVERSİTESİ  
ÇORLU MÜHENDİSLİK FAKÜLTESİ



BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
DÖNEM PROJESİ

ARDUİNO FOTO KAPAN

AZAT ATEŞ  
DOĞAN KARAGÖZ  
ENES KAPUCU

DANIŞMAN  
Dr.Öğr.Üyesi ERKAN ÖZHAN

TEKİRDAĞ 2019/2020

GİRİŞ.....	1
ÖZET.....	1
PROJE HEDEFİ.....	1
SİMGELER VE KISALTMALAR.....	2
FİZİBİLİTELER.....	3
a-1 FİZİBİLİTE NEDİR.....	3
a-2 TEKNİK FİZİBİLİTE.....	3
a-3 YASAL (HUKUKİ) FİZİBİLİTE.....	4
a-4 FİNANSAL FİZİBİLİTE.....	4
a-5 GANT ŞEMASI (ZAMAN ÇİZELGESİ) .....	5
KULLANILAN TEMEL MATERYALLER.....	6
b-1 ARDUİNO UNO.....	6
b-2 ARDUİNO UNO USB KABLOSU.....	9
b-3 BREADBOARD.....	9
b-4 PIR SENSOR.....	12
b-5 OV7670 ARDUİNO KAMERA.....	12
b-6 JUMPER KABLO.....	13
b-7 DİRENÇLER.....	13
b-8 LED .....	14
PROJE DEVRE ŞEMASI.....	15
ARDUİNO KODU.....	16
JAVA KODU.....	17
PROJENİN ÇALIŞTIRILMASI.....	27
DİYAGRAMLAR.....	28
c-1 USE CASE DİYAGRAM.....	28
c-2 KULLANIM SENARYOSU.....	29
c-3 SINIF DİYAGRAMI.....	29
KAYNAKLAR / RESOURCES.....	30.

## **GİRİŞ**

- Projemizde amacımız hareket sensör görüş alanında hareket eden nesneleri yakalayıp fotoğrafı çekip sisteme kaydetmek. (foto-treps)(foto-kapan)
- Sistem bilgisayardan aldığı güç ve yazılım içerikleri ile çalışmakta ve çıktıyı tekrar bilgisayara kaydetmekte.
- Sistemimiz başta güvenlik olmak üzere bir çok farklı alanda kullanılabilmektedir.

## **ÖZET**

- Bu projede arduino modülü kullanılarak hareket alanına giren nesnelerin görüntülerinin yakalanıp bilgisayarda kaydedilmesi hedeflenmektedir.
- Bu projede Arduino UNO temel modüldür.
- Proje temelde devre, java kodu ve arduino kodu olmak üzere 3 parçaya incelenebilir.

## **PROJE HEDEFİ**

- Projemiz alan güvenliği konusunda bize yardımcı olmak üzere tasarlanmış bir modül sunar.

## SİMGELER VE KISALTMALAR

Boyut	Birim	Simge
Gerilim	Volt	V
Uzunluk	Metre	m
Zaman	Saniye	s
Direnç	Ohm	$\Omega$
Akım	Amper	A

SE = Standard Edition

JAR = Java ARchieve

JDK = Java Development Kit

USB = Universal Serial Bus

COM = Communication Port

REG = Register

MTX = Matrix

CLK = Clock

VGA = Video Graphics Array

AUX = Auxiliary

RGB = Red Green Blue

AVR = Alf & Vegard's RISC Processing

GND = Ground(Supply)

VCC = Voltage Common Collector(Supply)

RESET = Reset (Active Low)(Input)

PWDN = Power Down (Active high)(Input)

HREF = Horizontal Synchronization(Output)

VSYNC = Vertical Synchronization(Output)

SIOC = SCCB Clock(Input)

SIOD = SCCB Data(Input/Output)

PCLK = Pixel Clock(Output)

XCLK = System Clock(Input)

D0-D7 = Video Parallel Output(Output)

## FİZİBİLİTELER

### FİZİBİLİTE NEDİR

- Fizibilite sözcüğü TDK'nin sözlüğüne göre 'yapılabilirlik' anlamını taşır. Fizibilite en basit tanımıyla, bir projenin seçilmiş kısıtlara göre uygulanabilir olup olmadığının saptanmasıdır.
- Bir yatırım projesinin, yapılabilirliği ancak bir fizibilite raporuyla belli olur. Çünkü fizibilite raporu projenin ekonomik, teknik ve hukuki açıdan etkinliğini araştırır, ortaya bu konularla ilgili bilimsel gerçekler koyar.
- Öncelikle projenin gerçekten hayata geçirilip geçirilemeyeceğini belirler, sonra da seçilen kriterleri sağlayıp sağlayamadığını değerlendirir. Eğer projenin uygulanması yönünde karar alınır, son olarak bu uygulama için alternatif yaklaşım ve çözümler sunar.

### TEKNİK FİZİBİLİTE


- Kullanım kolaylığı ve çoklu port desteği için Windows 10 İşletim Sistemi
- COM5 veya dahili bir COM portu(Sistem başka bir portu kullanıyorsa Java kodu ona göre düzenlenmeli)  
(COM portları sadece Windows 7/8/8.1/10 işletim sistemi bilgisayarlar içindir)  
(Linux bilgisayarlar da port ttyACM0 veya ttyUSB0 olduğundan desteklemez)  
(MAC bilgisayarlar da tty.usbmodem\* veya tty.usbserial\* olduğundan desteklenmez)
- Dijital ve Analog girişleri destekleyen Arduino UNO mikroişlemci
- Bağlantı tipine göre Dişi-Erkek, Erkek-Erkek ve Dişi-Dişi Kablolar ile kurulan ara bağlantılar
- Arduino ile tam uyumlu OV7670 Kamera
- Arduino ile tam uyumlu PIR Sensor(Hareket Sensörü)
- Devrenin zarar görmesini engelleyen dirençler(5 Adet)
- Devrenin çalışıp çalışmadığını kontrol eden LED
- Eclipse Oxygen IDE, Java SE 8(jdk 1.8\_172)
- Arduino IDE(v1.8.10)
- Bağlantıları sağlamlaştırmak için elektrik bandı


## **YASAL FİZİBİLİTE**

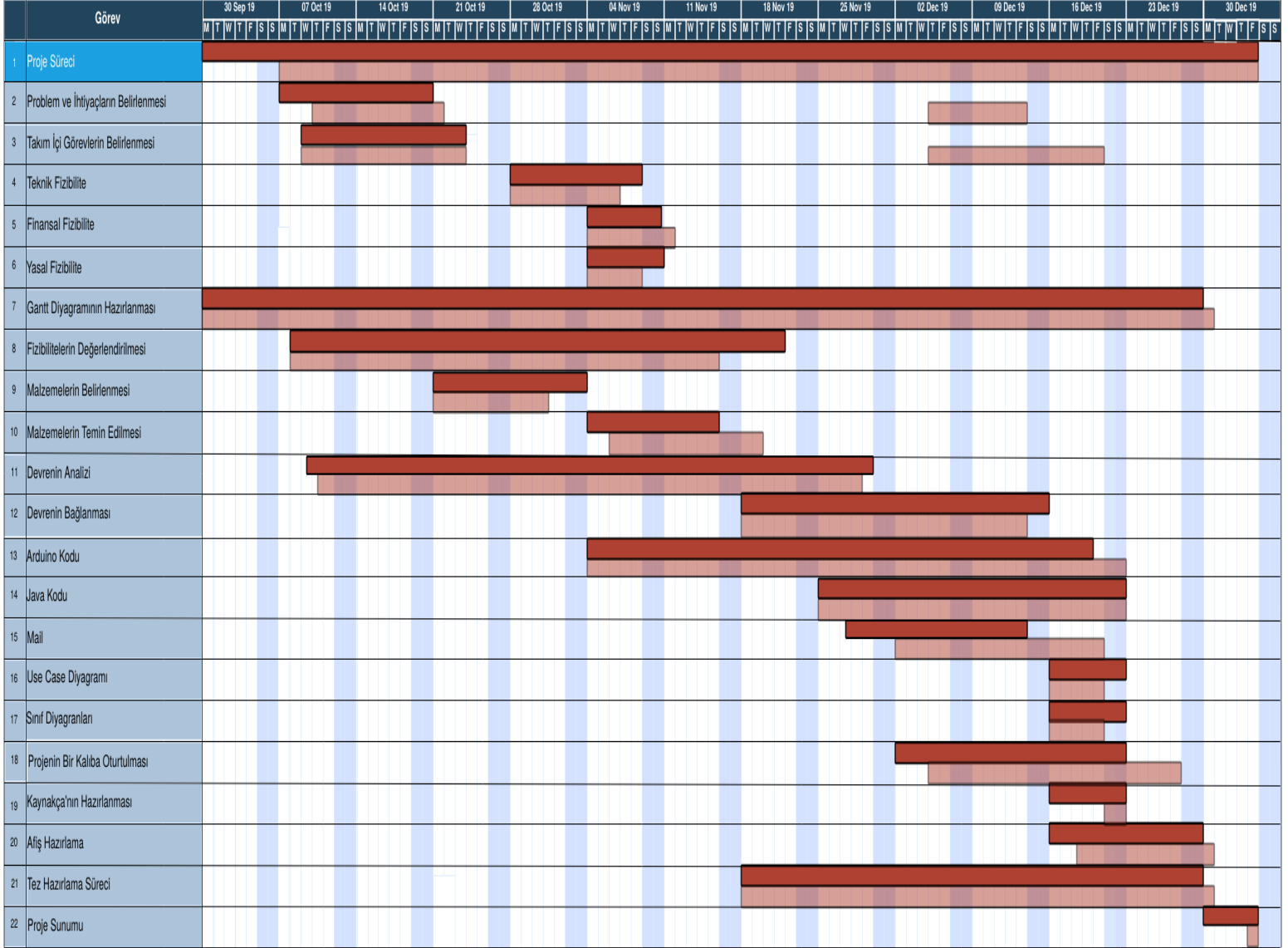
- Açık Kaynak Kodlu Eclipse IDE(Eclipse Oxygen)
- Açık Kaynak Kodlu JDK 8
- Açık Kaynak Kodlu Arduino IDE
- Açık kaynak kodlu Apache Commons Mail API (JAR)
- Açık Kaynak Kodlu Java Mail API (JAR)

## **FİNANSAL FİZİBİLİTE**

- |                        |   |
|------------------------|---|
| ➤ OV7670 Kamera Modülü | 42,25 ₺                                       |
| ➤ Jumper Kablolar      | 17,01 ₺ (3 farklı tip kullanıldı)             |
| ➤ LED(Yeşil, Kırmızı)  | 1,00 ₺ (Yeşil Test Edildi Kırmızı Kullanıldı) |
| ➤ Arduino Uno          | 0,00 ₺ (Halil Nusret Buluş'tan ödünç alındı)  |
| ➤ Dirençler            | 2,00 ₺ (5 farklı direnç kullanıldı)           |
| ➤ Breadboard           | 8,34 ₺ (Orta düzey yeterli görüldü)           |
| ➤ Pir Sensör           | 6,95 ₺  |
| ➤ Elektrik Bandı       | 3,00 ₺ (Kabloları sabitlemek için)            |

 = Planlanan

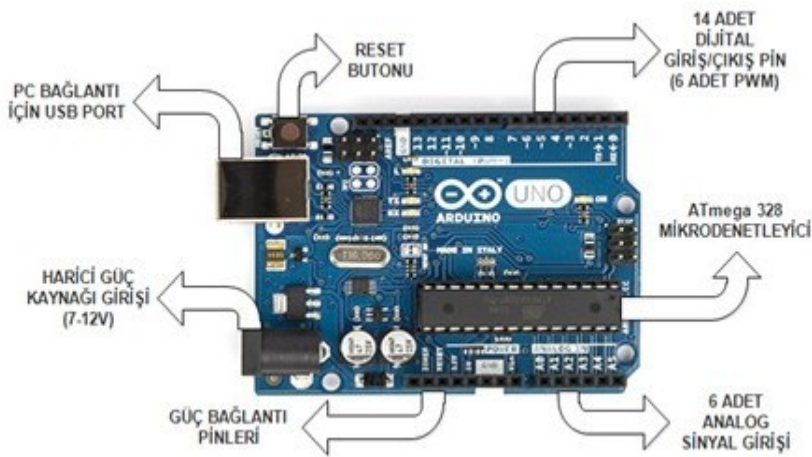
 = Gerçekleştirilen



## Temel Malzemeler

- Arduino Uno
- Arduino USB Kablosu
- Breadboard
- PIR Sensör
- OV7670
- Jumper Kablolar
- Dirençler
- LED

## Arduino UNO



- **Arduino** kolay kullanılabilir ve esnek bir yazılım/donanım mimarisine sahip, açık kaynak (open source) ailesine mensup bir elektronik geliştirme kartıdır.
- **Arduino Uno** ATmega328 mikrodeneleyici içeren bir **Arduino** kartıdır. **Arduino**'nın en yaygın kullanılan kartı olduğu söylenebilir.
- **Arduino Uno**, bilgisayarınızın USB portunu aşırı akım ve kısa devreden koruyan resetlenebilir bir çoklu sigortası bulunur. Çoğu bilgisayarın portlar için kendi korumaları olmasına rağmen bu sigorta ekstra bir koruma katmanı sağlar. Eğer USB portuna 500 mA den fazla bir yük binerse, sigorta otomatik olarak bağlantıyı kısa devre veya aşırı akım durumu ortadan kalkana dek keser.



- **Arduino Uno** barındırdığı ATmega 328 sayesinde 32 Kb lık bir hafızaya sahiptir ancak bunun 0.5 Kb lık kısmı Arduino bootloaderi tarafından kullanılmaktadır. Ayrıca Arduino Uno 2 Kb RAM ve 1 Kb EEPROM barındırır.
- **Arduino Uno** kendini korumak için 500 mA den fazla akım çektiğinde kendini kapatıp açar eğer bu sırada bilgisayara bağlıysa bağlantıyı keser.
- **Arduino Uno** ile programlamaya başlamadan önce bilgisayarınızda **Arduino'nun** kurulmuş olması gerekiyor. Kurulum işlemlerini tamamladıktan sonra **Arduino Uno'yu** bilgisayara tanıtmak için aşağıdaki işlemleri sırasıyla uygulayın :
  - **Araçlar>Kart>Arduino Uno** seçerek kartımızı belirtiyoruz.
  - **Araçlar>Port** seçtikten sonra Arduino Uno'nun bağlı olduğu portu seçiyoruz.
  - Böylelikle **Arduino Uno**, bilgisayarımıza tanımlanmış oluyor. **Yükle** butonuna basarak programı karta yüklüyoruz ve projemiz çalışmaya başlıyor.
- **Arduino Uno'nun** genel anlamda özellikleri aşağıdaki gibidir :
  - **Mikrodenetleyici** : ATmega328
  - **Çalışma Gerilimi** : 5V
  - **Giriş Gerilimi (önerilen)** : 7-12V
  - **Giriş Gerilimi (limit)** : 6-20V
  - **Dijital G/Ç Pinleri** : 14 (6 tanesi PWM çıkışı)
  - **Analog Giriş Pinleri** : 6
  - **Her G/Ç için Akım** : 40 mA
  - **3.3V Çıkış için Akım** : 50 mA
  - **Flash Hafıza** : 32 KB (ATmega328)
  - **SRAM** : 2 KB (ATmega328)
  - **EEPROM** : 1 KB (ATmega328)
  - **Saat Hızı** : 16 MHz
  - **Uzunluk** : 68.6 mm
  - **Genişlik** : 53.4 mm
  - **Ağırlık** : 25 g
- ✓ **Güç** :
- **Arduino Uno**, gücünü USB üzerinden veya adaptör girişinden alabilir. Yani bilgisayarınızın USB girişinden veya bilgisayarınızdan bağımsız olarak bir adaptör veya bataryadan güç elde edebilirsiniz. Doğrudan **Vin (+)** ve **GND (-)** pinlerinden de besleyebilirsiniz.

Harici güç kaynağı olarak 6-20V arası limit değerleri bulunmaktadır. Önerilen harici besleme gerilimi ise 7-12 V arasındır. Bunun sebebi 7V altındaki gerilimin stabil çalışmayıp, 12V üzeri gerilimin de aşırı ısınma sebebi olabilmesidir. Kart üzerinde bulunan regülatör sayesinde 7-12V arası gerilim 5V'a düşürülür ve kart bu şekilde çalışır :

- **Vin** : Harici güç kaynağı için kullanılan pin.
- **5V** : Regülatörden çıkan 5V çıkış gerilimini sağlar.
- **3V3** : Kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış verebilir.
- **GND** : Toprak (-) pinleridir.

✓ **Giriş/Çıkış (I/O) :**

14 adet dijital, 6 adet analog giriş/çıkış pini bulunmaktadır. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maksimum 40mA giriş ve çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir:

✓ **Seri Haberleşme- 0 (RX) ve 1 (TX) :**

TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Bilgisayardan karta program yüklenirken veya bilgisayar-UNO arasında karşılıklı haberleşme yapılırken de bu pinlerden faydalanılır. Bu sebeple, karta program yüklendiği esnada veya kartla bilgisayar arası iletişim kurulduğunda bu pinleri kullanmamak gerekir.

✓ **Harici Kesme (Interrupt)- 2 ve 3 :**

Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir.

✓ **PWM- 3,5,6,9,10 ve 11 :**

8-bit çözünürlükte PWM çıkış pinleridir.

✓ **SPI- 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) :**

SPI haberleşmesi için bu pinler kullanılır.

✓ **LED- 13 :**

Kart üzerinde dahili bir LED bulunmaktadır (L harfi ile gösterilmiş). Bu LED 13.pine bağlıdır. HIGH yapıldığında LED yanacak, LOW yapıldığında ise sönecektir.

✓ **Analog- A0,A1,A2,A3,A4,A5 :**

6 adet 10-bit çözünürlüğünde analog giriş pini bulunmaktadır. Bu pinler dijital giriş ve çıkış için de kullanılabilir. Pinlerin ölçüm aralığı 0-5V'tur.

✓ **I2C- A4 veya SDA pini ve A5 veya SCL pini :**

I2C haberleşmesi için bu pinler kullanılır.

✓ **HREF :**

Analog girişler için ölçüm referansı pini.

✓ **Reset :**

Resetleme işlemi için bu pin LOW yapılır. Bunun yerine kartta bulunan Reset butonuna da basılabilir.

**Not :** Arduino'da harici kesmede diğer pinler kullanılamaz duruma gelir

### Arduino USB Kablosu

- ✓ **Arduino Uno** birçok şekilde haberleşme işlemini gerçekleştirebilir.
- ✓ **RX** ve **TX** pinleri ile seri haberleşme imkanı mümkündür.
- ✓ **Atmega16u2** USB-seri dönüştürücü de bilgisayarda sanal bir seri port açarak Atmega328 ile bilgisayar arasında haberleşmeyi sağlar.
- ✓ **Arduino IDE** içerisinde yer alan seri monitör ile **Arduino** ile bilgisayar arasında metin temelli bilgilerin gönderilip alınmasını sağlar.
- ✓ **Arduino** ile bilgisayar arasında USB üzerinden bir haberleşme olduğunda **Arduino** üzerindeki RX ve TX yazan LED'ler yanar.



### Breadboard

**Breadboard** üzerinde devrelerimizi test ettiğimiz araçtır. Kurduğumuz devreleri birbirlerine lehimlemeden kolaylıkla test etmemizi sağlar. Tasarladığımız devreleri baskı devre veya delikli plaketler üzerine aktarmadan önce denememize olanak sağlar. Bu sayede devre bağlantılarını kontrol ederek bir hata olup olmadığını gözlemlemiş oluruz. Devreleri tak-çıkart şeklinde kurabildiğimiz için kullandığımız elektronik bileşenleri başka projelerde tekrar kullanma imkanı verir.

**Breadboard** iç yapısı dik ve yatay şekilde birbirlerine bağlı halde konumlanmış metal kısıpçılardan oluşur. Dışarıdan baktığımızda gördüğümüz kırmızı ve mavi kısımları **breadboardun** satır kısımlarıdır. Bu kısımlar boydan boya bir satır şeklinde iletim halindedir. Bu kısımları **breadboardun** iki yanında görebilirsiniz. Burada dikkat etmemiz gereken nokta bu satır bazı **breadboardlarda** ortadan ikiye ayrılmış durumdadır. Yani baştan sona kısa devre değildir.

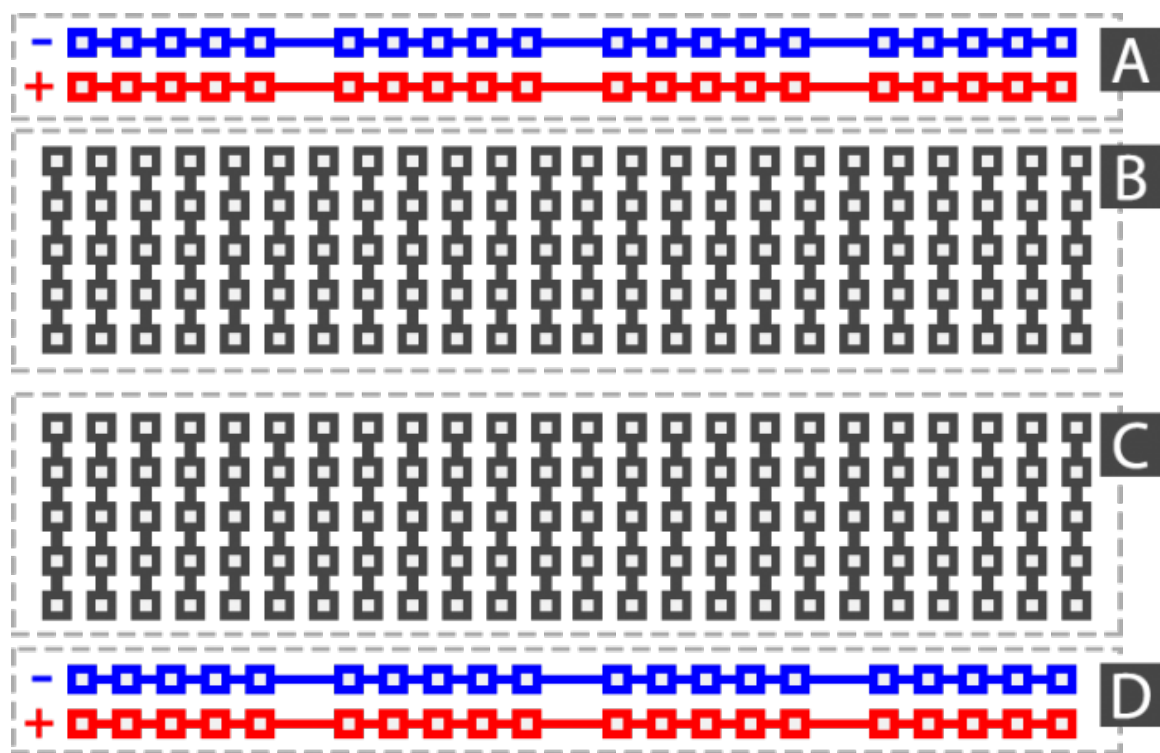
**Breadboardun** ortada kalan kısımları da sütun boyunca yerleştirilmiş iletkenlerden oluşur. Bu kısımlar da tıpkı satırlarda olduğu gibi **breadboardun** iki tarafında bulunur. Tüm bu iletkenlerin üstü elektronik bileşenlerin ayaklarını yerleştirmemiz için açılmış deliklerden oluşan bir plastik ile kapalıdır.

**Breadboardlar** kendi içlerinde ebatlarına göre birkaç çeşitten oluşurlar. Burada hangi **breadboardu** kullanacağımız projemizdeki elektronik bileşenlerin miktarı ve bacak sayıları ile alakalıdır. Projenizin büyüklüğüne göre istediğiniz **breadboardu** tercih edebilirsiniz. Karmaşık yapıll ve çok bileşenli elektronik projelerde büyük **breadboard** bile yeterli olmayabilir. Bu durumlarda birden fazla breadboardu yanlarındaki çentikler yardımıyla birleştirerek istenilen boyutta bir board elde edilir. Genel anlamıyla breadboardları mini boy, orta boy ve büyük boy breadboard şeklinde ayırabiliriz.

- Mini **Breadboard**
- Orta **Breadboard**
- Büyük **Breadboard**

**Breadboard** üzerindeki + ve – şeklinde belirtilmiş satırlar gerilim bağlantılarını sağlamak için kullanılır. Gerilim bağlantılarını sağladıktan sonra board üzerinde yerleştirilen bileşenlerin güç ihtiyaçları bu hatlar üzerinden karşılanır.

Orta kısımda 5 delikten oluşan sütunlar bulunmaktadır. Bu delikler aynı metal parçasının üzerinde konumlandığı için birbirleriyle iletim halindedir. Yani aynı sütun üstündeki deliklere bağlayacağımız bacaklar iletim durumuna geçer. Bu işlem sayesinde komponentleri birbirine bağlamış oluruz. Bunu iç yapısını incelediğimizde çok daha iyi anlayabiliyoruz. Board üzerinde bulunan kırmızı-mavi güç hattı ve sütunlar her iki tarafta da bulunmaktadır. Bu hatlar birbirinden ayrıktır, yani iletim halinde değildirler.



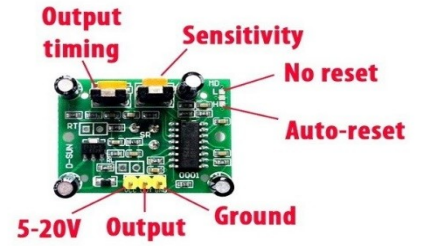
## PIR Sensor (Hareket Sensörü)

**PIR Sensörleri**, bir ortamda oluşan canlı hareketini algılamak için kullanılan sensörlerdir. Bu minik boyutlu sensör, çeşitli elektronik, robotik ve hobi uygulamalarında rahatça kullanılabileceğiniz, **Arduino** başta olmak üzere bir çok mikrodenetleyici platformu ile beraber kullanılabilir modüldür.

Dijital çıkışlı olan bu modül, ortamda hareket algılamadığı zaman lojik 0, hareket algıladığı zaman ise lojik 1 çıkışı vermektedir. Sensör üzerinde Sx ve Tx olmak üzere iki adet potansiyometre bulunmaktadır. Sx potansiyometresi sensörün görme mesafesini 3 ile 5 metre arasında değiştirmektedir. Tx potu ise sensör gördükten sonra ne kadar süre daha çıkış pininden lojik 1(3.3V) çıkışını vereceğini ayarlamaktadır.

### Özellikleri:

- Çalışma Voltajı: 5-12V
- Lojik Sinyal Çıkış Seviyesi: 3,3V
- Algılama Alanı: 3-5 metre
- Algılama Açısı: 140°
- Bekleme Süresi: 5-200 saniye
- Ürün Boyutları: 33x25x24 mm



## OV7670 Arduino Kamera Modülü

Bu kamera modülü robotlar, elektronik arabalar, görüntü işleme ve diğer bir çok elektronik cihaz için üretilmiş bir üründür. Elektronik-yazılım projeleri için ideal bir çözümdür.

### ÖZELLİKLER:

- **OV7670/OV7171** full-frame, sub-sampled ya da windowed 8-bit imajları, Serial Camera Control Bus (SCCB) arayüzü ile çeşitli formatlarda kullanıcıya sunar .
- Ürün, tamamen kullanıcı kontrolündeki kalite, format ve çıkış veri transferi ile 30fps ye kadar görüntü alabilir.
- SCCB arayüzü sayesinde gama eğrisi sürecinde, beyaz dengesi, doygunluk, renk tonu gibi tüm görüntü işleme fonksiyonları ayarlanabilir.
- Düşük çalışma gerilimi,
- Düşük ışık üretimi için yüksek hassasiyet,
- Gömülü taşınabilir uygulamalar için düşük çalışma gerilimi,
- ISS gürültü azaltma ve hata düzeltme,
- Otomatik pozlama ve otomatik beyaz ayarı,
- Otomatik bant filtresi ve otomatik siyah seviye kalibrasyonu



## TEKNİK ÖZELLİKLER:

- Ürünün VGA görüntü kalitesi: 30 kare/saniye.
- Optik boyutu: 1/6",
- Çözünürlük: 0.3 megapiksel. 640x480 VGA.
- Hassasiyet: 1.3V,
- Çalışma Sıcaklığı: -30°C~ 70°C,
- Görüntü desteği: VGA, QVGA, QQVGA, CIF, QCIF, QQCIF
- I2C arayüzü ile uyumlu Standart SCCB arayüzü
- Yüksek kaliteli F1.8/8mm lens
- Regülatör : onboard
- 52 db'lik dinamik aralık,
- Sinyal / gürültü oranı: 46db,
- Görüş açısı: 25°,
- Çıkış formatları : Raw RGB, RGB (GRB 4:2:2, RGB565/555/444)



## Jumper Wires (Jumper Kablolar)

Kısaca bir çeşit bağlantı kabloları diyebiliriz.

**Breadbord** ve **Arduino** arasında bağlantı kurmak

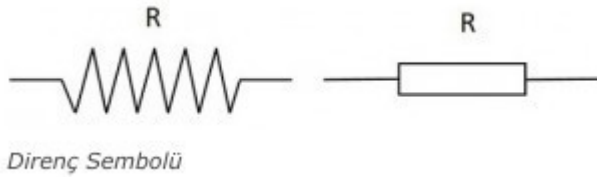
için oldukça kullanışlıdır. Uçlarında erkek ve dişi girişlerin bulunmasına göre 3 çeşit **jumper kablo** bulunmaktadır :

- 1- Erkek – Erkek
- 2- Erkek – Dişi
- 3- Dişi – Dişi



## Dirençler

Elektrik devrelerinde direnç, bir iletken üzerinden geçen elektrik akımının karşılaştığı zorlanmadır. Mekanik sistemlerdeki sürtünmeye benzer özellikler gösterir. Direncin birimi **Ohm ( $\Omega$ )**'dur. Denklemlerde **R** harfi ile gösterilir. Elektronik devrelerde direncin sembolü 2 farklı şekilde gösterilebilir :



**Dirençler**, elektrikli devrelerde akımı sınırlayarak belli bir değerde tutmaya yararlar. Bunun haricinde hassas devre elemanlarının üzerlerinden yüksek akım geçmesini önlerler, besleme gerilimini ve akımı bölmek için de kullanılırlar. Ayrıca **dirençlerin** üzerlerine düşen akım değeri yükseldikçe ısınmalarından da faydalanılmaktadır.

Renk	1. Şerit (ilk basamak)	2. Şerit (ikinci basamak)	3. Şerit (üçüncü basamak)	4. Şerit (katsayı)	5. Şerit (tolerans)	6. Şerit (sıcaklık katsayısı)
Siyah	0	0	0	$\times 10^0$		
Kahve	1	1	1	$\times 10^1$	$\pm \%1$	100
Kırmızı	2	2	2	$\times 10^2$	$\pm \%2$	50
Turuncu	3	3	3	$\times 10^3$	$\pm \%3$	15
Sarı	4	4	4	$\times 10^4$	$\pm \%4$	25
Yeşil	5	5	5	$\times 10^5$	$\pm \%0.5$	
Mavi	6	6	6	$\times 10^6$	$\pm \%0.25$	10
Mor	7	7	7	$\times 10^7$	$\pm \%0.1$	5
Gri	8	8	8	$\times 10^8$	$\pm \%0.05$	
Beyaz	9	9	9	$\times 10^9$	$\pm \%1$	
Altın					$\pm \%5$	
Gümüş					$\pm \%10$	

4 adet renk şeridi,  
soldan sağa şu  
değerleri ifade eder:

1. Şerit : İlk basamak
2. Şerit : İkinci basamak
3. Şerit : Çarpma katsayısı
4. Şerit : Tolerans

### LED( Light Emitting Diode)(Işık Yayan Diyot)

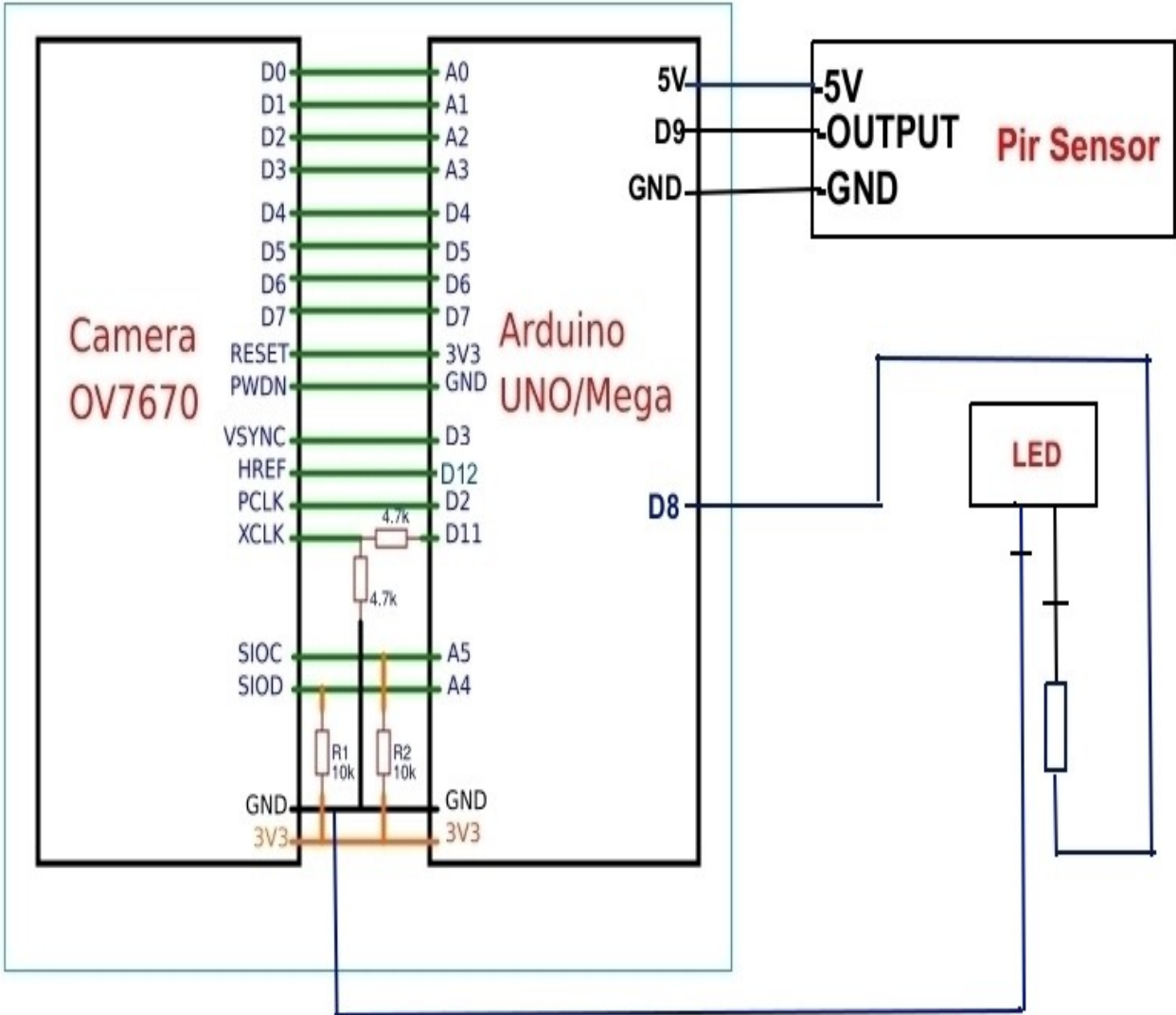
**LED**, bir diyottur. Bildiğimiz üzere diyot, akımın yalnızca bir yönden geçmesini sağlayan iki bacaklı yarı-iletken bir devre elemanıdır.

**LED'in** de diğer diyotlar gibi yapısında p-tipi ve n-tipi olmak üzere iki farklı çeşit yarı-iletken madde bulunur. P-tipi yarı-iletken pozitif yük taşıyıcılar, n-tipi yarı-iletken ise negatif yük taşıyıcılar bulunur. Bu sayede, diyot üzerinden yalnızca anottan katot yönünde elektrik akımı geçişi mümkündür.





## DEVRENİN BAĞLANMASI



## ARDUİNO CODE

```
/* OV7670.h dosyası uzunluğundan dolayı teze eklenmemiştir dosyaya
https://github.com/azatates7/FirstTermProject/blob/master/camera/OV7670.h
linkinden erişebilirsiniz. */

#include </home/azatates7/Downloads/Project/camera/OV7670.h>

void setup(){
  int pirPin = 9; // PIR pin
  int deger = 0; // PIR value
  pinMode(9,OUTPUT);  pinMode(pirPin, INPUT);
  Serial.begin(9600); //Serial Porttan veri göndermek için baudrate ayarı
  arduinoUnoInut(); camInit();
  setResolution(); setColor();
  writeReg(0x11, 10);
}

void loop(){
  deger = digitalRead(pirPin); // Dijital pin okunuyor
  Serial.println(deger);      // Okunan değer seri porttan okunuyor.
  if (deger == HIGH) {
    captureImg(320, 240); digitalWrite(9, HIGH);
    delay(2000); digitalWrite(9, LOW); }
  else{
    delay(2000);
    digitalWrite(9, LOW);
  }
}
```

## **JAVA KODU**

Java kodunu eksiksiz çalıştırmak için “comm.jar”, “commons-email-1.5.jar”, “mail.jar” ve “win32com.dll” dosyalarını import etmemiz gerekir.

### **SimpleRead.java (Main Class)**

```
//package main;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.Enumeration;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.comm.CommPortIdentifier;
import javax.comm.PortInUseException;
import javax.comm.SerialPort;
import javax.comm.UnsupportedCommOperationException;
import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.HtmlEmail;
```

```

public class SimpleRead {
    private static final char[]COMMAND = {'*', 'R', 'D', 'Y', '*'};
    private static final int WIDTH = 320; // 640;
    private static final int HEIGHT = 240; // 480;

    private static CommPortIdentifier portId;
    InputStream inputStream;
    SerialPort serialPort;

    public static void main(String[] args) {
        Enumeration portList = CommPortIdentifier.getPortIdentifiers();
        while (portList.hasMoreElements()) {
            portId = (CommPortIdentifier) portList.nextElement();
            if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL) {
                System.out.println("Port name : " + portId.getName());
                if (portId.getName().equals("COM5")) { // port kontrolü
                    SimpleRead reader = new SimpleRead();
                }
            }
        }
    }

    public SimpleRead() throws UnsupportedOperationException {
        int[][] rgb = new int[HEIGHT][WIDTH];
        int[][] rgb2 = new int[WIDTH][HEIGHT];
    }
}

```

```

try { // HATA KONTROL BLOĞU

    serialPort = (SerialPort) portId.open("SimpleReadApp", 1000);
    inputStream = serialPort.getInputStream();
    serialPort.setSerialPortParams(1000000,
        SerialPort.DATABITS_8,
        SerialPort.STOPBITS_1,
        SerialPort.PARITY_NONE);

    int counter = 0;
    while (true) {
        System.out.println("Looking for image");
        while (!isImageStart(inputStream, 0)) {
        }
        System.out.println("Found image : " + counter); // Çekilen Resim Sayısı

        for (int y = 0; y < HEIGHT; y++) {
            for (int x = 0; x < WIDTH; x++) {
                int temp = read(inputStream);
                rgb[y][x] = ((temp & 0xFF) << 16) | ((temp & 0xFF) << 8) |
                    (temp & 0xFF);
            }
        }

        for (int y = 0; y < HEIGHT; y++) {
            for (int x = 0; x < WIDTH; x++) {
                rgb2[x][y] = rgb[y][x];
            }
        }
    }
}

```

```

// dosya adı için zaman bilgisi elde etme
DateFormat sdf = new SimpleDateFormat("yyyyMMddHHmmss");
Date date = new Date();

// bmp formatında resim kaydetme
BMP bmp = new BMP();
bmp.saveBMP("C:/Users/DK/Downloads/dk/" + sdf.format(date) +
".bmp", rgb2);
System.out.println("Saved image : " + counter);
// mail için dosya kontrolü
String filename = "C:/Users/DK/Downloads/dk/" +
sdf.format(date) + ".bmp";
File file = new File(filename);
if (!file.exists()) {
    System.out.println("File Not Created");
}
// toplu mail için kullanıcıları tanımlama ve mail ayarlarını yapmak
ArrayList<String> mailaddress = new ArrayList<String>();
mailaddress.add("enes.kapucu@outlook.com.tr");
mailaddress.add("azatates4977@gmail.com");
mailaddress.add("hikmetarasdk@gmail.com");
for (String mail : mailaddress) {

```

```

HtmlEmail email = new HtmlEmail(); // email nesnesi yaratılıyor

    email.setHostName("smtp.gmail.com"); // host ayarı tanımlama
    email.setSmtpport(465); // port bilgisi tanımlama
    email.setAuthenticator(new DefaultAuthenticator
        ("testmailjavaapache@gmail.com", "Apachemail1234")); //
güvenlik ayarları yapılmış yönlendirme servisi

    email.setSSLonConnect(true); // ssl desteğini aktif etme
    email.setFrom("testmailjavaapache@gmail.com");

    email.setSubject("Test Mail"); // konu
    email.setMsg("Test Mail Sended"); // mesaj
    email.attach(file); // gönderilecek dosya

    email.addTo(mail); // mail adresi bilgisi ekleme
    email.send(); // gönder
    System.out.println("File Sended Succesfully");
}
}
}
catch (IOException | EmailException | PortInUseException ex) {
    System.out.println("Hata Algılandı : " + ex.getMessage()); // hata
yakalama
}
}

```

```

        private void copyFileUsingApacheCommonsIO(File path, File path1)
throws IOException {
            Files.copy(path.toPath(), path1.toPath());
        }

```

```

private int read(InputStream inputStream) throws IOException {
    int temp = (char) inputStream.read();
    if (temp == -1) {
        throw new IllegalStateException("Exit");
    }
    return temp;
}

```

```

private boolean isImageStart(InputStream inputStream, int index) throws
IOException {
    if (index < COMMAND.length) {
        if (COMMAND[index] == read(inputStream)) {
            return isImageStart(inputStream, ++index);
        } else {
            return false;
        }
    }
    return true;
}
}

```



### **BMP.java (Main Class'ından çağrılan BMP yapısı oluşturma sınıfı)**

```
//package main;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

public class BMP {
    byte [] bytes;

    public int[][] readBMP(String fileName) {
        byte[]buf = new byte[54];    int[][]rgb = null;

        try {
            FileInputStream fos = new FileInputStream(new File(fileName));
            fos.read(buf, 0, buf.length);

            int width = ((buf[21]&0xFF) << 24) + ((buf[20]&0xFF) << 16) +
            ((buf[19]&0xFF) << 8) + (buf[18]&0xFF);

            int height = ((buf[25]&0xFF) << 24) + ((buf[24]&0xFF) << 16) +
            ((buf[23]&0xFF) << 8) + (buf[22]&0xFF);
            rgb = new int[height][width];
```

```

    for (int y = 0; y < height; y++) {
        for (int x = 0; x < width; x++) {
            fos.read(buf, 0, 3);
            rgb[y][x] = ((buf[2]&0xFF) << 16) + ((buf[1]&0xFF) << 8) +
(buf[0]&0xFF);
        }
    }
    fos.close();
}
catch (IOException e) {
    throw new IllegalStateException(e);
}
return rgb;
}

public void saveBMP(String filename, int [][] rgbValues){
    try {
        FileOutputStream fos = new FileOutputStream(new File(filename));
        bytes = new byte[54 + 3*rgbValues.length*rgbValues[0].length];
        saveFileHeader();
        saveInfoHeader(rgbValues.length, rgbValues[0].length);
        saveBitmapData(rgbValues);
        fos.write(bytes);
        fos.close();
    } catch (IOException e) {
        throw new IllegalStateException(e);
    }
}

```

```
private void saveFileHeader() {  
    bytes[0]='B';  
    bytes[1]='M';  
    bytes[5]=(byte) bytes.length;  
    bytes[4]=(byte) (bytes.length>>8);  
    bytes[3]=(byte) (bytes.length>>16);  
    bytes[2]=(byte) (bytes.length>>24);  
    bytes[10]=54;  
}
```

```
private void saveInfoHeader(int height, int width) {  
    bytes[14]=40;  
    bytes[18]=(byte) width;  
    bytes[19]=(byte) (width>>8);  
    bytes[20]=(byte) (width>>16);  
    bytes[21]=(byte) (width>>24);  
    bytes[22]=(byte) height;  
    bytes[23]=(byte) (height>>8);  
    bytes[24]=(byte) (height>>16);  
    bytes[25]=(byte) (height>>24);  
    bytes[26]=1;  
    bytes[28]=24;  
}
```

```

private void saveBitmapData(int[][] rgbValues) {
    for(int i=0;i<rgbValues.length;i++){
        writeLine(i, rgbValues);
    }
}

private void writeLine(int row, int [][] rgbValues) {
    final int offset=54;
    final int rowLength=rgbValues[row].length;
    for(int i=0;i<rowLength;i++){
        int rgb=rgbValues[row][i];
        int temp=offset + 3*(i+rowLength*row);

        bytes[temp + 2] = (byte) (rgb>>16);
        bytes[temp + 1] = (byte) (rgb>>8);
        bytes[temp] = (byte) rgb;
    }
}

```

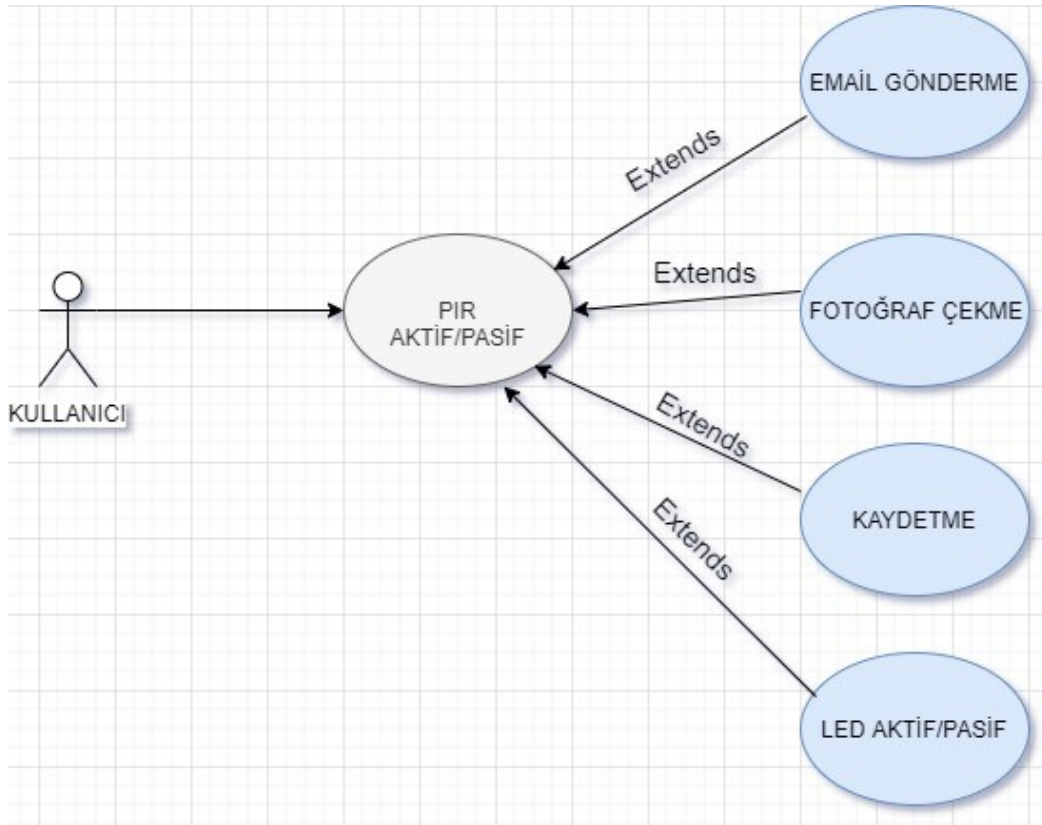
## **PROJENİN ÇALIŞTIRILMASI**

- Kabloyu bilgisayara tak.
- Arduino programını aç.
- Arduino Kodlarının olduğu dosyayı aç.
- Arduino Programında portu kontrol et aktif değilse aktifleştir.
- Arduino Kodunu derle hata varsa hata kontrol edip düzelt.
- Arduino kodunu yükle.
- Java IDE (Eclipse vb.) aç.
- Import kısmından rojeyi import edelim.
- IDE'de Proje içerisinde yer alan SimpleRead class'ını çalıştır.
- Hata alınırsa kabloyu çıkarıp tekrar takınız(Port Çakışması)
- Hata alınmazsa hareket sensörü devreye girdikçe Java Kodu Arduino yu tetikler ve kameradan kaydedilen görüntüler java kodu aracılığıyla .bmp formatında bilgisayara kaydedilir.
- Kaydedilen her resim belirlenen mail adreslerine toplu olarak yollanır.
- Java Kodu kapatılana kadar çalışma sürer.
- Java kodu durdurulup yeniden başlatıldığında hata verirse kabloyu çıkarıp yeniden takınız(Port Çakışması)

## DİYAGRAMLAR

**Diyagram**, herhangi bir olayın gelişimini, değişimini gösteren grafiktir.

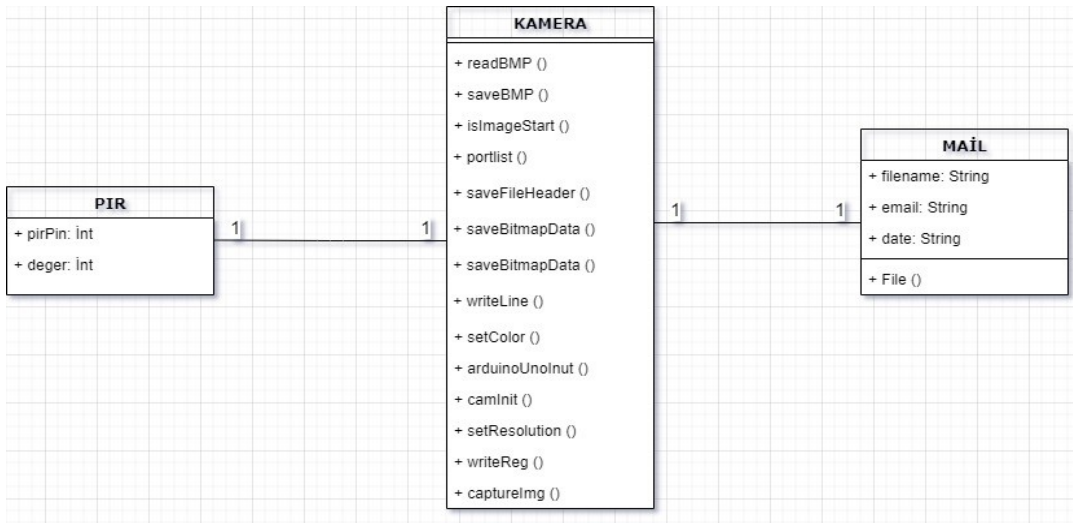
### USE CASE DİYAGRAMI



## KULLANIM SENARYOLARI

SENARYO KS 1	PIR AKTİF/PASİF
Birinci Aktör	Kullanıcı
İlgililer ve Beklentileri	Kullanıcı: Kabloyu takmış ve sistemi başlatmış olması gerekir.
Ön Koşullar	Sistemin hareket alması gerekli.
Son Koşullar	Yok
Ana Akış	1-Kabloyu bilgisayara tak. 2-Sistemi başlat. 3-Hareket almasını sağla.
Alternatif Akış	3a Algılanmazsa çalışmaz.2a Port hatası verebilir. Fotoğraf çekmediği için Mail de yollayamaz.

## SINIF DİYAGRAMI



## KAYNAKLAR / RESOURCES

- \* <https://blog.mikrox.com.tr/fizibilite-raporu-nedir-nasil-hazirlanir/>
- \* <https://www.arducam.com/product/arducam-ov7670-0-3-megapixel-camera-module-for-arduino-boards/>
- \* <http://embeddedprogrammer.blogspot.com/2012/07/hacking-ov7670-camera-module-sccb-cheat.html>
- \* <https://circuitdigest.com/microcontroller-projects/how-to-use-ov7670-camera-module-with-arduino>
- \* <https://github.com/ArduCAM/Arduino>
- \* <https://diyot.net/arduino-uno-r3/>
- \* <https://www.devreokulu.com/DirencHesaplama.html>
- \* <https://github.com/metinakkin/Ov7670>
- \* <https://www.open-electronics.org/a-complete-guide-to-arduino-based-video-camera/>
- \* <https://www.youtube.com/watch?v=iIlXHo9wG3Q> (Arduino ile Led Yakmak - Blink Uygulaması #2 -Robotistan) [https://www.youtube.com/watch?v=\\_2JJ29kmwRU](https://www.youtube.com/watch?v=_2JJ29kmwRU)&list=PLDRcccSktQd5mfXDtGv975V77RCrW6H7U&index=2 (Arduino Nedir? Nasıl Kurulur ve Neler Yapılabilir? #1 - Robotistan)
- \* <https://www.youtube.com/watch?v=jgPZsxYvSIk>&list=PLDRcccSktQd5mfXDtGv975V77RCrW6H7U&index=11 (Arduino ile Ultrasonik Sensör (HC-SR04) Kullanımı - Park Sensörü Yapımı -Robotistan)



- \* <http://www.alselectro.com/arduino-camera-ov7670.html>
- \* <https://hub360.com.ng/product/ov7670-camera-module-for-arduino/>
- \* <https://www.youtube.com/watch?v=NoeeYSo0jeg&list=PLDRcccSktQd5mfXDtGv975V77RCrW6H7U&index=17> (Arduino Hareket Sensörü Kullanımı (PIR Sensörü) - Robotistan)
- \* <https://create.arduino.cc/projecthub/techmirtz/visual-capturing-with-ov7670-on-arduino-069ebb>
- \* <https://www.teachmemicro.com/arduino-camera-ov7670-tutorial/>
- \* <https://www.electronics-lab.com/using-ov7670-camera-sensor-arduino/>
- \* <http://acoptex.com/project/254/basics-project-052a-ov7670-camera-sensor-module-without-fifo-ram-at-acoptexcom/#sthash.x99DafwZ.dpbs>
- \* <https://maker.robotistan.com/breadboard/>
- \* <https://dersimizkodlama.com/jumper-kablo-nedir/>
- \* <https://www.guvenlikonline.com/makale/219/pir-sensor.html>
- \* <http://devreokulu.com/LedAnotKatot.html>
- \* <https://robotizmo.net/vga-kamera-modulu-ov7670-41>
- \* <https://maker.robotistan.com/direnc/>
- \* <https://maker.robotistan.com/raspberry-pi-dersleri-10-pir-sensor-ile-hareket-algilama/>
- \* <https://www.robotistan.com/hc-sr501-ayarlanabilir-ir-hareket-algilama-sensoru-pir>
- \* <https://maker.robotistan.com/led/>

- \* <https://www.robotekno.com/arduino-kamera-modulu-ov7670-pic-kamera-modulu>
- \* <https://shop.rasp.io/products/assorted-breadboarding-jumper-wires-dupont-connector-female-male-10cm-prototyping-raspberry-pi>
- \* <https://www.instructables.com/id/OV7670-Arduino-Camera-Sensor-Module-Framecapture-T/>
- \* <https://maartentech.wordpress.com/arduino-projects/ov7670-camera-module/>
- \* <https://hkalasua.wordpress.com/2017/09/11/ov7670-arduino-sd/>
- \* <https://lucidar.me/en/arduino/camera-ov7670-and-arduino/>
- \* Pınar Tüfekçi Sistem Analizi ADYS Notları
- \* Alpay Doruk Yazılım Mühendisliği ADYS Notları
- \* <https://maker.robotistan.com/arduino-uno/>
- \* <https://hayaletveyap.com/arduino-uno-r3-nedir/>
- \* <https://www.slideshare.net/SomnathSharma4/arduino-camera-interfacing-ov7670>
- \* <http://kathyjan.blogspot.com/2014/12/how-to-use-ov7670-camera-module-with.html>
- \* <https://urun.n11.com/diger/ov7670-vga-kamera-modulu-P378338174>