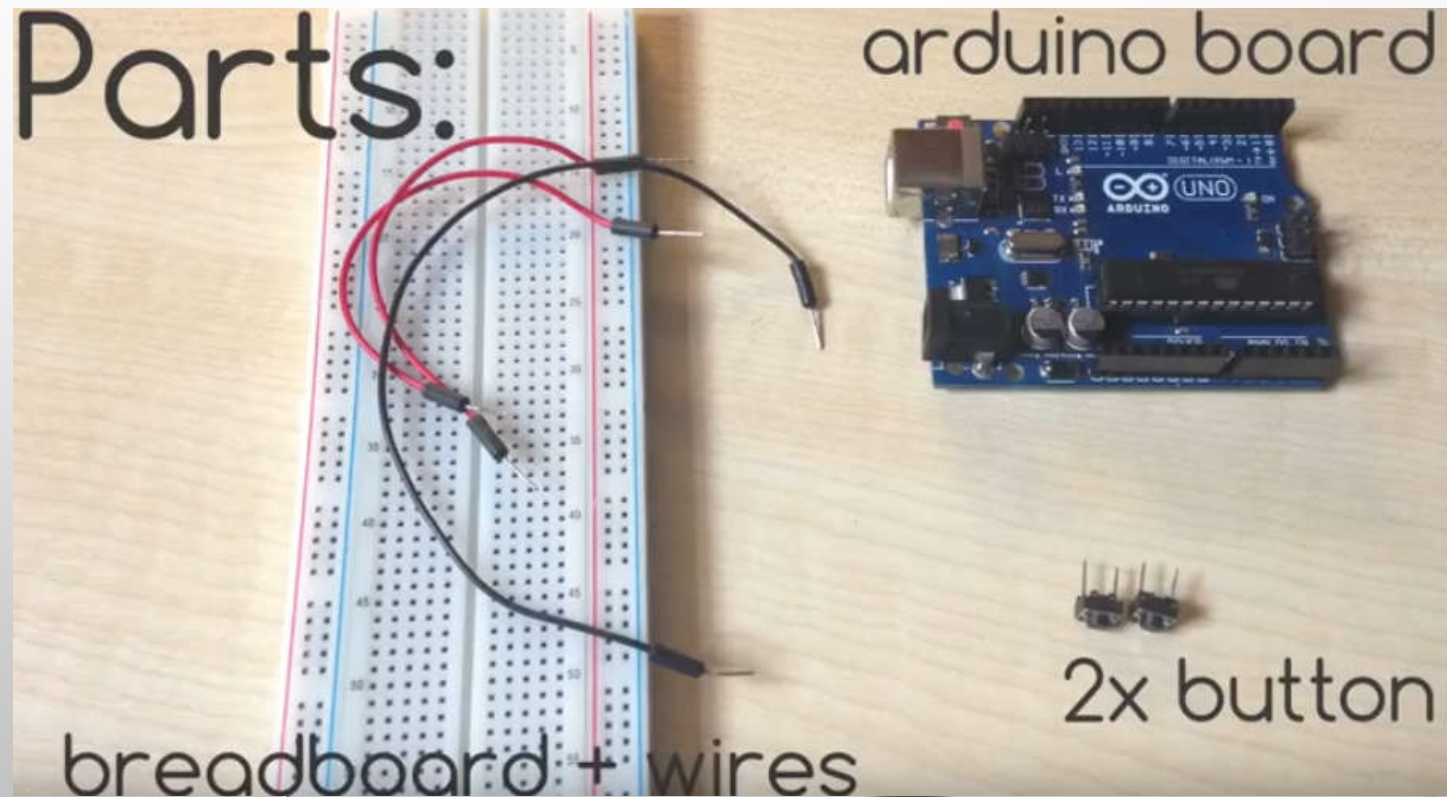


UNITY GÜNLÜĞÜ ARDUINO KONTROL BUTTON KONTROL POT KONTROL LDR KONTROL

UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

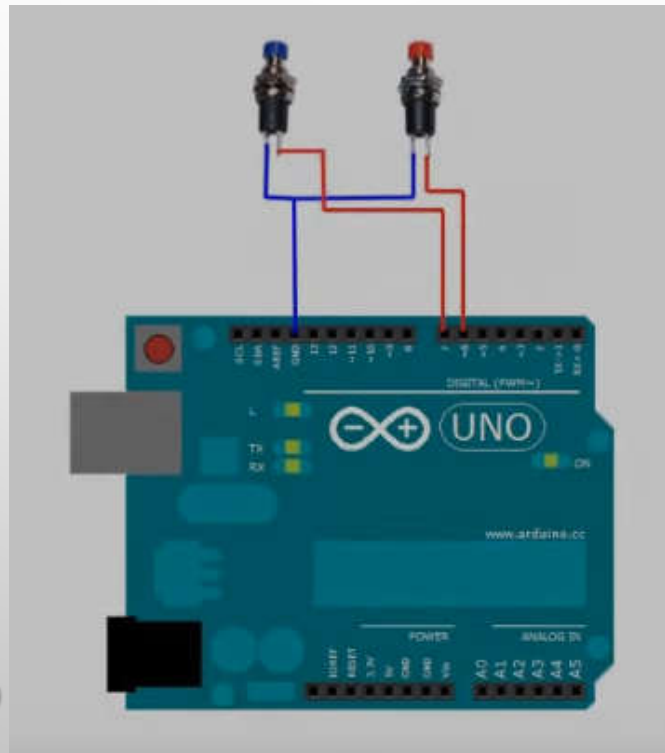
Düzenek için gerekli parçalar



UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

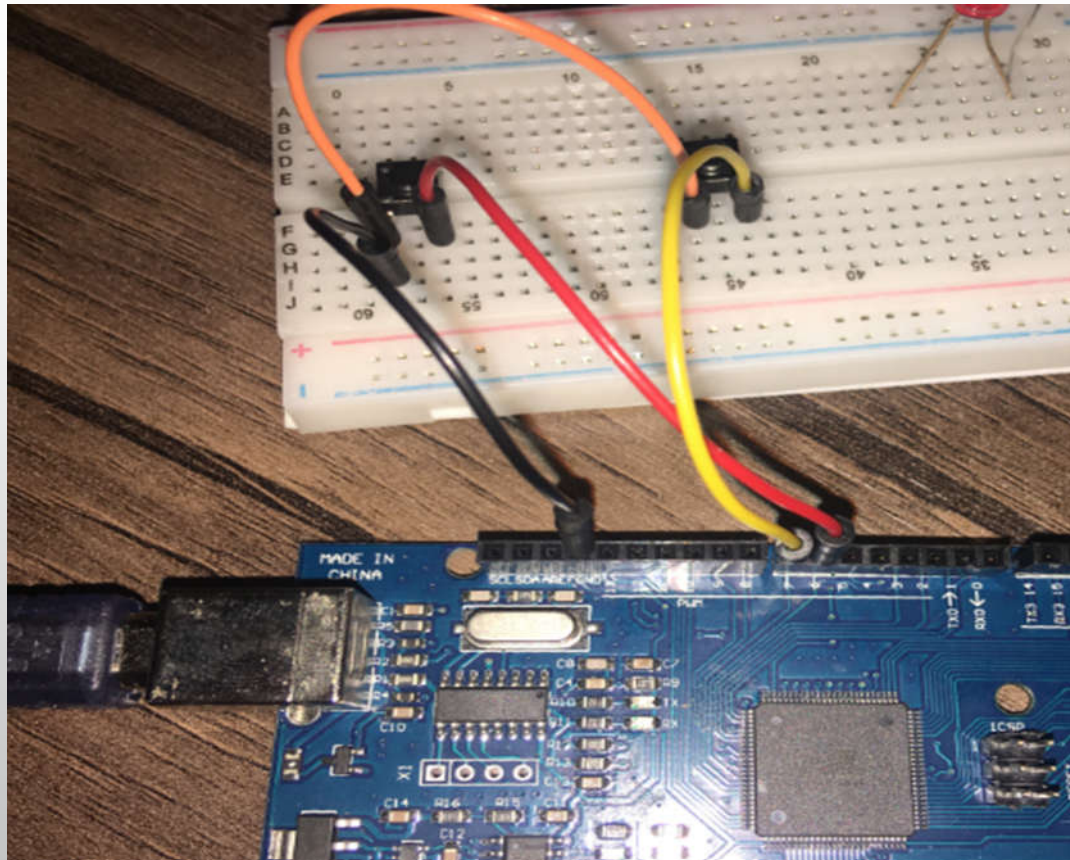
Buttonları bağlamak için 6 ve 7 nolu dijital girişleri belirliyoruz.

```
const int butPin1=6; //sol buton  
const int butPin2=7; //sag buton
```



UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Devreyi kuruyoruz. Butonların bir bacağı ortak bağlanıp, GND'a bağlandı. Diğer bacakları sıra ile Digital 6 ve Digital 7 pinlerine bağlandı.



UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Arduino için yazılacak script;

```
const int butPin1=6; //sol  
const int butPin2=7; //sag
```

```
void setup() {
```

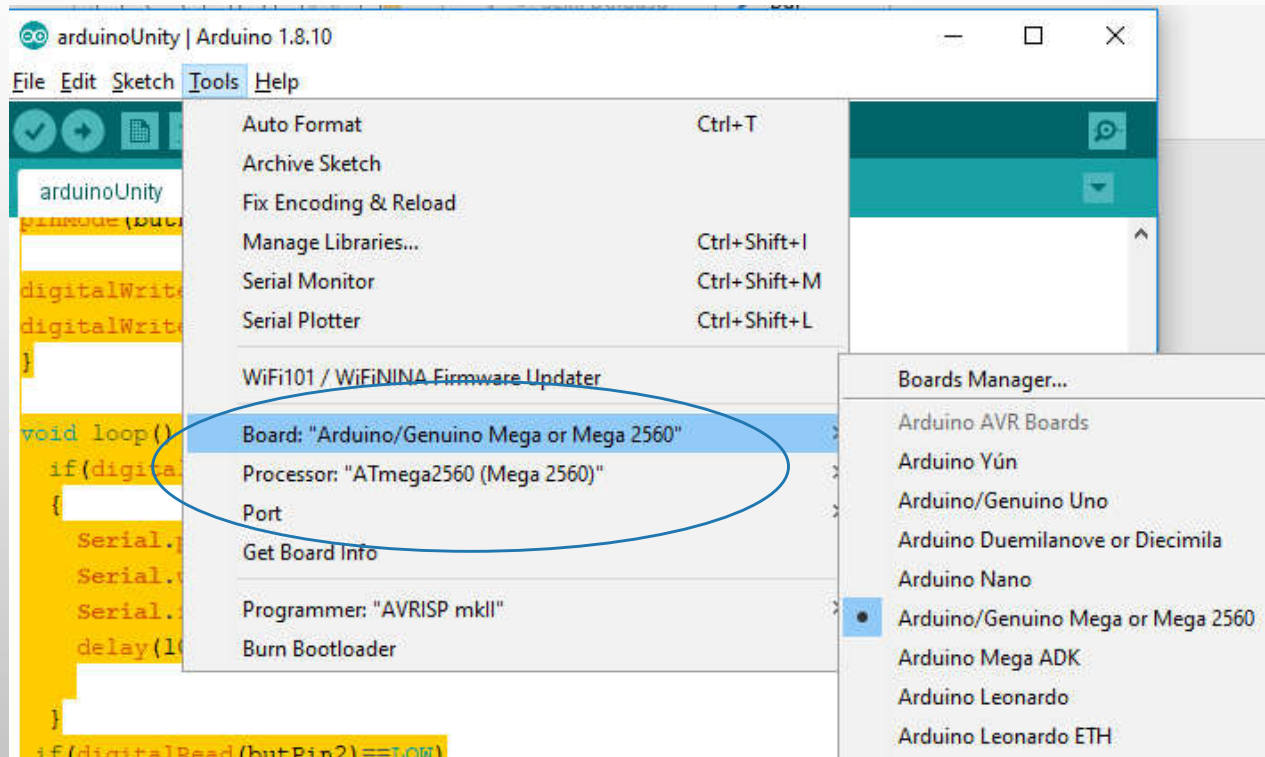
```
  Serial.begin(9600);  
  pinMode(butPin1, INPUT);  
  pinMode(butPin2, INPUT);
```

```
  digitalWrite(butPin1, HIGH);  
  digitalWrite(butPin2, HIGH);  
}
```

```
void loop() {  
  if(digitalRead(butPin1)==LOW)  
  {  
    Serial.println("LEFT");  
    Serial.write(1);  
    Serial.flush();  
    delay(10);  
  }  
  if(digitalRead(butPin2)==LOW)  
  {  
    Serial.println("RIGHT");  
    Serial.write(2);  
    Serial.flush();  
    delay(10);  
  }  
}
```

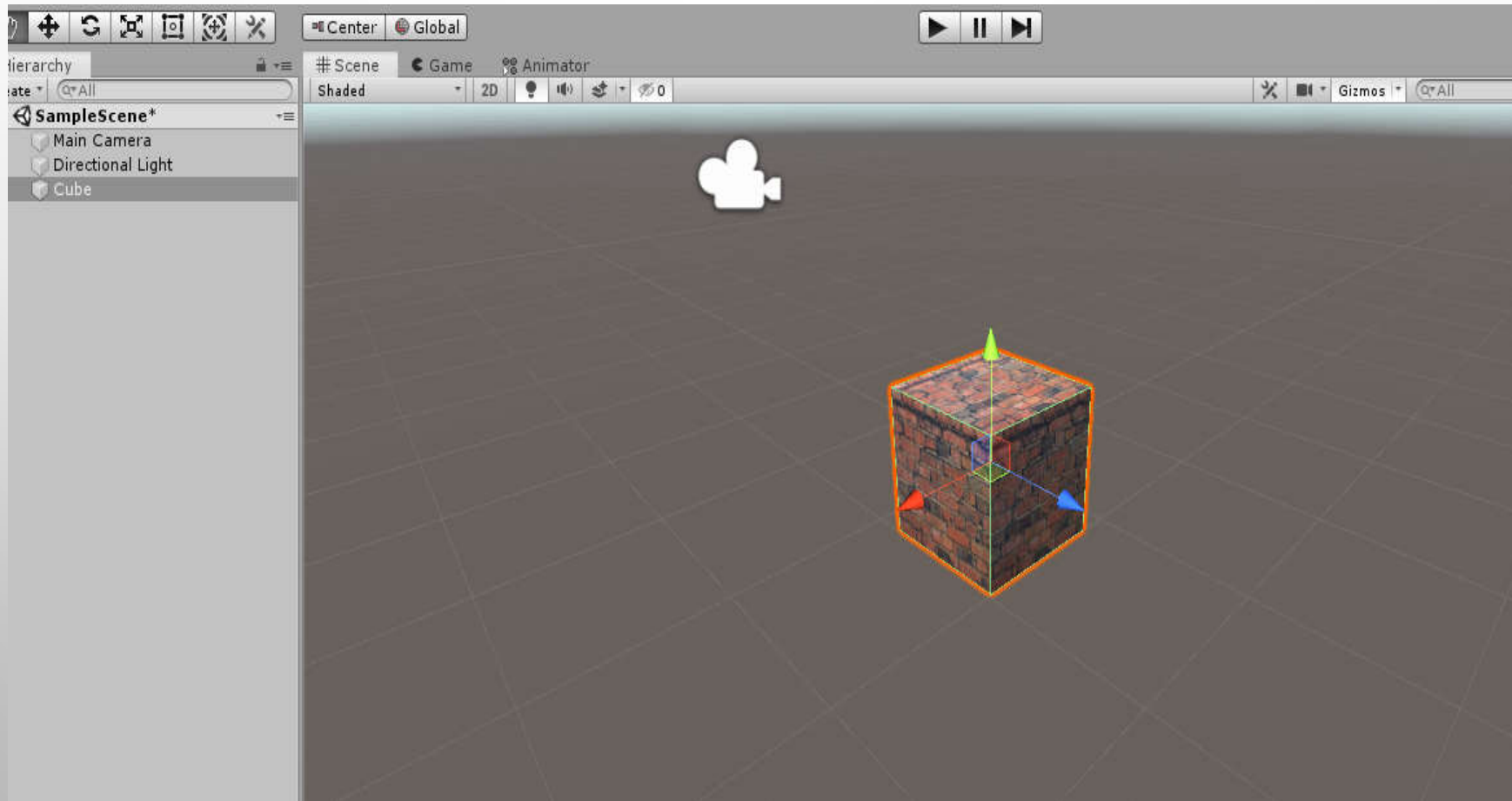

UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Arduino kodunu yüklemek için kullanılan arduino kartı ve Port seçilir. Upload ile kod Arduino içine gömülür ve çalışmaya başlar.



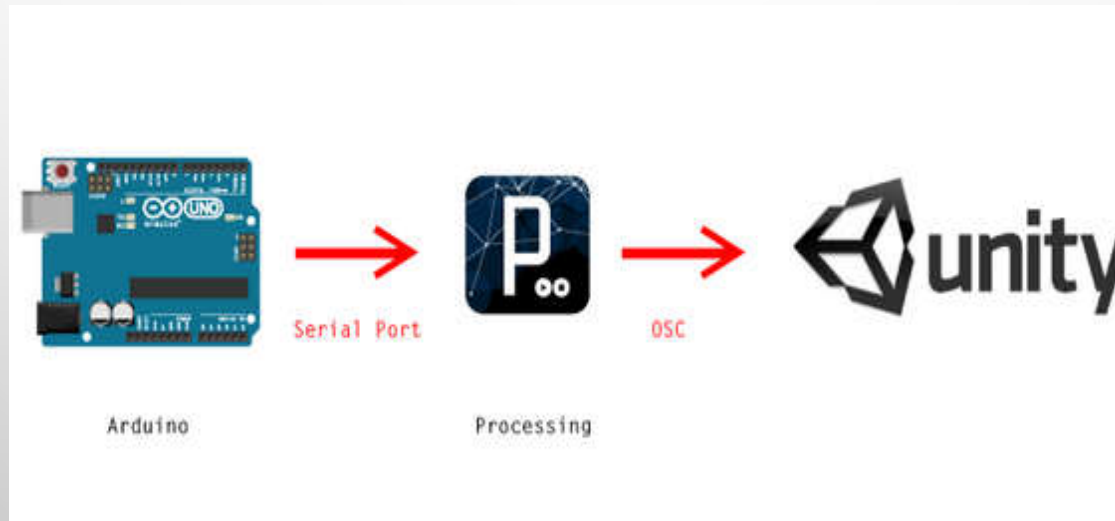
UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Unity için yeni bir proje açıp projeye bir küp ekliyoruz.



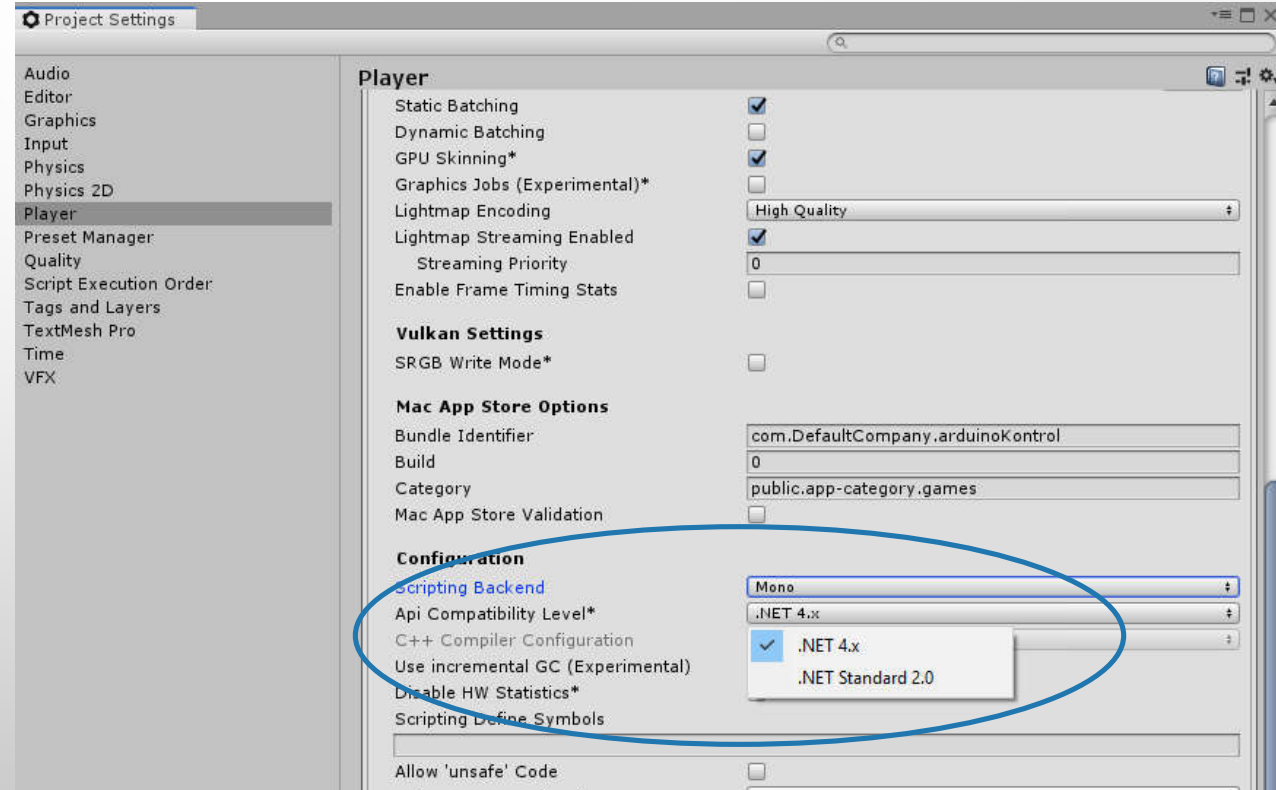
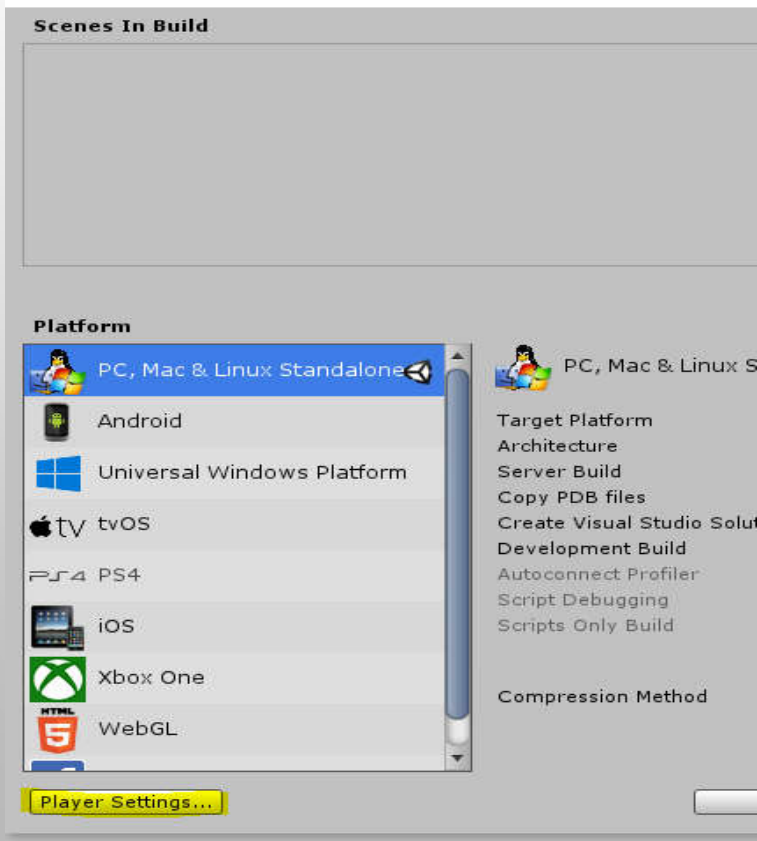
UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Arduino'dan okunan buton değerini Seri bağlantı aracılığı ile Unity içine çekip, seri yazdırılan bu değer Unity içindeki bir C# kodu ile elde edilir. Gelen bilgiye göre Unity sahnesinde nesne istenilen şekilde hareket ettirilir yada döndürülür



UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Unity içinde Seriport kullanımı için API .NET setinin ayarlanması gereklidir. Proje Ayarları içinde Player Settings kısmındaki API uyumluluğu kısmından uygun .NET seti seçilir.



UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Ayarlar yapıldıktan sonra, yeni bir C# kodu eklenir.

```
using UnityEngine;
using System.IO.Ports;

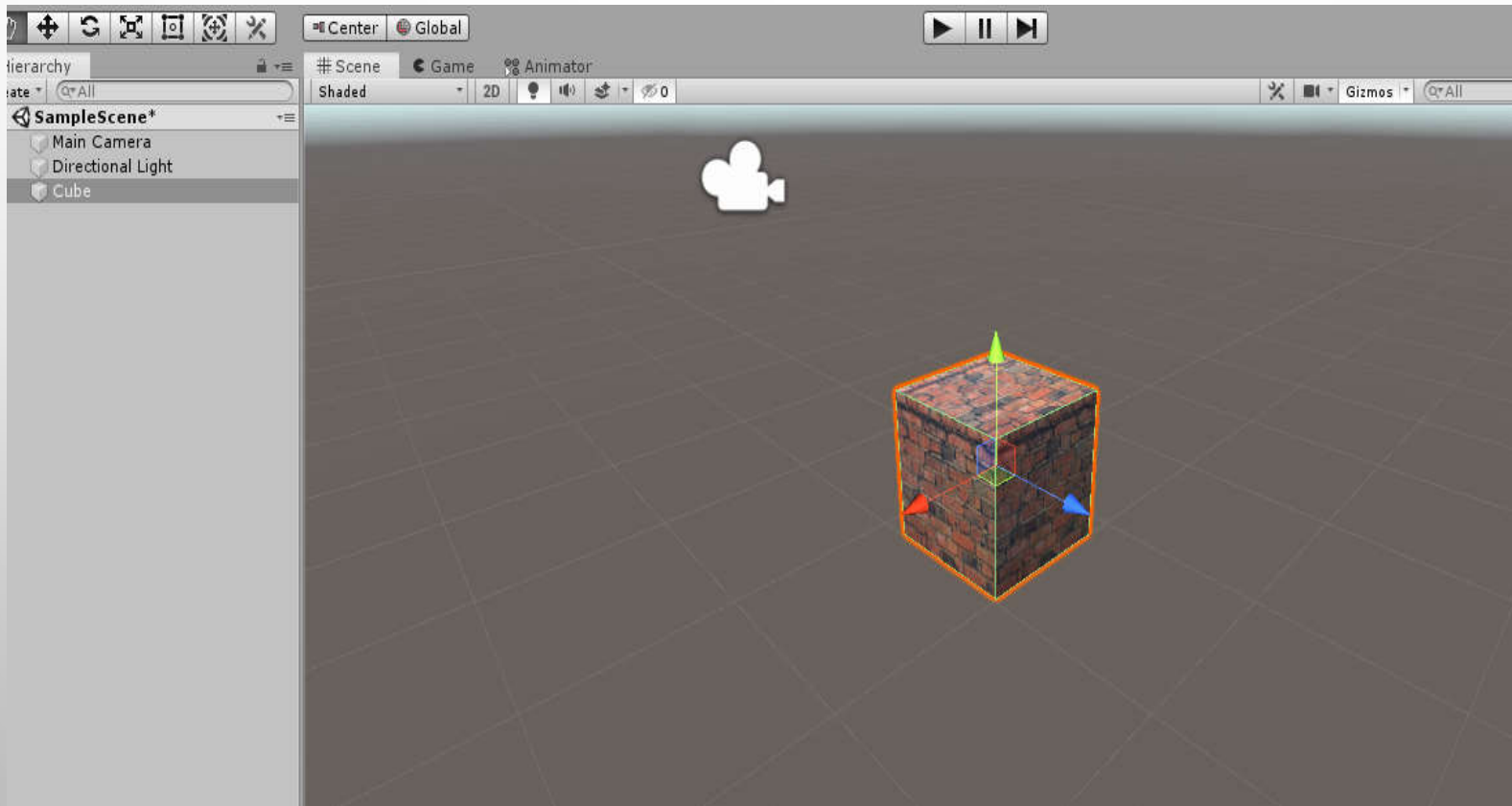
public class arduinoKontrol : MonoBehaviour
{
    SerialPort sp = new SerialPort("COM12", 9600);
    public int hiz = 5;

    void Start()
    {
        sp.Open();
        sp.ReadTimeout = 1;
    }

    void Update()
    {
        if(sp.IsOpen)
        {
            try
            {
                if(sp.ReadByte()==1)
                {
                    transform.Rotate(hiz * Time.deltaTime, hiz * Time.deltaTime, hiz * Time.deltaTime);
                    // transform.Translate(Vector3.left * Time.deltaTime * hiz);
                }
                if (sp.ReadByte() == 2)
                {
                    transform.Translate(Vector3.right * Time.deltaTime * hiz,Space.World);
                }
            }
            catch (System.Exception)
            {
            }
        }
    }
}
```

UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Kod küp'e atanır ve sahne çalıştırılır. Sağ Sol butonu ile nesne hareket ettirilir

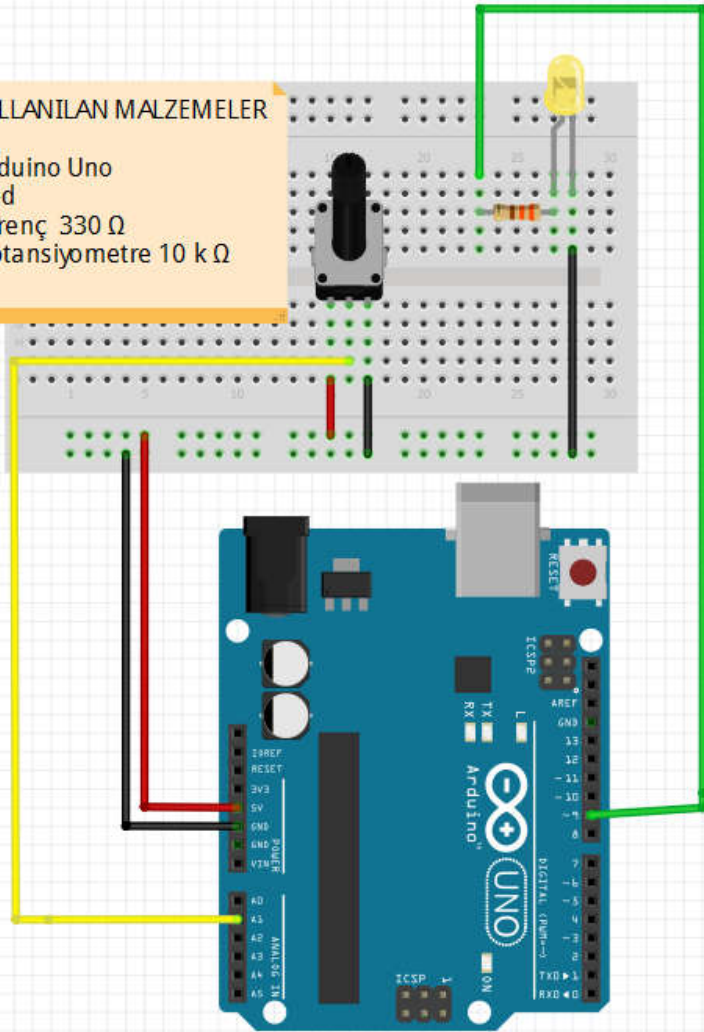


UNITY GÜNLÜĞÜ ARDUINO KONTROL-POT KONTROL

UNITY GÜNLÜĞÜ ARDUINO KONTROL-POT KONTROL

KULLANILAN MALZEMELER

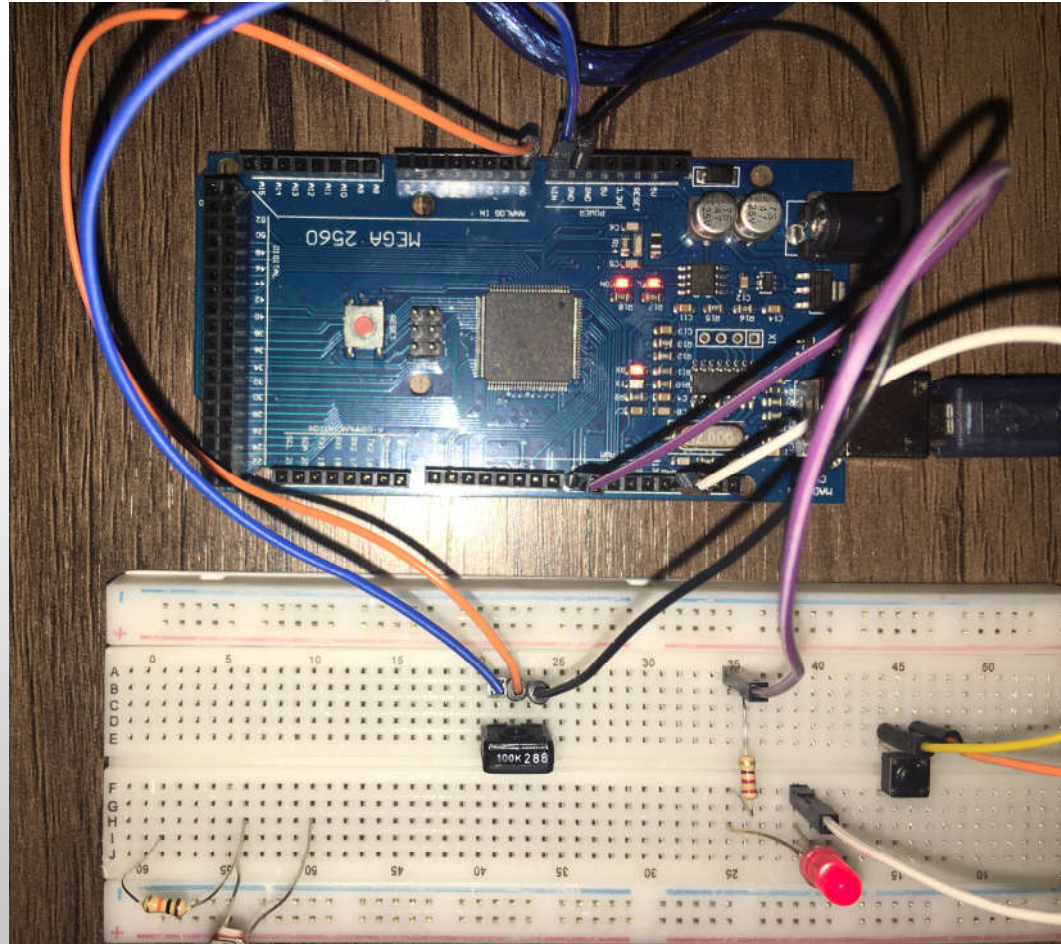
- Arduino Uno
- Led
- Direnç 330 Ω
- Potansiyometre 10 k Ω



Potansiyometrede ki okunan değer **0-10k Ω** arasında olduğundan orta pinini Analog girişlerimizden A1 'le birleştiriyoruz. Potansiyometrenin diğer uçlarından birine **+ 5v'a** diğerini **gnd'**ye (toprak) bağlıyoruz. Led'in yanma riskini azaltmak için önüne 330 Ω 'luk bir direnç bağlayarak Digital uçlardan 9 numaralı uca bağlantı yaptık. Ledin diğer ucunu GND'ye bağladık. 9 numaralı uç PWM girişi olduğundan Ledin giriş ucu olarak tercih ettik.

Arduino digital uçları (1,2,3...) HIGH (+5v) veya LOW (0v) değerini alır PWM uçlarında bu durum biraz farklı işler PWM uçları 1 yada 0 mantığına ilave olarak 0-255 arasında değer alarak o pindeki voltaj değerini 0-5 v arasında artırır veya azaltır.Bu durum lambanın parlaklığını etkiler.

UNITY GÜNLÜĞÜ ARDUINO KONTROL-POT KONTROL



Arduino kurulu devre şeması

UNITY GÜNLÜĞÜ ARDUINO KONTROL-POT KONTROL

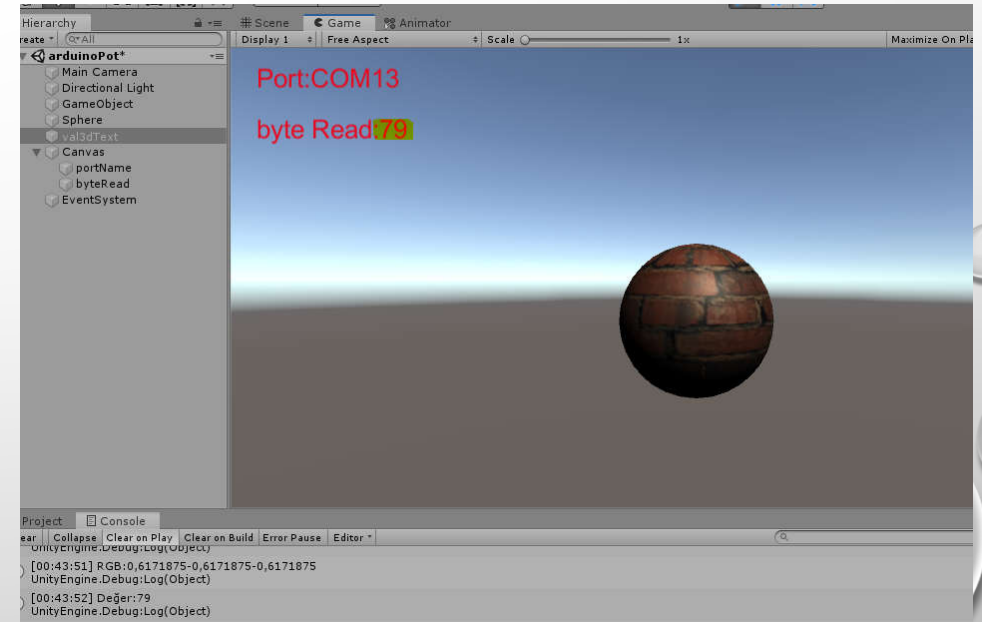
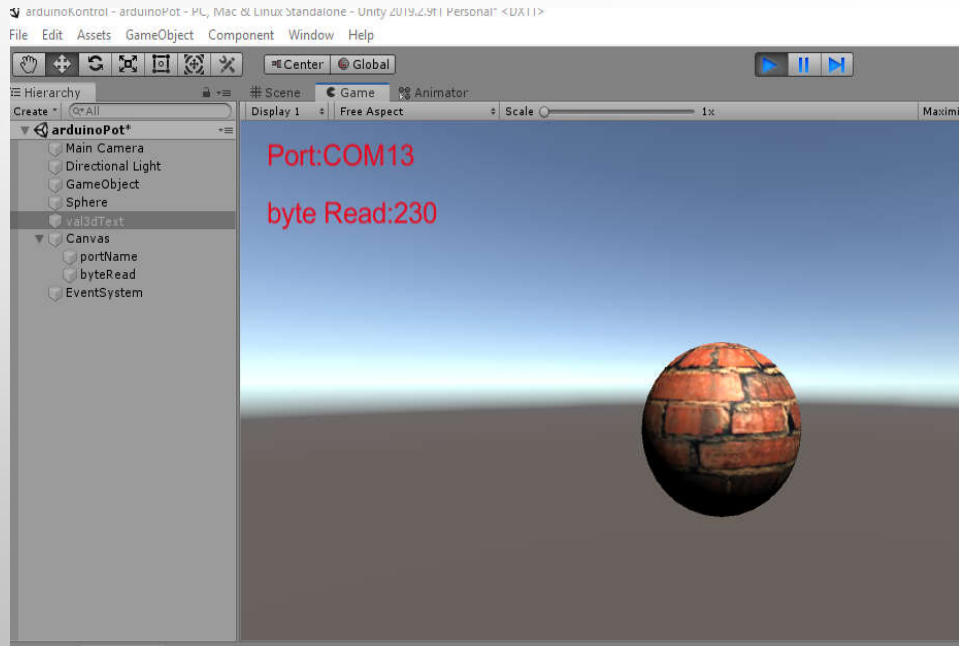
//Arduino Kodu

//Pot değerine göre Led ışığı ayarlanıyor

```
int LEDRed=8; // d8
int sensorPin=0; //a0
int val;
void setup(){
    Serial.begin(9600);
    pinMode(LEDRed,OUTPUT);
    Serial.flush();
}
void loop(){
    val=map(analogRead(sensorPin),0,1023,0,255);
    Serial.println(val);
    delay(200);
    analogWrite(LEDRed,val);
}
```

UNITY GÜNLÜĞÜ ARDUINO KONTROL-POT KONTROL

Unity Sahnesi. Farklı değerler için RGB değerine göre küre koyu ve açık materyal durumu.



UNITY GÜNLÜĞÜ ARDUINO KONTROL-POT KONTROL

Küreye atanacak **potKontrol.cs** kodu. Renderer ile Color kontrolü yapılmakta.

```
public class potKontrol : MonoBehaviour
{
    public Text portTxt, byteReadTxt;
    public TextMesh val3dTxt;
    public int Val;
    string strVal;
    public Color altColor = Color.green;
    public Renderer rend;

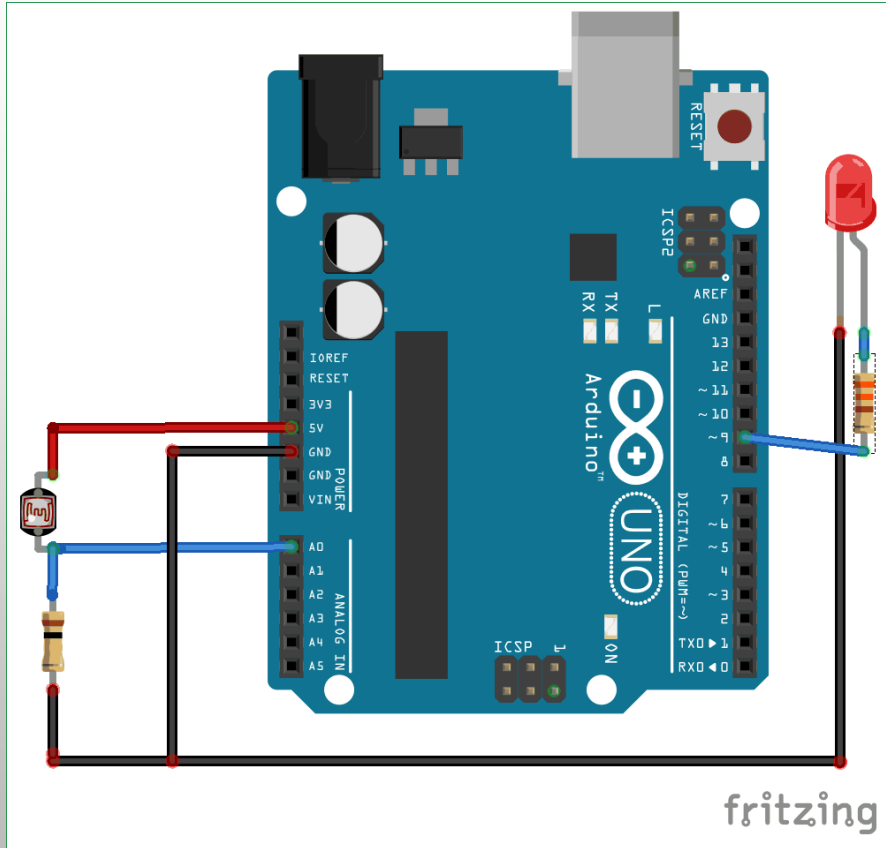
    SerialPort sp = new SerialPort("COM13", 9600);

    void Start()
    {
        sp.Open();
        sp.ReadTimeout = 500;
        rend = GetComponent<Renderer>();
        rend.material.color = altColor;
    }
}
```

```
void Update()
{
    if(sp.IsOpen)
    {
        //readLine()
        strVal = sp.ReadLine();
        Debug.Log("Değer:" + strVal);
        portTxt.text = "Port:" + sp.PortName;
        byteReadTxt.text = "byte Read:" + strVal;
        val = int.Parse(strVal);
        altColor.r = val/128f;
        altColor.g = val / 128f;
        altColor.b = val / 128f;
        Debug.Log("RGB:" + altColor.r + "-" + altColor.g + "-" + altColor.b);
        rend.material.color = altColor;
    }
}
```

UNITY GÜNLÜĞÜ ARDUINO KONTROL-LDR KONTROL

UNITY GÜNLÜĞÜ ARDUINO KONTROL-LDR KONTROL



Devre şeması

Bu devrede, odanın veya dış alanın ışık durumuna göre **bir LED'i kontrol etmek için Arduino ile LDR kullanarak bir Işık Sensörü** yapıyoruz.

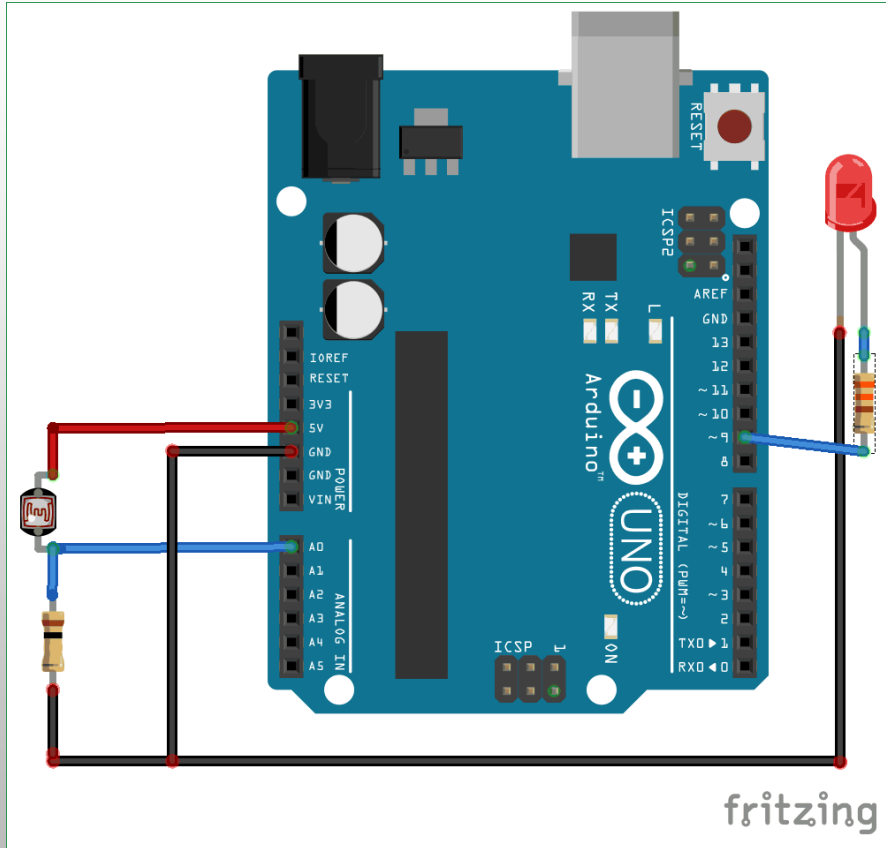
Gerekli Malzeme

- Arduino UNO
- LDR (Işığa Bağımlı Direnç)
- Direnç (100k-1; 330ohm-1)
- LED - 1
- Röle modülü - 5v
- Ampul / CFL
- Kabloları bağlama
- Breadboard

LDR

LDR, **Işığa Bağımlı Dirençtir** . LDR'ler, ışığa duyarlı özelliklerine sahip olmalarını sağlamak için yarı iletken malzemelerden yapılır. Işık LDR'nin yüzeyine düştüğünde (bu durumda) elemanın iletkenliği artar veya başka bir deyişle, ışık LDR'nin yüzeyine düştüğünde LDR'nin direnci düşer. LDR direncindeki bu azalmanın özelliği, yüzeyde kullanılan yarı iletken malzemenin bir özelliği olması nedeniyle elde edilir.

UNITY GÜNLÜĞÜ ARDUINO KONTROL-LDR KONTROL



Devre şeması

Arduino kullanarak LDR kontrollü LED'in çalışması

Devre şemasına göre, LDR ve 100k direnç kullanarak bir [voltaj bölücü devresi](#) yaptık. Voltaj bölücü çıkışı Arduino'nun analog pimine beslenir. Analog Pin voltajı algılar ve Arduino'ya bir miktar analog değer verir. Analog değer LDR direncine göre değişir. Böylece, ışık LDR üzerine düştüğünde, direnci azalır ve böylece voltaj değeri artar.

Işık yoğunluğu ↓ - Direnç ↑ - Analog pindeki voltaj ↓ - Işık yanar

Arduino koduna göre, analog değer 700'ün altına düşerse, bunu karanlık olarak kabul ederiz ve ışık yanar. Değer 700'ün üzerine çıkarsa, bunu parlak olarak kabul ederiz ve ışık söner.

UNITY GÜNLÜĞÜ ARDUINO KONTROL-LDR KONTROL

//Arduino Kodu

```
const int ledPin=13; //13 nolu dijital pin Led'i beslicecek
const int ldrPin=A0; //LDR direnç ortak ucunun bağlandığı analog giriş
void setup() {
    Serial.begin(9600);
    pinMode(ledPin,OUTPUT);
    pinMode(ldrPin,INPUT);
}

void loop() {
    int ldrStatus=analogRead(ldrPin);
    Serial.println(ldrStatus);
    delay(100);
    if(ldrStatus<=300)
    {
        digitalWrite(ledPin,HIGH);
        Serial.println("Ldr kapandı,LED açık");
    }
    else
    {
        digitalWrite(ledPin,LOW);
        Serial.println("-----");
    }
}
```

UNITY GÜNLÜĞÜ ARDUINO KONTROL- BUTTON KONTROL

Ayarlar yapıldıktan sonra, yeni bir C# kodu eklenir.

```
using UnityEngine;
using System.IO.Ports;

public class arduinoKontrol : MonoBehaviour
{
    SerialPort sp = new SerialPort("COM12", 9600);
    public int hiz = 5;

    void Start()
    {
        sp.Open();
        sp.ReadTimeout = 1;
    }

    void Update()
    {
        if(sp.IsOpen)
        {
            try
            {
                if(sp.ReadByte()==1)
                {
                    transform.Rotate(hiz * Time.deltaTime, hiz * Time.deltaTime, hiz * Time.deltaTime);
                    // transform.Translate(Vector3.left * Time.deltaTime * hiz);
                }
                if (sp.ReadByte() == 2)
                {
                    transform.Translate(Vector3.right * Time.deltaTime * hiz,Space.World);
                }
            }
            catch (System.Exception)
            {
            }
        }
    }
}
```