

T.C  
TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ  
ÇORLU MÜHENDİSLİK FAKÜLTESİ



**BİLGİSAYAR MÜHENDİSLİĞİ PROJE-II  
BİTİRME ÇALIŞMASI**

**ARDUINO FOTO KAPAN 2.0**

**1160606905 KAAN TUNAHAN YAGAN**

**1160606801 DOĞAN KARAGÖZ**

**1160606001 AZAT ATEŞ**

**DANIŞMAN**  
**Dr. Öğr. Üyesi Erkan ÖZHAN**

**TEKİRDAĞ-2020**

# İÇİNDEKİLER

GİRİŞ .....	3
PROJE HEDEFİ .....	3
ÖZET .....	3
SİMGELER VE KISALTMALAR .....	4
FİZİBİLİTELER .....	5
FİZİBİLİTE NEDİR.....	5
TEKNİK FİZİBİLİTE .....	5
YASAL FİZİBİLİTE.....	6
FİNANSAL FİZİBİLİTE .....	6
GANTT DİYAGRAMI .....	7
Temel Malzemeler.....	8
Arduino MEGA .....	8
Arduino USB Kablosu .....	11
Breadboard .....	11
PIR Sensor (Hareket Sensörü).....	14
OV7670 Arduino Kamera Modülü.....	14
Jumper Wires (Jumper Kablolar) .....	15
Dirençler.....	15
SD Kart Modülü .....	16
9 V Pil Başlığı .....	18
9 V Pil.....	18
LM 7805 Voltaj Regülatörü .....	18
Servo Motor.....	19
OTG Kablosu .....	20
USB Uzatma Kablosu .....	20
DEVRENİN BAĞLANMASI.....	21

ARDUINO KODU 1 .....	22
ARDUINO KODU 2 .....	25
JAVA KODU .....	27
SimpleRead.java (Main Class) .....	27
BMP.java (Main Class'ından çağrılan BMP yapısı oluşturma sınıfı) .....	33
PROJENİN ÇALIŞTIRILMASI .....	37
DİYAGRAMLAR .....	38
USE CASE DİYAGRAMI .....	38
KULLANIM SENARYOLARI .....	39
SINIF DİYAGRAMI .....	39
KAYNAKÇA .....	40

## GİRİŞ

Bu projede Arduino modülü kullanılarak hareket algılanan bölgeye kamera çevrilerek kamera aracılığıyla görüntü alınır ve direk SD Card a kaydedilir. Proje pil desteğiyle doğa fotoğrafçılığı gibi alanlar için de kullanışlı hale getirilmiştir.

Sistemimiz başta güvenlik olmak üzere birçok farklı alanda kullanılabilmektedir.

Ayrıca ilk dönem projesi kodları üzerinde modifiye yapılarak ikinci dönem projesi üzerinde kullanılabilir hale getirilmiştir. (Bilgisayar bağlantısı ile bilgisayara kaydetme ve mail atma)

## PROJE HEDEFİ

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

Hareket sensörleri aracılığıyla gelen hareket yönüne bağlı olarak kamera döner ve aldığı görüntüyü SD Card a doğrudan aktarır. Böylece bir güç ve depolama cihazına bağlı olmadan hem güç problemi yaşamaz hem de görüntü direk kaydedilir.

## ÖZET

Bu projede Arduino modülü kullanılarak hareket algılanan bölgeye kamera çevrilerek kamera aracılığıyla görüntü alınır ve direk SD Card a kaydedilir. Proje pil desteğiyle doğa fotoğrafçılığı gibi alanlar için de kullanışlı hale getirilmiştir.

Bu projede Arduino Mega temel modüldür.

Proje temelde devre ve Arduino kodu olmak üzere 2 parçada incelenebilir.

Ayrıca ilk dönemki proje için Java ve Arduino kodu ile proje bilgisayar bağlantısı üzerinden çalışabilir hale getirilmiştir.

Proje, devrenin bir dış kalıba konulması ile devre daha sade ve hoş görünen bir hale getirilmiştir.

**Anahtar Kelimeler:** Arduino, Mikrodenetleyici, Foto Kapan, Yazılım, Servo, PIR, Kamera, Java

## SİMGELER VE KISALTMALAR

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

SE = Standard Edition  
JAR = Java ARchieve  
JDK = Java Development Kit  
SD Card = Secure Digital Card  
mSD = micro SD  
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 kıstaslara göre uygulanabilir olup olmadığıının saptanmasıdır.

Fizibilite raporu projenin ekonomik, teknik ve hukuki açıdan etkinliğini araştırır, ortaya bu konularla ilgili bilimsel gerçekler koyar.

Öncelikle projenin gerçekten hayatı geçirip geçirilemeyeceğini belirler, sonra da seçilen kriterleri sağlayıp sağlayamadığını değerlendirir. Eğer projenin uygulanması yönünde karar alınırsa, 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 dâhili 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 Mega 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(4 Adet)
- Devrenin çalışıp çalışmadığını kontrol eden LED
- LM 7805 Voltaj Regülatörü
- Servo Motor
- SD Card Modülü
- 16 GB SD Card
- Arduino IDE(v1.8.10)
- Çift taraflı bant
- Elektrik bandı
- Vodabond Çok Amaçlı Yapıştirıcı

## **YASAL FİZİBİLİTE**

- Açık Kaynak Kodlu Arduino IDE
- Açık Kaynak Kodlu OV7670 Kütüphanesi
- Açık Kaynak Kodlu Date Kütüphaneleri
- Açık kaynak kodlu Apache Commons Mail API (JAR)
- Açık Kaynak Kodlu Java Mail API (JAR)
- Açık Kaynak Kodlu Eclipse IDE
- Açık Kaynak Kodlu JDK 8

## **FİNANSAL FİZİBİLİTE**

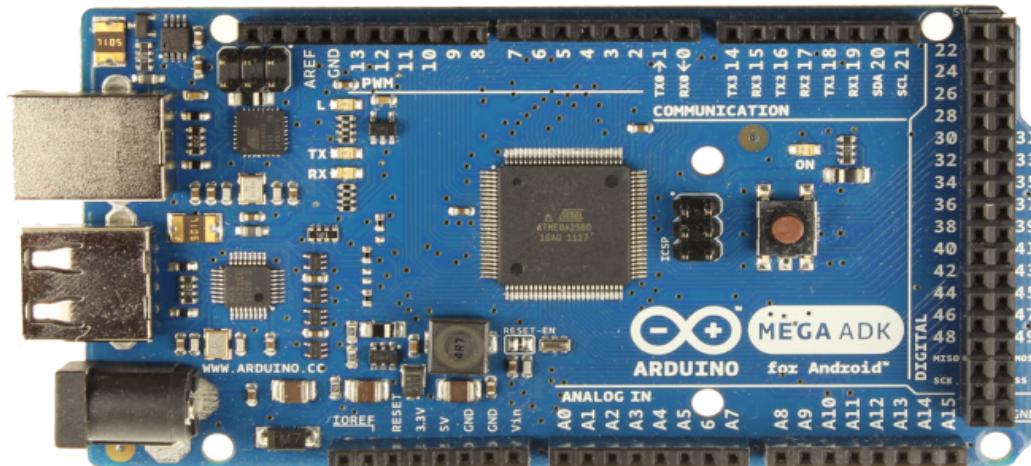
OV7670 Kamera Modülü	42,25₺
Jumper Kablolar	17,01₺ (3 farklı tip kullanıldı)
Servo Motor x3	18,00 ₺
Arduino Mega	56,00 ₺
Dirençler	2,00₺ (4 farklı direnç kullanıldı)
Breadboard	8,34 ₺ (Orta düzey yeterli görüldü)
Pir Sensör x3	20,85 ₺
Elektrik Bandı	3,00 ₺
SD Card Modülü	7,20₺
SD Card 16 GB	28,30₺
Çift Taraflı Bant	20,00₺
LM 7805 Voltaj Regülatörü	12,25₺
Vodabond Çok Amaçlı Yapıştirıcı	19, 90₺



## Temel Malzemeler

- Arduino Mega
- Arduino USB Kablosu
- Breadboard
- PIR Sensör
- OV7670
- Jumper Kablolar
- Dirençler
- Servo Motor
- SD Card Modülü
- SD Card 16GB
- 9V Pil
- 9V Pil Başlığı
- LM 7805 Voltaj Regülatörü

## Arduino MEGA



**Arduino** kolay kullanılabilir ve esnek bir yazılım/donanım mimarisine sahip, açık kaynak (open source) ailesine mensup bir elektronik geliştirme kartıdır.

**Arduino Mega** ATmega2560 mikrodenetleyici içeren bir **Arduino** kartıdır.

**Arduino**'nun en yaygın kullanılan kartı olduğu söylenebilir.

**Arduino Mega**, bilgisayarınızın USB portunu aşırı akım ve kısa devrelerden koruyan resetlenebilir bir çoklu sigortası bulunur. Çoğu bilgisayarın portları 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 Mega** barındırdığı ATmega2560 sayesinde 256 Kb lık bir hafızaya sahiptir ancak bunun 8 Kb lık kısmı Arduino bootloaderı tarafından kullanılmaktadır. Ayrıca Arduino Mega 8 Kb RAM ve 4 Kb EEPROM barındırır.

**Arduino Mega** kendini korumak için 500 mA den fazla akım çektığında kendini kapatıp açar eğer bu sırada bilgisayara bağlıysa bağlantıyı keser.

**Arduino Mega** ile programlamaya başlamadan önce bilgisayarınızda **Arduino'nun** kurulmuş olması gerekiyor .Kurulum işlemlerini tamamladıkten sonra **Arduino Mega'yı** bilgisayara tanıtmak için aşağıdaki işlemleri sırasıyla uygulayın :

**Araçlar>Kart>Arduino Mega** seçerek kartımızı belirtiyoruz.

**Araçlar>Port** seçtiğten sonra Arduino Uno'nun bağlı olduğu portu seçiyoruz.

Böylelikle **Arduino Mega**, bilgisayaramıza tanımlanmış oluyor. **Yükle** butonuna basarak programı karta yükliyoruz ve projemiz çalışmaya başlıyor.

**Arduino Mega'nın** genel anlamda özellikleri aşağıdaki gibidir :

**Mikrodenetleyici** : ATmega2560

**Çalışma Gerilimi** : 5V

**Giriş Gerilimi (önerilen)** : 7-12V

**Giriş Gerilimi (limit)** : 6-20V

**Dijital G/Ç Pinleri** : 54 (15 tanesi PWM çıkışı)

**Analog Giriş Pinleri** : 6

**Her G/Ç için Akım** : 40 mA

**3.3V Çıkış için Akım** : 50 mA

**Flash Hafıza** : 256 KB (ATmega2560)

**SRAM** : 8 KB (ATmega2560)

**EEPROM** : 4 KB (ATmega2560)

**Saat Hızı** : 16 MHz

**Uzunluk** : 101.6 mm

**Genişlik** : 53.4 mm

**Ağırlık** : 36 g

**Güç** :

**Arduino Mega**, 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 arasıdır. 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 :

**V<sub>in</sub>** : 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/Cıkış (I/O) :**

54 adet dijital, 16 adet analog giriş/cıkış 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, Serial: 0 (RX) ve 1 (TX), Serial1: 19 (RX) ve 18 (TX), Serial2: 17 (RX) ve 16 (TX), Serial3: 15 (RX) ve 14 (TX) :**

TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Pin 0 ve 1 doğrudan kart üzerinde bulunan Atmega16u2 usb-seri dönüştürücüsüne bağlıdır. Yani bilgisayardan karta kod yüklerken veya bilgisayar-mega arasında karşılıklı haberleşme yapılırken de bu pinler kullanılır. O yüzden karta kod yüklerken veya haberleşme yapılırken hata olmaması için mecbur kalınmadıkça bu pinlerin kullanılmamasında fayda vardır.

#### **Harici Kesme, 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), 21 (interrupt 2) :**

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

#### **PWM, 2-13 ve 44-46:**

8-bit çözünürlükte PWM çıkış pinleridir.

#### **SPI, 53 (SS), 51 (MOSI), 50 (MISO), 52 (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-A15 :**

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, 20 veya SDA pini ve 21 veya SCL pini:**  
I2C haberleşmesi için bu pinler kullanılır.

#### **AREF :**

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

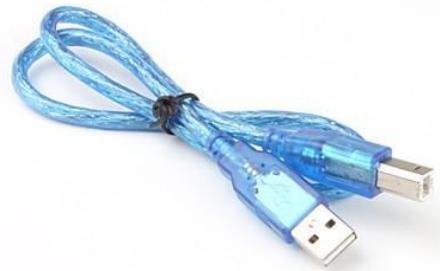
## **Arduino USB Kablosu**

**Arduino Mega** 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 Atmega2560 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-çıkar şeklinde kurabildiğimiz için kullandığımız elektronik bileşenleri başka projelerde tekrar kullanma imkanı verir.

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

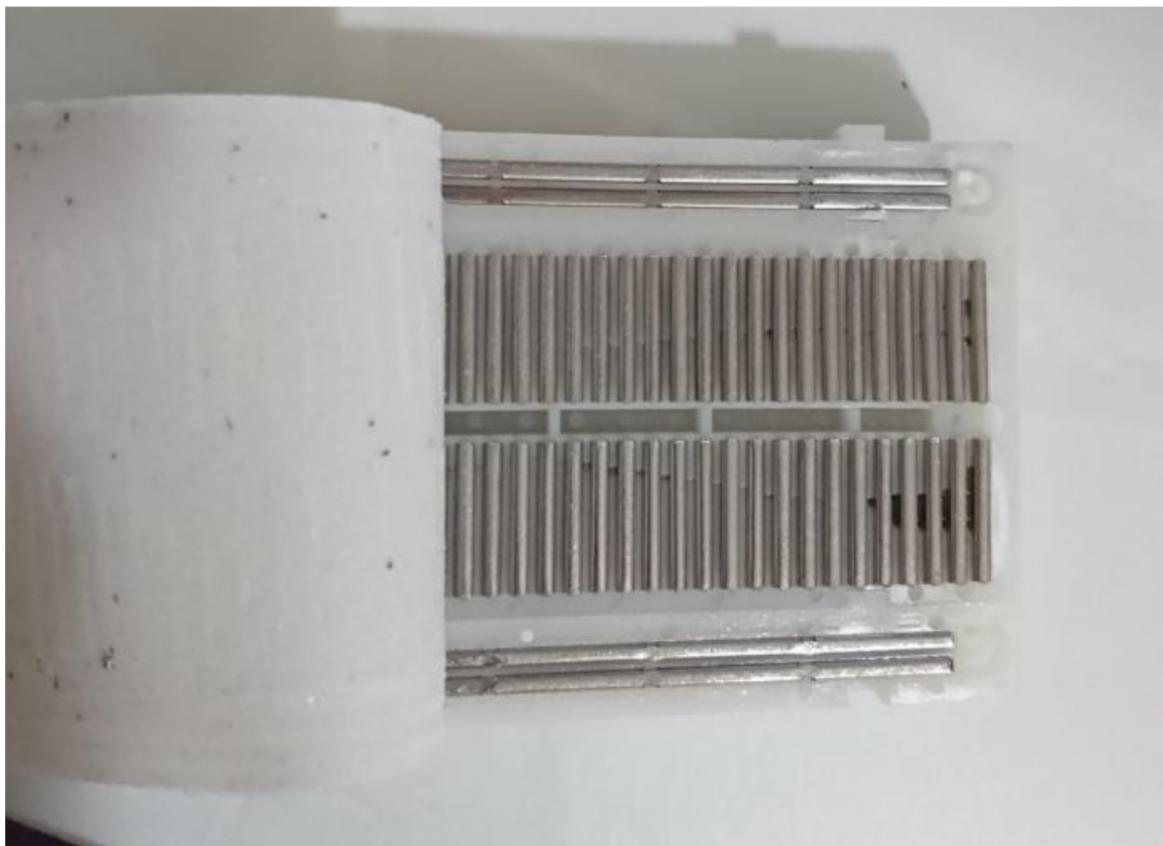
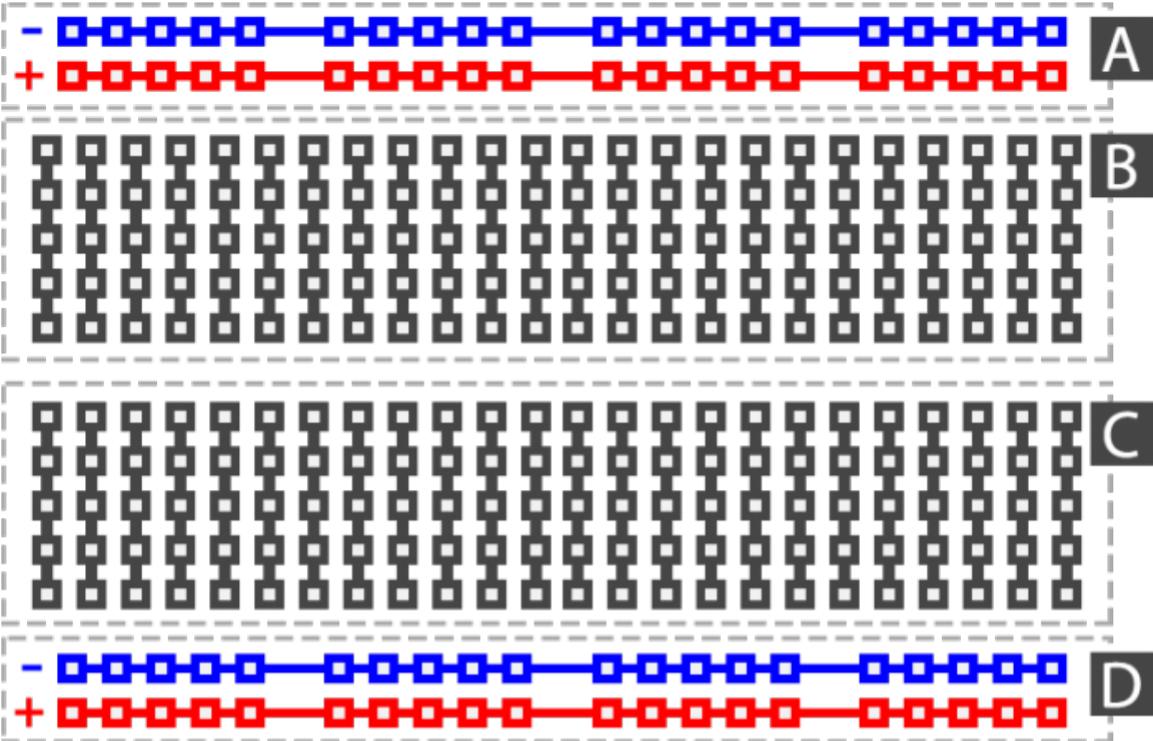
**Breadboardun** ortada kalan kısımları da sütun boyunca yerleştirilmiş iletkenlerden oluşur. Bu kısımlar da tipki satırlarda olduğu gibi **breadboardun** iki tarafında bulunur. Tüm bu iletkenlerin üstü elektronik bileşenlerin ayaklarını yerleştirmemiz için açılmış deliklerden oluşan bir plastik ile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

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

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

**Breadboard** üzerindeki + ve – şeklinde belirtilmiş satırlar gerilim bağlantılarını sağlamak için kullanılır. Gerilim bağlantılarını sağladıkten 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 üzerindeki 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 ayıktır, yani iletim halinde değillerdir.



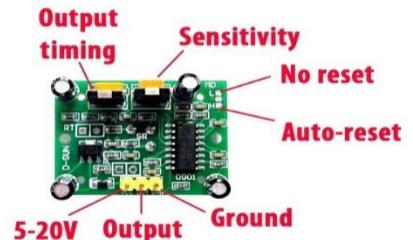
## PIR Sensor (Hareket Sensörü)

**PIR Sensörleri**, bir ortamda oluşan canlı hareketini algılamak için kullanılan sensörlerdir. Bu minik boyutlu sensör, çeşitli elektronik, robotik ve hobi uygulamalarında rahatça kullanabileceğiniz, **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 çözümüdür.

### Özellikleri:

- **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



## Teknik Özellikleri:

- Ürünün VGA görüntü kalitesi: 30 kare/saniye.
- Optik boyutu: 1/6",
- Çözünürlük: 0.3 megapixsel. 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

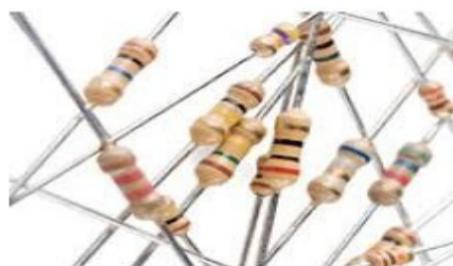
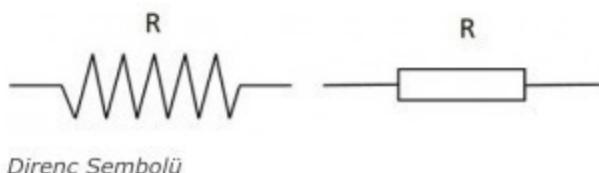
icin oldukça kullanışlıdır. Uçlarında erkek ve dişi girişlerin bulunmasına göre 3 çeşit **jumper kablo** bulunmaktadır :

- 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 simbolü 2 farklı şekilde gösterilebilir :



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

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

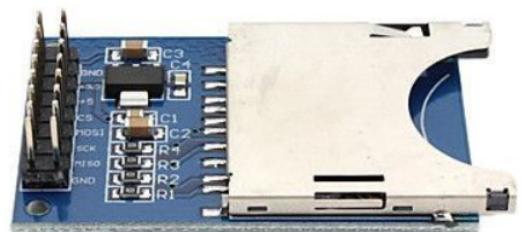
basamak 3. Şerit :

Çarpan katsayısı

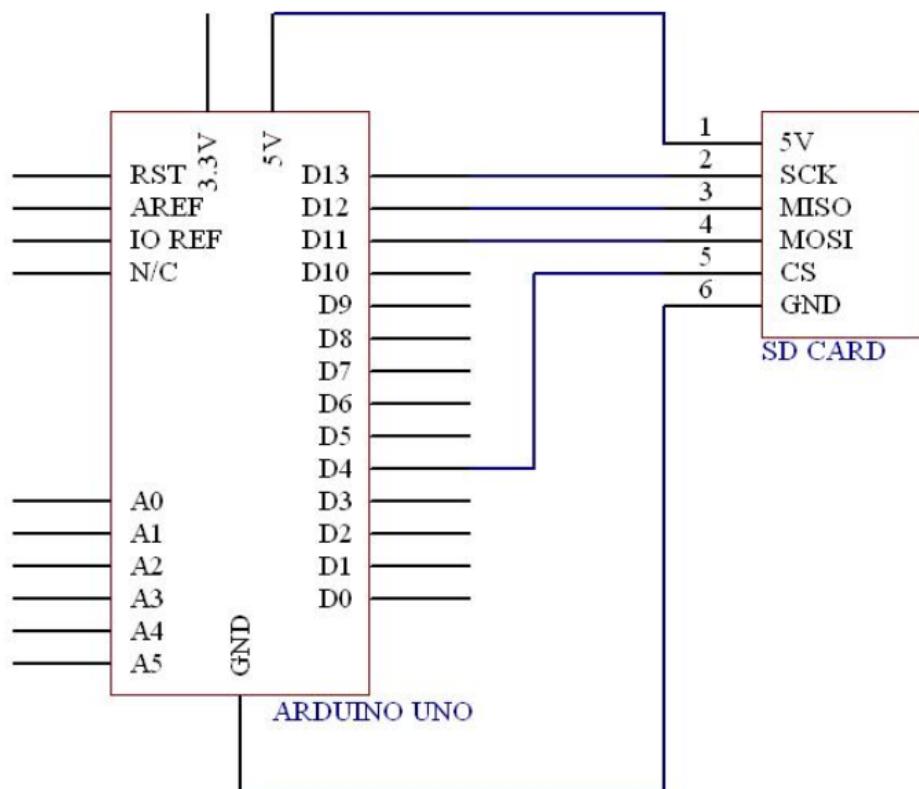
4. Şerit : Tolerans

## SD Kart Modülü

SD kart modül, SPI protokolü üzerinden SD kartlara okuma ve yazma yapabileceğiniz oldukça uygun fiyatlı ve kullanışlı bir ürünüdür. Arduino başta olmak üzere birçok mikrodenetleyici platformu üzerinden rahatlıkla kullanabileceğiniz modüle SD hafıza kartlarını direk, microSD hafıza kartlarını ise adaptör yardımı ile takabilirsiniz.



Bilgi saklama, okunan bilgileri hafıza kartına yükleme, sürekli gelen bilgileri kayıt altına alma gibi birçok uygulamada kullanılabilmektedir. Kart üzerinde dahili 3.3V voltaj regülatörü de bulunmaktadır. Bu sayede 3.3V'luk ve 5V'luk sistemlerle rahatlıkla kullanılabilmektedir.



## **9 V Pil Başılığı**

9V pillerin bağlantı kablosudur.



## **9 V Pil**

Güç gerektiren elektronik cihazlarda kullanılan standart pillerdendir. Şarj edilebilir nikel-kadmiyum pili (NiCd) ve nominal 9V dereceli Nikel-metal hidrit pil (NiMH) altı ila sekiz arasında 1,2 voltlu hücrelere sahiptir.



## **LM 7805 Voltaj Regülatörü**

**LM 7805**, sabit 5V çıkış veren, üç bacaklı bir pozitif voltaj regülatöründür. Sabit voltaj kaynağı olması dışında, uygun devre ile minimumu 5V olan ayarlanabilir voltaj kaynağı olarak da kullanılabilir.



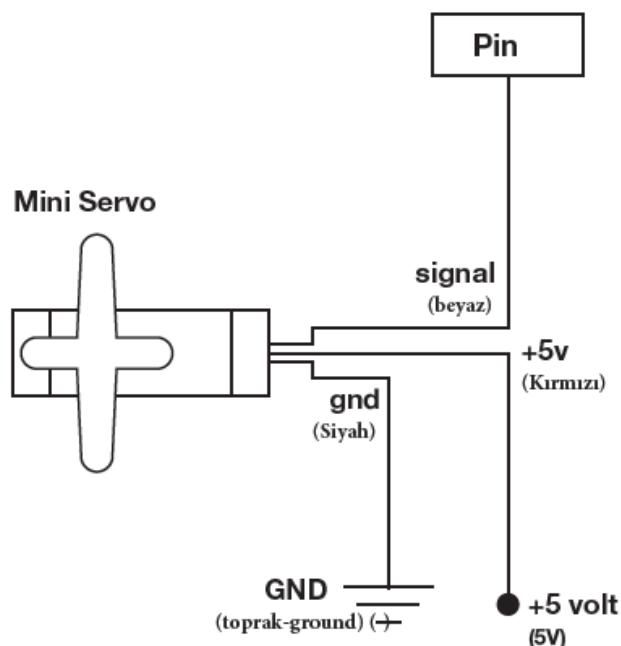
## Servo Motor

Servo motor 0 ile 180 derece arasında 1 derece hassasiyetle dönebilen bir motor çeşididir. Tam tur atamazlar. Genellikle robot kol gibi tam tur dönmesine gerek olmayan, hassas açılı yerlerde kullanılır. Servo motor içerisinde bir adet DC motor bulunur. DC motorun ucuna bağlı dişli sisteminin yardımıyla servo mili daha fazla yük kaldırıbmaktadır. Bu işlem sırasında servonun dönüş hızı da yavaşlamış olur. Kullanılan dişli sistemine göre servo motorların kaldırabileceği yük değişir.



Servo motorlar entegre elektronik uygulamalar için son derece idealdir. Dönen normal motorların aksine istenilen herhangi bir yönde dönebilir. Robot teknolojilerinde en çok kullanılan motor çeşididir. Servoya ulaşan voltaj değişimine göre servo belirli bir pozisyon'a girer. Örneğin 20 milisaniyelik bir değişim ile servo 90 derece hareket ettilerilebilir. Bu uygulamada, servoyu kontrol edip yönlendirmek için Arduino'nun PWM pinlerini (~3, ~5, ~6, ~9, ~10, ~11 numaralı dijital pinler) kullanacağız.

Servo motorun şemasından anlaşılacağı üzere beyaz renkli kablo PWM pine, kırmızı renkli kablo VCC pinine, siyah renkli kablo ise GROUND (GND) toprak hattına bağlanmalıdır.



Servo Motor Şeması

## **OTG Kablosu**

OTG, On The Go kelimelerinin baş harflerinden oluşan bir kısaltmadır. Akıllı cep telefonları ve tabletler, Arduino gibi mini bilgisayarlarda yer alan ve genellikle mikro-USB biçiminde bulunan şarj ve data aktarımı için kullanılan portu, cihaza USB host portu olarak tanıtan kabloya USB OTG kablosu ismi verilir. Günümüzde Android işletim sistemi çalıştırılan akıllı telefon ve tabletlerin neredeyse tamamı bu tip bağlantıyı destekleyen portlara sahiptir. Arduino gibi kartlarda ise tam boy USB portu yerine OTG destekli mikro-USB portları kullanılarak boyutlardan tasarruf yapılması amaçlanmıştır. SD Card okuyucu bu kablo üzerinden direk telefonda incelenebilir hale getirilebilir.

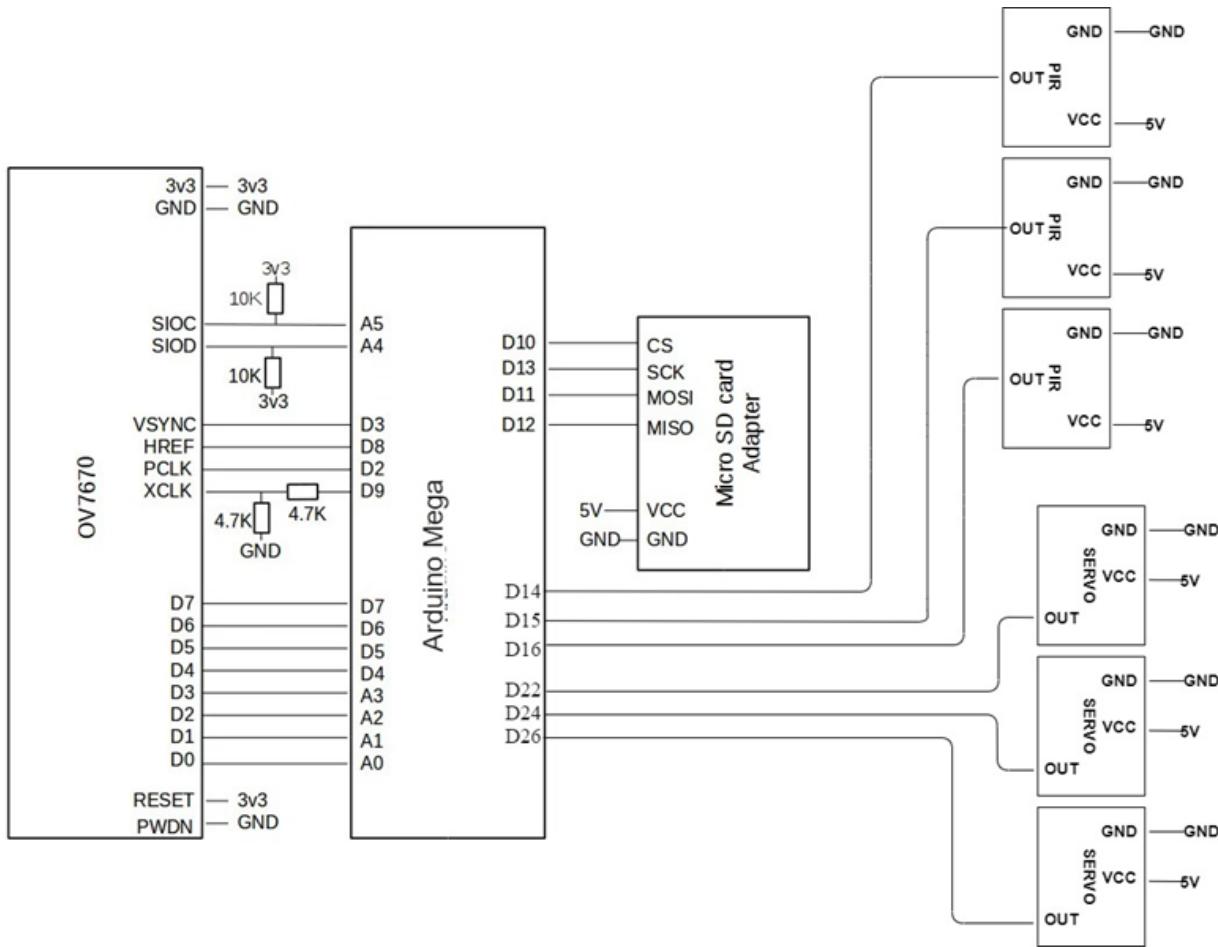


## **USB Uzatma Kablosu**

Arduino kartlarınızı ya da benzer kartlarınızı program yüklemek için kullanabileceğiniz bir tarafı type-A diğer tarafı dışı bir USB kablosudur. Arduino kablosunun boyunu uzatmak için kullanılabilen opsiyonel bir kablodur.



## DEVRENİN BAĞLANMASI



## ARDUINO KODU 1 (İkinci Dönem Projesi Kodları)

/\* Libraries.h dosyası uzunluğundan dolayı teze eklenmemiştir dosyaya  
[https://github.com/azatates7/SecondTermProject/Libraries.h linkinden erişebilirsiniz.](https://github.com/azatates7/SecondTermProject/Libraries.h) \*/

```
#include <C:\Users\KaanPC\Desktop\SecondTermProject\Libraries.h>
#include <Arduino.h>
#include <Wire.h>
#include <SD.h>
#include <Servo.h>
#include <SPI.h>
#include <user_config.h>
#include <time.h>
#include <TimeLib.h>
#include <inttypes.h>

int pirPin1 = 14;
int pirPin2 = 15;
int pirPin3 = 16;
int pirValue1;
int pirValue2;
int pirValue3;

Servo leftServo;
Servo centerServo;
Servo rightServo;
int pos = 0; // keep servo position value
int leftServoPin = 22;
int centerServoPin = 24;
int rightServoPin = 26;

void setup()
{
    leftServo.attach(leftServoPin);
    centerServo.attach(centerServoPin);
    rightServo.attach(rightServoPin);
```

```

rtc.begin();
rtc.autoTime();
if (!rtc.isrunning())
{
    Serial.println("Clock is not running");
}

pinMode(pirPin1, INPUT);
pinMode(pirPin2, INPUT);
pinMode(pirPin3, INPUT);
noInterrupts(); //Disable all interrupts
XCLK_SETUP(); //Setup 8MHz clock at pin 11
OV7670_PINS(); // Setup Data-in and interrupt pins from camera

delay(1000);

TWI_SETUP(); // Setup SCL for 100KHz
interrupts();
Wire.begin();
Init_OV7670();
Init_QVGA();
Init_YUV422();
WriteOV7670(0x11, 0x1F); //Range 00-1F
noInterrupts();
Serial.begin(9600);
pinMode(CS_Pin, OUTPUT);
SD.begin(CS_Pin);
}

void loop()
{
int leftPir = digitalRead(pirPin1); // read pin1 value
int centerPir = digitalRead(pirPin2); // read pin2 value
int rightPir = digitalRead(pirPin3); // read pin3 value
Serial.println(leftPir);
Serial.println(centerPir);
Serial.println(rightPir);

if(centerPir == HIGH)
{
    QVGA_Image(print_time() + ".bmp"); // Capture image
    delay(2000);
}
}

```

```
else
{
    if(leftPir = HIGH)
    {
        rightServo.write(90);
        centerServo.write(90);
        QVGA_Image(print_time()+" bmp"); // Capture image
        delay(2000);
        centerServo.write(270);
        rightServo.write(90);
    }

else
{
    leftServo.write(90);
    centerServo.write(270);
    QVGA_Image(print_time()+" bmp"); // Capture image
    delay(2000);
    centerServo.write(90);
    leftServo.write(90);
}
}

delay(1000);
while(1);
}
```

## ARDUINO KODU 2

### (İlk Dönem Kodunun İkinci Dönem Projesine Göre Düzenlenmiş Hali)

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

```
#include <C:\Users\DK\Desktop\FirstTermProject-master\camera\OV7670.h>

int pirPin1 = 14;
int pirPin2 = 15;
int pirPin3 = 16;
int pirValue1;
int pirValue2;
int pirValue3;

Servo leftServo;
Servo centerServo;
Servo rightServo;
int pos = 0; // keep servo position value
int leftServoPin = 22;
int centerServoPin = 24;
int rightServoPin = 26;

void setup(){

    leftServo.attach(leftServoPin);
    centerServo.attach(centerServoPin);
    rightServo.attach(rightServoPin);

    pinMode(pirPin1, INPUT);
    pinMode(pirPin2, INPUT);
    pinMode(pirPin3, INPUT);

    Serial.begin(9600);
    ArduinoUnoInut();
    camInit(); setResolution();
    setColor();
    writeReg(0x11, 10);

}
```

```

void loop(){
    int leftPir = digitalRead(pirPin1); // read pin1 value
    int centerPir = digitalRead(pirPin2); // read pin2 value
    int rightPir = digitalRead(pirPin3); // read pin3 value
    Serial.println(leftPir);
    Serial.println(centerPir);
    Serial.println(rightPir);

    if(centerPir == HIGH)
    {
        QVGA_Image(print_time()+" bmp"); // Capture image
        delay(2000);
    }

    else
    {
        if(leftPir == HIGH)
        {
            rightServo.write(90);
            centerServo.write(90);
            captureImg(320, 240);
            delay(2000);
            centerServo.write(270);
            rightServo.write(90);
        }
        else
        {
            leftServo.write(90);
            centerServo.write(270);
            captureImg(320, 240);
            delay(2000);
            centerServo.write(90);
            leftServo.write(90);
        }
    }

    delay(1000);
    while(1);
}

```

## JAVA KODU

### (İlk Dönem Kodunun İkinci Dönem Projesine Göre Düzenlenmiş Hali)

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

#### **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 UnsupportedCommOperationException {
        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("kaantunahanyagan@gmail.com.tr");
mailaddress.add("karagozuniquedk@gmail.com");
mailaddress.add("azatates4977@gmail.com");
```

```

for (String mail : mailaddress) {

    Date dt = new Date();

    HtmlEmail email = new HtmlEmail(); // email nesnesi yaratlıyor
    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 "+dt);

}

}

}

catch (IOException | EmailException | PortInUseException ex) {
    System.out.println("Hata Algılandı : " + ex.getMessage());
}

}

```

```
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**

**\*\*\* = Arduino ya kodu yüklemek için. Arduino başka kod yüklenmeyecekse tekrar yapmaya gerek yoktur.**

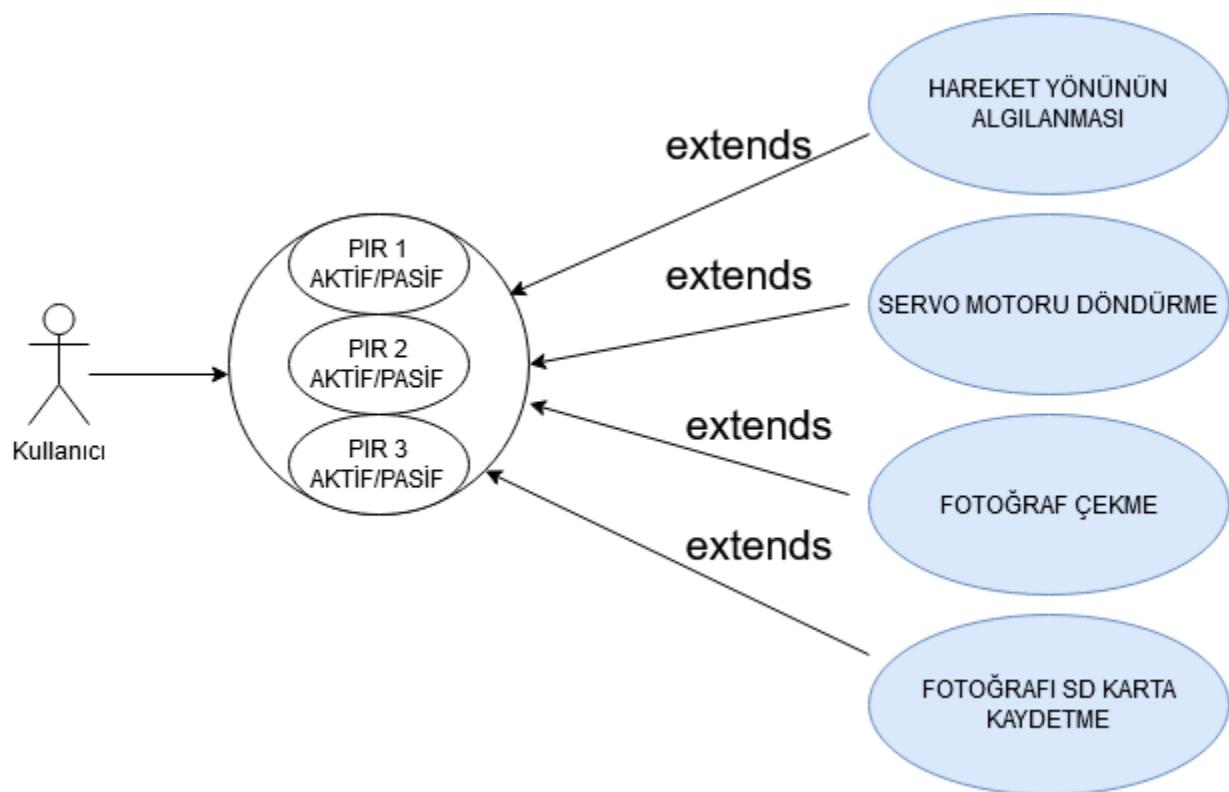
**%% = 2. Kod İçin.**

- **Kabloyu bilgisayara tak, Arduino IDE 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. ( %% )**
- **SimpleRead class'ını çalıştır. (%%)**
- **Hata alırsa kabloyu çıkarıp tekrar takınız(Port Çakışması) (%%)**
- **PIR Sensörü devreye girdikçe Kamera Servo motor aracılığıyla hareket yönüne döner görüntüyü alır ve SD Kart a kaydeder. Baş konuma geri döner.**
- **Java kodu aracılığıyla. bmp formatında bilgisayara kaydedilir.(%%)**
- **Kaydedilen her resim belirlenen mail adreslerine toplu olarak yollanır.(%%)**
- **Java Kodu kapatılana kadar çalışma sürer.(%%)**

# DİYAGRAMLAR

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

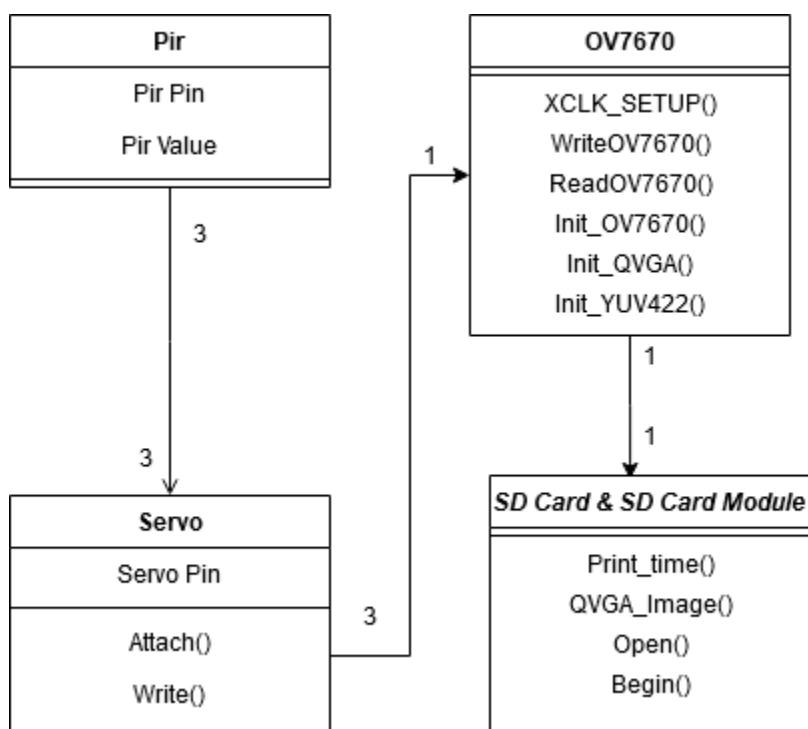
## USE CASE DİYAGRAMI



## KULLANIM SENARYOLARI

SENARYO KSI	PIR AKTİF / PASİF
Birinci Aktör	Kullanıcı
İlgililer ve Beklentileri	Pili Arduino ya bağlı ve kodlar Arduino ya tanımlı olmalı
Ön Koşullar	Sistemin hareket alması, Servo nun kamerayı hareket ettirmesi
Son Koşullar	Yok
Ana Akış	1-Pili tak 2- Sistemi başlat 3-Hareket almasını sağla.
Alternatif Akış	Hareket algılanmazsa çalışmaz. Port hatası ortaya çıkabilir. Pil gücü yetersiz olabilir Fotoğraf olmadığı için SD Kart işlemi olmaz.

## SINIF DİYAGRAMI



**Referanslar 2017-03-22 fiz 9 mar malz 23 mar**

**A Guide to Arduino Based Video Camera - Open Electronics - Open Electronics**

<https://www.open-electronics.org/a-complete-guide-to-arduino-based-video-camera/>

Accessed: **2020-03-23**

**Absolute Beginner datetime and string manipulation: arduino**

[https://www.reddit.com/r/arduino/comments/67fv66/absolute\\_beginner\\_datetime\\_and\\_string\\_manipulation/](https://www.reddit.com/r/arduino/comments/67fv66/absolute_beginner_datetime_and_string_manipulation/)

Accessed: **2020-05-20**

**Arducam OV7670 0.3 Megapixel Camera Module for Arduino Boards - Arducam**

<https://www.arducam.com/product/arducam-ov7670-0-3-megapixel-camera-module-for-arduino-boards/>

Accessed: **2020-03-23**

**Arduino Camera (OV7670) Tutorial | Microcontroller Tutorials**

<https://www.teachmemicro.com/arduino-camera-ov7670-tutorial/>

Accessed: **2020-03-30**

**Arduino Camera Interfacing OV7670**

<https://www.slideshare.net/SomnathSharma4/arduino-camera-interfacing-ov7670>

Accessed: **2020-03-23**

**Arduino: Servo Motor Kontrolü**

<http://arduinocucuk.blogspot.com/2014/11/servo-motor-kontrolu.html>

Accessed: **2020-05-18**

**Arduino Playground - DateTime**

<https://playground.arduino.cc/Code/DateTime/>

Accessed: **2020-05-20**

**Basics: Project 052a OV7670 camera sensor module without FIFO RAM at Acoptex.com /**

**ACOPTEX.COM**

<http://acoptex.com/project/254/basics-project-052a-ov7670-camera-sensor-module-without-fifo-ram-at-acoptexcom/#shash.x99DafwZ.z4dVcyWw.dpbs>

Accessed: **2020-03-23**

**Direnç Nedir? Direnç Renk Kodları, Hesaplama ve Çeşitleri**  
<https://maker.robotistan.com/direnc/>  
Accessed: **2020-05-18**

**Direnç Renk Kodları ile Direnç Hesaplama Aracı**  
<http://devreokulu.com/DirencHesaplama.html>  
Accessed: **2020-05-18**

**Electronics/SDcard\_Test.ino at master · hardikkalasua/Electronics · GitHub**  
[https://github.com/hardikkalasua/Electronics/blob/master/OV7670\\_SDcard/Arduino%20Code/SDcard\\_Test.ino](https://github.com/hardikkalasua/Electronics/blob/master/OV7670_SDcard/Arduino%20Code/SDcard_Test.ino)  
Accessed: **2020-03-30**

**Embedded Programmer: Hacking the OV7670 camera module (SCCB cheat sheet inside)**  
<http://embeddedprogrammer.blogspot.com/2012/07/hacking-ov7670-camera-module-sccb-cheat.html>  
Accessed: **2020-03-30**

**Fizibilite raporu nedir, nasıl hazırlanır? | MikroX Blog**  
<https://blog.mikrox.com.tr/fizibilite-raporu-nedir-nasil-hazirlanir/>  
Accessed: **2020-03-09**

**Getting Date and Time with ESP32 (NTP Client) | Random Nerd Tutorials**  
<https://randomnerdtutorials.com/esp32-ntp-client-date-time-arduino-ide/>  
Accessed: **2020-05-20**

**GitHub - ArduCAM/Arduino: This is ArduCAM library for Arduino boards**  
<https://github.com/ArduCAM/Arduino>  
Accessed: **2020-03-23**

**GitHub - metinakkin/Ov7670**  
<https://github.com/metinakkin/Ov7670>  
Accessed: **2020-03-23**

**How to Use OV7670 Camera Module with Arduino Uno**  
<https://circuitdigest.com/microcontroller-projects/how-to-use-ov7670-camera-module-with-arduino>  
Accessed: **2020-03-23**

**OV7670 Arduino Camera Sensor Module Framecapture Tutorial: 11 Steps (with Pictures) - Instructables**

<https://www.instructables.com/id/OV7670-Arduino-Camera-Sensor-Module-Framecapture-T/>

Accessed: **2020-03-23**

**OV7670 camera and arduino · Lulu's blog**

<https://lucidar.me/en/arduino/camera-ov7670-and-arduino/>

Accessed: **2020-03-23**

**OV7670 Camera Module For Arduino - HUB360**

<https://hub360.com.ng/product/ov7670-camera-module-for-arduino/>

Accessed: **2020-03-23**

**OV7670 camera module | maartentech**

<https://maartentech.wordpress.com/arduino-projects/ov7670-camera-module/>

Accessed: **2020-03-23**

**OV7670+Arduino Uno+SD module – HKalasua**

<https://hkalasua.wordpress.com/2017/09/11/ov7670-arduino-sd/>

Accessed: **2020-03-23**

**Print a timestamp to SD card on Arduino · GitHub**

<https://gist.github.com/dhhagan/f1cca00cdb479558b6d1>

Accessed: **2020-03-23**

**Started from 2014: How to use OV7670 camera module with Arduino?**

<http://kathyjan.blogspot.com/2014/12/how-to-use-ov7670-camera-module-with.html>

Accessed: **2020-03-23**

**Using OV7670 Camera Sensor With Arduino - Electronics-Lab**

<https://www.electronics-lab.com/using-ov7670-camera-sensor-arduino/>

Accessed: **2020-03-23**