

ARDUİNO DERS NOTLARI

Arduino kolay bir şekilde çevresiyle etkileşime girebilen sistemler tasarlayabileceğiniz, hem acemi hem de ileri düzeydeki elektronik meraklılarına hitap eden, kolayca programlanabilen ve üzerine elektronik devre kurulabilen açık kaynaklı bir geliştirme platformudur.

Arduino uno açık kaynak kodlu geliştirme kartıdır. Arduino Uno en yaygın kullanılan arduino kartıdır.

14 dijital giriş / çıkış'a (6 PWM çıkışı), 6 analog girişe, bir 16 MHz kristal osilatöre, bir USB bağlantısına, bir güç girişine, bir ICSP başlığına ve bir reset düğmesine sahiptir.

Led devreleri, Robot projeleri, Çevre etkileşimli projeler, Otomasyon sistemleri gibi uygulamaları gerçekleştirebilirsiniz.

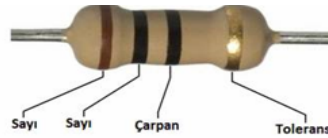
Aslında bu soruyu saydıklarımızla sınırlandırmamak daha uygun olur. Çünkü Arduino ile hayal gücünüzü kullanarak bir çok uygulama gerçekleştirebilirsiniz. Otomatik çiçek sulama, otomatik balık yeme, kamera zamanlama, programlanabilir trafik ışıkları..

Devre elemanları

- ▶ Arduino Kartı
- ▶ Bread Board
- ▶ Atlama Kablosu (Jumper)
- ▶ Diyot
- ▶ Led (Light Emitting Diode)
- ▶ Direnç
- ▶ Basma Butonu
- ▶ Potansiyometre
- ▶ Foto Direnç
- ▶ Buzzer
- ▶ Sıcaklık sensörü
- ▶ Transistor
- ▶ Dc motor
- ▶ Servo Motor
- ▶ Röle
- ▶ Entegre
- ▶ ...

Mikrokontrolörü	ATmega328P
Çalışma Voltajı	5V
Önerilen Giriş Voltajı	7-12V
Giriş Voltajının Limitleri	6-20V
Dijital Giriş-Çıkış Pinleri	14 (6'sı PWM özellikli)
Analog Giriş Pinleri	6
GND Pin sayısı	3

Direnç : Akımı sınırlamaya yarayan devre elemanıdır. Devreye uygulanan gerilim ve akım bir uçtan diğer uca ulaşmaya kadar izlediği yolda bir takım zorluklarla karşılaşır. İşte bu kuvvetlere **DİRENÇ** denir. R ile gösterilir. Birimi OHM'dur.



Asağıdaki tabloda dirençlerin üzerlerindeki renklere karşılık gelen sayısal değerler verilmiştir.

RENK	SAYI	ÇARPAN	TOLERANS
Siyah	0	$10^0 = 1$	-
Kahverengi	1	$10^1 = 10$	%1
Kırmızı	2	$10^2 = 100$	%2
Turuncu	3	$10^3 = 1000$	-
Sarı	4	$10^4 = 10.000$	-
Yeşil	5	$10^5 = 100.000$	%0,5
Mavi	6	$10^6 = 1.000.000$	%0,25
Mor	7	$10^7 = 10.000.000$	%0,1
Gri	8	$10^8 = 100.000.000$	-
Beyaz	9	$10^9 = 1.000.000.000$	-
Altın	-	0,1	%5
Gümüş	-	0,01	%10

☐ Dirençlerin değerleri , direnç renk kodları ile belirlenir.

☑ **Direnç renk kodu** kullanılan bu sistemde, dirençler dört, beş veya altı renkli olarak imal edilirler.

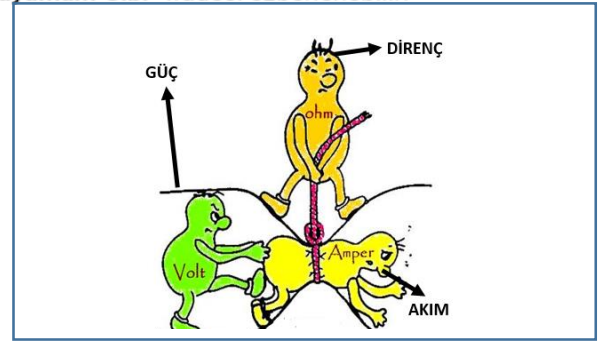
Dirençler seri bağlanır ise değerler toplanır.

İki adet 10 Ω direnç seri bağlanırsa 20 Ω direnç elde edilir.

İki adet 10 Ω direnç paralel bağlanırsa 5 Ω direnç elde edilir.

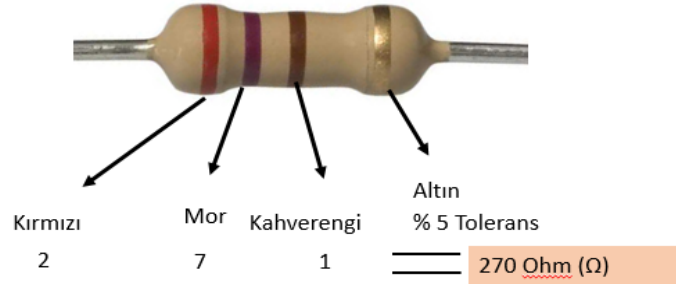
Renklerin sayı değerlerini kolayca akılda tutmak için "Sokakta Sayamam Gibi" ifadesi ezberlenebilir.

0	1	23	4	5	6	7	8	9
S	K	K	S	S	M	M	G	B
i	a	I	a	Y	a	M	i	B
y	h	r	r	ş	ş	r	r	y
a	v	r	r	ş	ş	r	r	y
h	r	m	r	ş	ş	r	r	y
	e	n	r	ş	ş	r	r	y
	r	g	r	ş	ş	r	r	y
	e	n	r	ş	ş	r	r	y
	r	g	r	ş	ş	r	r	y
	e	n	r	ş	ş	r	r	y
	r	g	r	ş	ş	r	r	y



Dirençin en küçük birimi ohm(Ω) dur.

1000 Ω = 1KiloOhm (k Ω)
1000 k Ω =1megaOhm (m Ω)



* Birinci ve ikinci renk aynen yazılır. Sonuna üçüncü renk kadar sıfır(0) konular. Elde edilen sonuç Ohm cinsinden değer olur. Son renk ise tolerans değerini gösterir. Örneğin yukarıdaki direnç %5 alt ve üst değerler için kullanılabilir.

Arduino Uno:

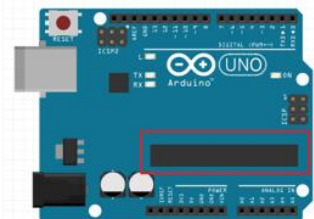
Haberleşme Pinleri : RX , TX	:	Serial out TX -0	Serial in RX - 1
Pwm : Dijital pinlerden analog çıkış alınabilmesi, 3,5,6,9,10,11 pinler			
0 ile 5 arasında değer alabilir.			
Analog Giriş-Çıkış: A0, A1 , A2 , A3 , A4 , A5			
Dijital Giriş-Çıkış Pinleri: 14 Dijital Pin ,akım var yada yok, 5v gerilim için high komutu verilir anot ucu bağlanır, Topraklama için low komutu verilir katot ucu bağlanır.			
Giriş Çıkış olma durumu kodlama ile belirlenir.			
Direnç: akım yavaşlatır. Elektrik akımını, akışını , gerilimi azaltan devre elemanıdır.			
Jumper : atlama kablosu			
Led: ışık yayan diyot.			
Ledlerin anot ve katot olmak üzere iki bacağı vardır.			
Pozitif gerilime Anot + ucu,			
Negatif (GND-ground-topraklama) katot – ucuna bağlanır.			
Diyot: Elektriğin tek yönde akmasına izin verir.			
Kutupları vardır. (Ters kutup –Güneş enerjisi ile)			
5V Akım Arduino 5V da çalışır. Arduino'yu bilgisayara bağladığınızda aldığınız güç kurduğunuz devreleri çalıştıracaktır.			

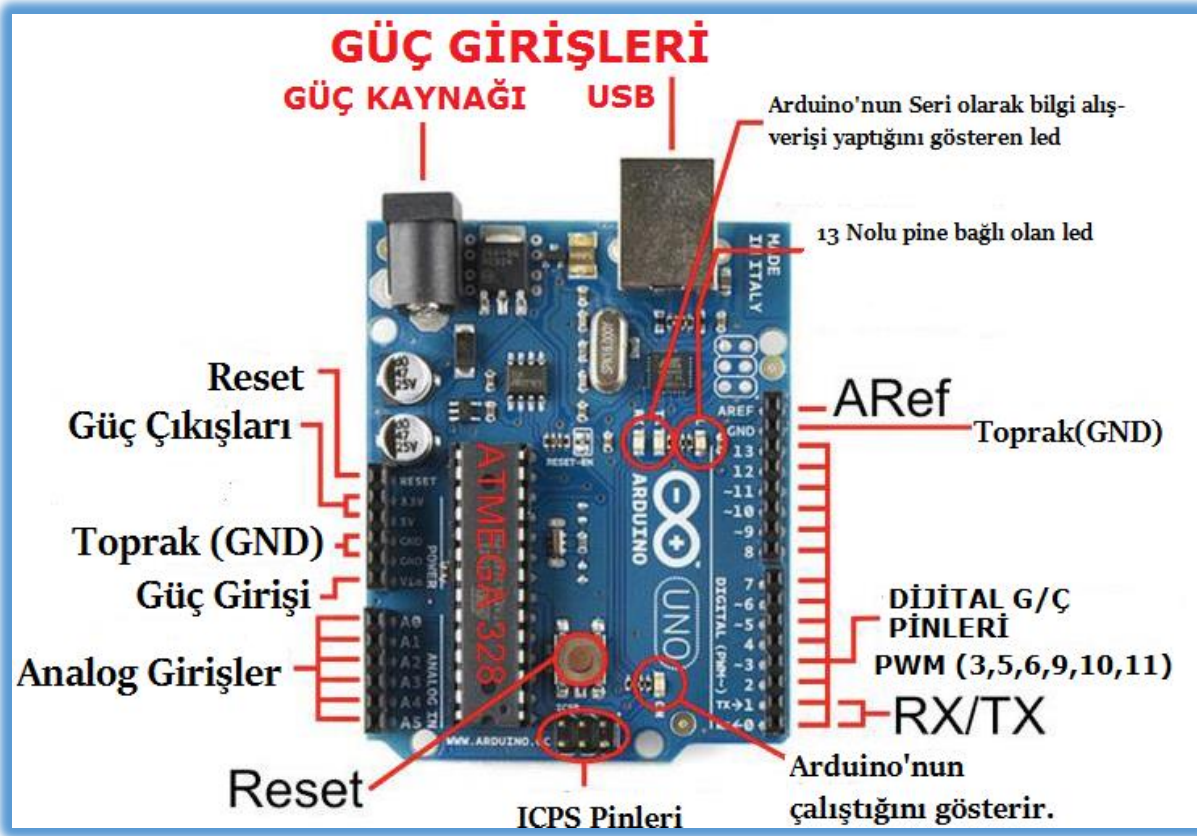
$V=I.R$
 $1000\text{mA}(\text{miliamper})=1\text{A}$
 $5\text{V}=0,020 * R$ ise $\rightarrow R=250$
(220 ohm direnç kullanmak yeterli)
Arduino kartı 5V gerilim ile çalışır.
(Arduino kart pinlerinde 40 mA akım vardır)
LEDler ise 20 mA ile çalışır.
Led veya kartın zarar görmemesi için direnç kullanılır.
Direnç lede seri bağlanır.



ATMEGA 328

İşlemci ATMEL firmasının Atmega işlemcisini görüyoruz. Yazdığımız programlar bu işlemciye gönderilir.





Bread Board

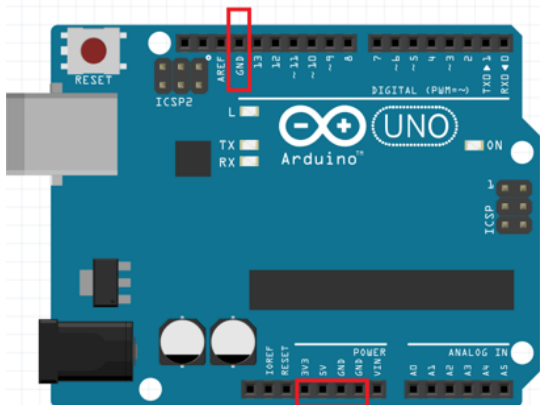
- Devre tahtası ile projelerimizi lehim yapmadan kolayca kurabiliriz.
- Genel olarak içerisinde birbirine bağlı hatları barındıran devre tahtası üzerine elektronik bileşenleri yerleştirerek projelerimizi çalışır hale getirebiliriz.
- Devre tahtası üzerinde bir birine bağlantılı paralel hatlar bulundurulur.
- Sol ve sağ yanlarda dikey olarak uzanan kırmızı ve mavi hatlar genellikle gerilim bağlantıları için kullanılır.
- Kırmızı hatta +, mavi hatta ise toprak hattını bağlayıp daha sonra devrenizin diğer bölümlerinde bu hatlar üzerinden gerilimlere (güç) ulaşabilirsiniz.
- Devre tahtalarının değişik boyuttaki türleri olsa da temel özelliği aynıdır. İhtiyacınıza ve kurmak istediğiniz devrelerin boyutlarına göre değişik tipte devre tahtalarını piyasada bulabilirsiniz.



Dijital ve Analog dünya

Dijital Dünya = ya vardır ya yoktur.

Analog Dünya= Az ya da çoktur. (Belirli bir değer aralığındadır)



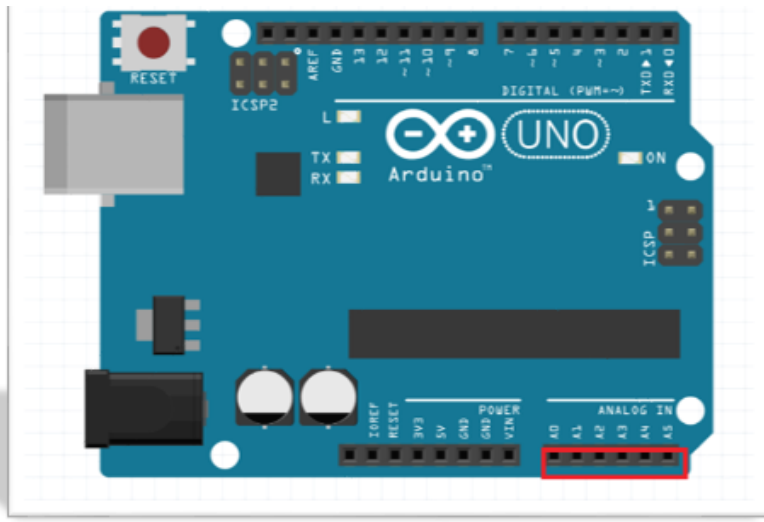
Pinler (5V, 3.3V, GND)

- Arduino üzerinde bulunan pinleri, jumper kablolar yardımıyla devre tahtasına (breadboarda) bağlarız.
- Her pinin farklı işlevleri vardır. Numaralandırılmış olan bu pinleri tek tek inceleyelim.

5V (4) ve 3.3V : Bu pinlerden 5 volt ve 3.3 volt değerinde **çıkış gerilimi** alabiliriz.

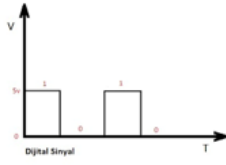
GND: Ground 'un kısaltılmış halidir. Topraklama bağlantısıdır.

ANALOG SİNYAL



Arduino UNO kartında A0, A1, A2, A3, A4, A5 olmak üzere 6 adet analog pin bulunuyor. Bu pinler 0 ile 1023 arasında toplam 1024 adet sayısal değer alıyor ve veriyor.

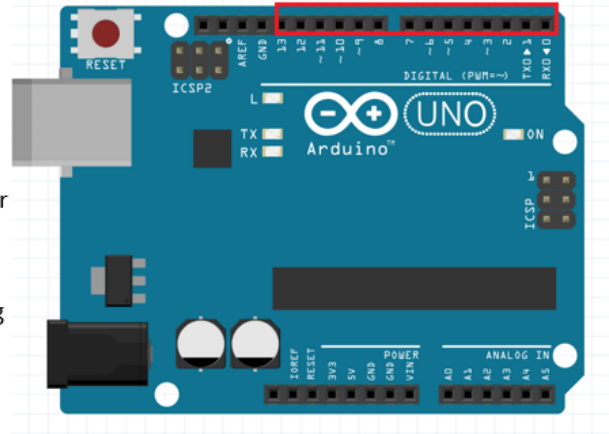
DİJİTAL SİNYALLER



0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 olmak üzere 14 adet dijital pin bulunuyor.

Bu pinler HIGH(yüksek) ve LOW(alçak) değerlerini alıyor ve veriyor.

Önünde “~” işareti gördüğünüz dijital pinlerden analog çıkış alabiliriz. Bu pinlere PWM pinleri denir.



Ayrıca bazı pinlerin özel fonksiyonları vardır:

Serial 0 (RX) ve 1 (TX) : Bu pinler TTL seri data almak (receive – RX) ve yaymak (transmit – TX) içindir.

Harici kesmeler (2 ve 3) : Bu pinler bir kesmeyi tetiklemek için kullanılabilir.

PWM: 3, 5, 6, 9, 10, ve 11 : Bu pinler analogWrite () fonksiyonu ile 8-bit PWM sinyali sağlar.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) : Bu pinler SPI kütüphanesi ile SPI haberleşmeyi sağlar.

LED 13 : Dijital pin 13 e bağlı bir leddir. Pinin değeri High olduğunda yanar, Low olduğunda söner.

Arduino kartı üzerinde bulunan pinleri kullanmak için pinleri giriş veya çıkış olarak ayarlamanız gerekmektedir.

Dijital giriş/çıkış pinlerini kullandık fakat Arduino kartımızda bir de “Analog Input” kısmı mevcut. Bu pinleri kullanarak dijitalden analoğa dönüşüm yaparak voltaj okumamız mümkün.

Potansiyometre : Aslında çevremizde her gün kullandığımız cihazların neredeyse hepsinde mevcut olan bir devre elemanıdır. Örneğin, müzik setimizin ses seviyesini değiştirmek için çevirdiğimiz düğme bir potansiyometredir. En basit açıklama ile potansiyometre, değerini elimizle çevirerek ayarladığımız bir dirençtir. Mikrokontrolcü uygulamalarında ise genellikle gerilim bölücü olarak kullanılır. (*Potansiyometre : Değişen direnç*)

Potansiyometre bir **direnç** türüdür. Fakat diğer direnç türlerinden ayrılan en büyük özelliği ise direnç değeri **değiştirilebilir** olmasıdır.

Analogdan Dijitale Çeviriciler (Analog-to-Digital Converter, ADC)

Arduino UNO kartımızdaki işlemcide, 10-bit çözünürlüğe sahip analogdan dijitale dönüştürücü (ADC – analog to digital converter) mevcuttur. Peki, bu 10-bit ne anlama geliyor? Bildiğimiz üzere Arduino'muzun mikrokontrolcüsü 5V gerilimle çalışmakta. Bu mikrokontrolcüde sahip olduğumu söylediğimiz 10-bit ADC, 0V ile 5V arası gerilimleri $2^{10} = 1024$ adım hassasiyet ile okuyabilir. Yani analog input pinlerinden birine vereceğimiz 0V gerilim bize 0 değerini; aynı şekilde 5V gerilim ise 1023 değerine denk düşüyor.

İki çıkış pini 5 Volta bağlandığında ortadaki pin potansiyometredeki kontrol düğmesinin konumuna göre 0V ile 5V arasında çıkış verir. Potansiyometreyi bir yöne çevirdiğimizde yan yana olan iki bacağına direnci değişir. Bunu bir multimetreye ölçerek görebiliriz.

Potansiyometre Bağlantıları



1. Pin	2. Pin	3. Pin
+5 volt	Analog Giriş	GND

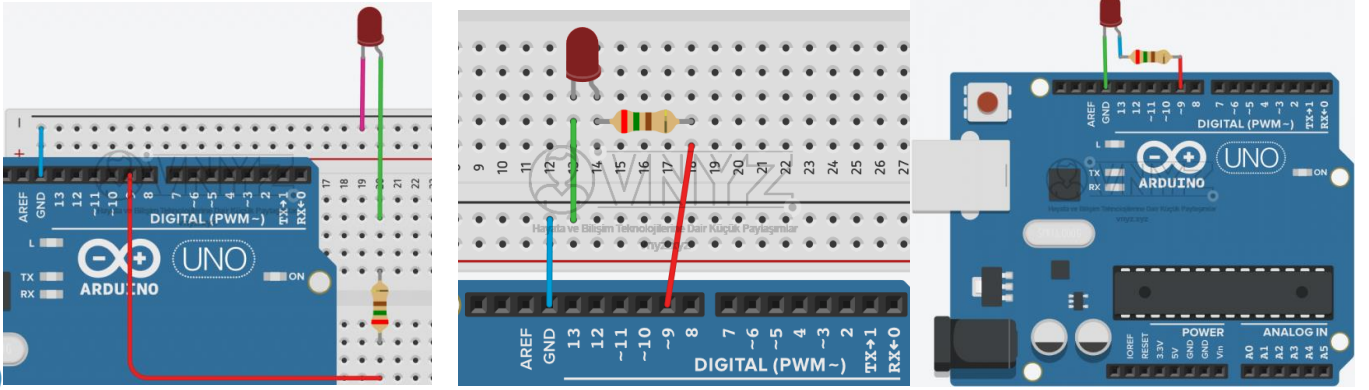
Butonlar:

- Butonlar mikroişlemcilerle yapılan her proje önemli elemanlardır. Yazılan programları çalıştırmada ya da çalışan programlara anında müdahale için butonlar kullanılmaktadır.
- Butonlar önemli çünkü mikroişlemciler ile yapacağımız herhangi bir robotik uygulamada, robota bu butonlar sayesinde yön verme gibi bir seçeneğimiz bulunmakta. Oyuncak arabalar için git/gel komutları gibi.

Arduino UNO'da sadece reset butonu olduğu için bu butonu kendimiz harici olarak ekleyeceğiz.

Rgb Led: Yanıp sönen bir LED'den daha eğlenceli şey nedir sizce? Tabi ki renkli LED'ler! RGB, yani red-bluegreen LED'ler 3 adet renk barındıran ve her tür renk için kombin edilebilen bir diyet türüdür. Bu devrede Bir RGB LED kullanarak nasıl renk kombinasyonları oluşturabileceğinizi öğreneceksiniz. Her bir diyetun parlaklığına bağlı olarak neredeyse bütün renkleri elde etmeniz mümkün.

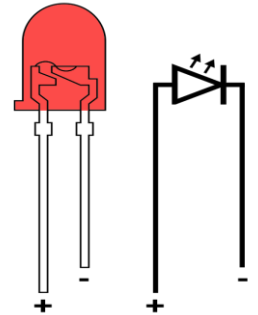
DEVRE TASARIMI (Arduino Led Yakıp Söndürme (Blink- Dijital Pinler)



LED'in kısa bacağı(-) kart üzerindeki GND (Toprak, -) pinine,

uzun bacağı(+) ise 250 ohm'luk direnç ile 9 nolu dijital giriş çıkış pinine bağlayın.

UYARI – LED lamba bir diyet olduğu için devrenize bağlarken + (artı-anot) ve – (eksi-katot) bacıklarını doğru şekilde bağladığınıza dikkat edin.



UYARI –Arduino 5V ile çalışır. LED ise yaklaşık 20mA akıma ihtiyaç duyar. LED'i direkt olarak Arduino'ya bağlarsak LED veya Arduino zarar görebilir. Bu yüzden yaklaşık olarak 250 ohm dirençle birlikte takmalıyız. ($V=ixR > 5=0,02xR > R=250 \text{ ohm}$)

2- ARDUINO IDE (Arduino Led Yakıp Söndürme (Blink- Dijital Pinler))

```
1. void setup() {
2.   pinMode(9,OUTPUT); //9 nolu pine çıkış görevi verdik.
3. }
4.
5. void loop() {
6.   digitalWrite(9,HIGH); //9 nolu pine enerji(sinyal) ver. (HIGH-YÜKSEK-1)
7.   delay(1000); //1000 milisaniye bekle. (1000 milisaniye=1 saniye)
8.   digitalWrite(9,LOW); //9 nolu pine enerji(sinyal) verme. (LOW-DÜŞÜK-0)
9.   delay(1000); //1000 milisaniye bekle. (1000 milisaniye=1 saniye)
10. }
```

ARDUİNO KOMUTLARI, AÇIKLAMALARI VE ÖRNEK KULLANIMLARI

-// – Tek Satırlık Açıklama yazmak için kullandığımız işaretir. // işaretini yadığımız kısımdan sonra satırdaki yazıları program atlar. Birden fazla satırda açıklama eklemek için /* */ işaretini kullanırız.

UYARI – Kodlarınızı { } (süslü parantez) arasına yazmalısınız.

pinMode() \\pinMode komutu belirtilen pini giriş veya çıkış olarak ayarlar. **pinMode(pinNo,mod);** Şeklinde yazılır

-**pinMode()** – Kartın üzerindeki bir pinin hangi görevde çalışacağını belirler. **pinMode(pin,görev);**

pinMode(pinNo,INPUT); pinMode(pinNo,OUTPUT);

İlk parametre elemanın takılı olduğu pin nosu, ikinci parametre ise o pine hangi görevi vereceğimizi belirleriz.

pinMode(10,INPUT); \\ 10 Nolu pin Giriş olarak tanımlandı.

pinMode(13,OUTPUT); \\ 13 Nolu pin Çıkış olarak tanımlandı.

Dijital bir pin çıkış olarak ayarlandığında, ayarlanan pine LOW veya HIGH yazmak için kullanılır. LOW değeri pini 0V (GND) seviyesine çeker, HIGH değeri ise arduino nun çıkış voltajı olan +5V a yükseltir. Ancak bu Arduino DUE gibi kartlar 3.3V ile çalıştığı için çıkışta 3.3V alınır.

digitalWrite(pinNo,HIGH); veya **digitalWrite(pinNo,LOW);** şeklindedir.

1. Parametre hangi pinin kullanılacağı,

2. parametre ise pinin durumu.

digitalWrite(7,HIGH); \\ 7 nolu pinden 5V çıkış verildi (Lojik-1)

digitalWrite(9,LOW); \\ 9 nolu pinden 0V çıkış verildi (Lojik-0)

-**digitalWrite()** – Herhangi bir dijital ayağın çıkış vermesi için kullanırız. **digitalWrite(pin,değer);** İlk parametre hangi pine çıkış vericeksek o pinin nosu, ikinci parametre ise o pinin alacağı değeri yazarız. Bu kısımda iki değer

yazabiliriz; **HIGH** (YÜKSEK-1) veya **LOW** (DÜŞÜK-0).

İPUCU– HIGH değeri yazıldığında o pinden 5V 25mA civarında elektrik çıkışı olur. LOW değeri yazıldığında ise elektrik çıkışı kesilir.

digitalRead()

Dijital giriş olarak tanımlanan bir pin üzerindeki değeri okur ve bir sonuç döndürür. Bu sonuç HIGH veya LOW dur. Biz bu sonuca göre işlemler yapabiliriz. En basit örnek olarak buton verilebilir. Butona basıldığında digitalRead() fonksiyonu bize HIGH değeri döndürür ve led yanar. Kullanımı : digitalRead(pin) şeklindedir. Buradaki “pin” bizim giriş olarak ayarladığımız pin numarasıdır. Veya pin numarasını atadığımız değişken olabilir.

Arduino ile PWM sinyalin üretilmesi – analogWrite(pin, değer) : Analog olarak çıkış sinyalinin

üretilmesi analogWrite(pin, değer) komutu ile yapılmaktadır. Pin yazılan kısma çıkış alınmak istenen pinin numarası yazılırken, değer olarak belirtilen kısma 0-255 aralığında bir değer girilmelidir.

Örneğin analogWrite(6, 255); yazıldığında Arduino’nun 6. Pininden 5 Voltluk gerilim alınabilir.

analogWrite(6, 127); yazıldığında ise 6. Pinden yaklaşık olarak 2.5 Voltluk bir gerilim alınır.

Örnek Kod

```
int ledPin = 10;          // LED 10 numaralı pine bağlandı
void setup()
{ pinMode(ledPin, OUTPUT); // ledPin ( pin 10 ) çıkış olarak ayarlandı }
void loop()
{ digitalWrite(ledPin, HIGH); // Led Yanıyor
  delay(1000);              // 1 saniye bekliyoruz
  digitalWrite(ledPin, LOW); // Led Sönüyor
  delay(1000);             // 1 saniye bekliyoruz }
```

Not: digitalWrite(ledPin,HIGH) yerine digitalWrite(10,HIGH) yazabilirdik

Yukarıdaki örnek ile arduino nun 10. pini çıkış olarak ayarlandı ve 1 saniye aralıkla 10. numaralı pin durumu değiştirildi. 10 numaralı pine bir led bağlayarak ledin 1 saniye aralıkla yanıp söndüğünü görebiliriz.

Örnek Kod

```
int ledPin = 13; // LED 13 numaralı pine bağlandı
int inPin = 7; // Buton 7 numaralı pine bağlandı
int val = 0; // Okunan değer saklanacağı değişkenimiz
void setup()
{ pinMode(ledPin, OUTPUT); // Dijital 13 Numaralı pin çıkış olarak ayarlandı
  pinMode(inPin, INPUT); // Dijital 7 Numaralı pin giriş olarak ayarlandı }
void loop()
{ val = digitalRead(inPin); // Giriş pinimizin değeri okunuyor.
  digitalWrite(ledPin, val); // Giriş pininden okunan değer ledPinine yazılıyor. }
```

-void setup() {...} – Arduino çalıştığı anda yüklenmesi gereken ayar ve kurulum kodları buraya yazılır.

```
void setup() {
```

```
//led pinleri, motorlar, sensorler, vs..
```

```
}
```

-void loop() {...} – Projemizin çalışması için gerekli olan kodların yazıldığı alandır.

-delay() – Parantez içine yazılan sayı değeri kadar milisaniye program bekler. (1000 milisaniye = 1 saniye)

```
void setup() {
  pinMode(9, OUTPUT);
}

void loop()
{
  digitalWrite(9, HIGH);
  delay(1000);
  digitalWrite(9, LOW);
  delay(1000);
}
```

port ismi

Giriş / Çıkış Yönlendirmesi

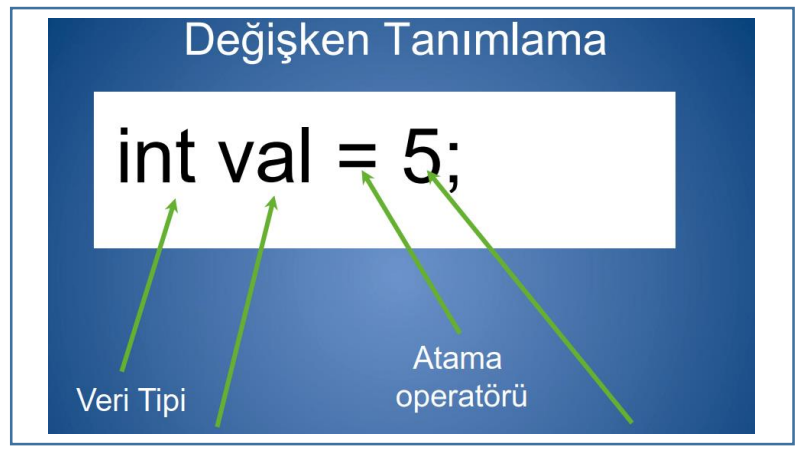
LED yak, söndür

1 sn bekle

pinMode komutu ile bir pini Giriş olarak yönlendirmek için INPUT, çıkış olarak yönlendirmek için ise OUTPUT deyimini kullanılır.

Arduino Etkinlikler :

1. Buton ile led yakma (buton iki kere bas),
2. Potansiyometre ile led parlaklığı,
3. Potansiyometre ile rgb led renk deęiřtirme,
4. Trafik ışığı,
5. Çoklu led yakma,

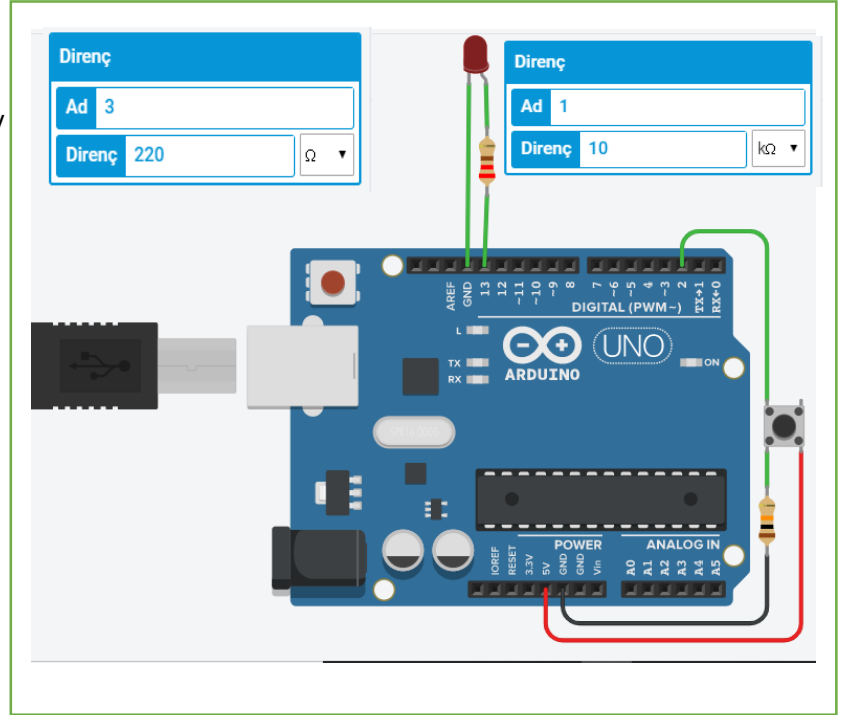


BUTON İLE LED YAKMA : (Tek buton ile led yakıp söndürüyoruz iki kere basınca yanacak,bir kere basınca sönecek)

/*Tek buton ile led yakıp söndürüyoruz

iki kere basınca yanacak bir kere basınca sönecek */

```
int buton = 0;
int ledyak = 0;
void setup() {
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  pinMode(2,INPUT);
  digitalWrite(13,LOW);}
void loop() {
  if (digitalRead(2) == HIGH) {
    buton=buton+1;
    if(digitalRead(2)==HIGH){
      while(1)
      {    if(digitalRead(2)==LOW)
        break;    }    }
    if(ledyak==1){
      buton=0;
      digitalWrite(13,LOW);
      ledyak=0; } }
  if (buton>=2){
    digitalWrite(13,HIGH);
    ledyak=1;
  }
  Serial.println(buton);
  delay(100); }
```

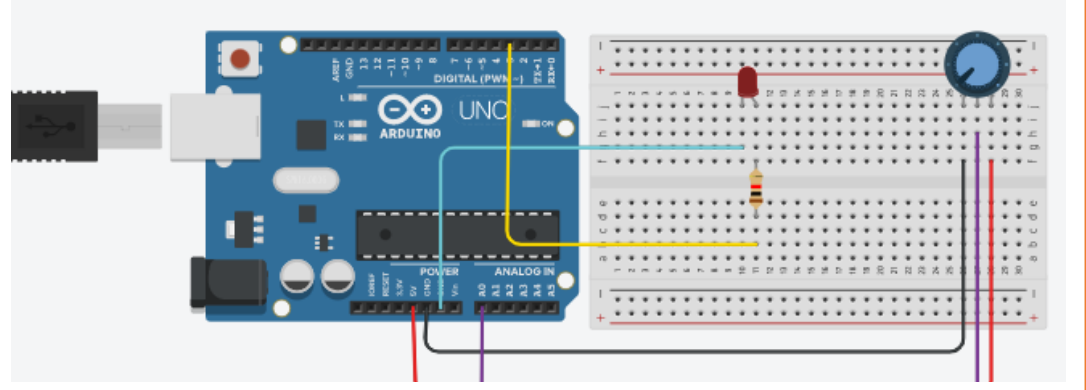


Potansiyometre İle Led Parlaklığını Kontrol Etme

Malzeme Listesi:

- 1) Arduino Uno
- 3) potansiyometre (250 Ω)
- 4) Bread Board
- 5) Yeter sayıda jumper kablo
- 6) LED Diyot
- 7) 1 k Ω direnç

Kurulacak Devre :



long deger;

void setup()

{

pinMode(3, OUTPUT);

pinMode(A0,INPUT);

}

void loop()

{

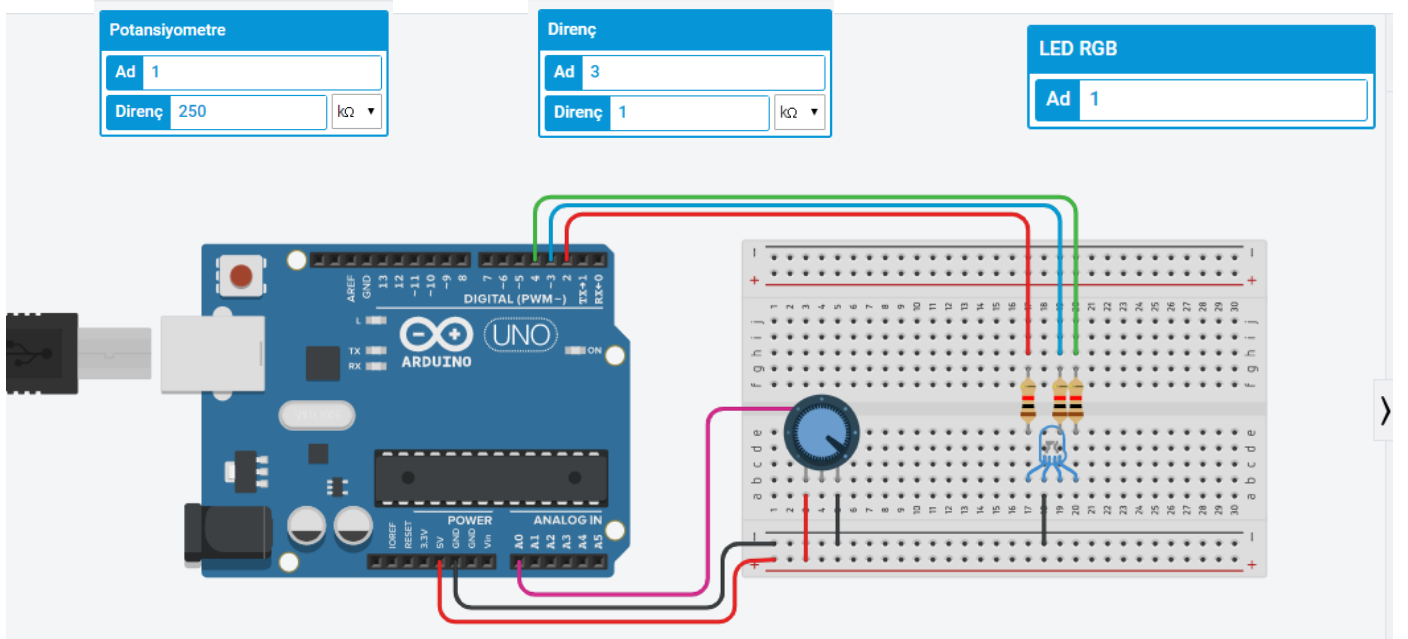
deger=analogRead(A0)/4;

analogWrite(3,deger);

}

Potansiyometre İle RGB Led Kontrolü

Kurulacak Devre:



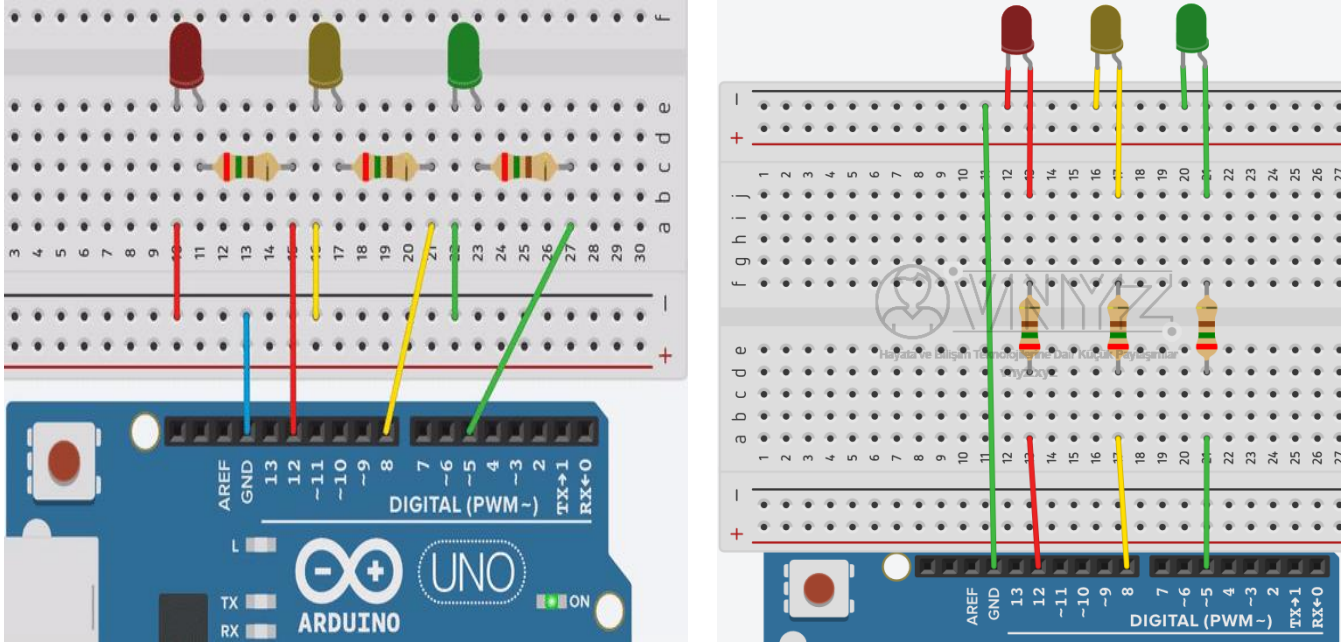
Potansiyometre İle RGB Led Kontrolü

```
int redPin = 2;
int bluePin = 3;
int greenPin = 4;
int potPin = A0;
int val;
void setup()
{
    pinMode(redPin, OUTPUT);
    pinMode(bluePin, OUTPUT);
    pinMode(greenPin, OUTPUT);
}
void loop()
{
    val = analogRead(potPin);
    if(val < 320)
    {
        analogWrite(redPin, 255);
        analogWrite(bluePin, 0);
        analogWrite(greenPin, 0);
    }
    else if(val < 640)
    {
        analogWrite(redPin, 0);
        analogWrite(bluePin, 255);
        analogWrite(greenPin, 0);
    }
    else
    {
        analogWrite(redPin, 0);
        analogWrite(bluePin, 0);
        analogWrite(greenPin, 255);
    }
}
```

Trafik ışığı :

Arduino UNO , Breadboard LED x 3 (Kırmızı, Sarı, Yeşil) , Direnç x 3 (250ohm) ,Jumper Kablolar

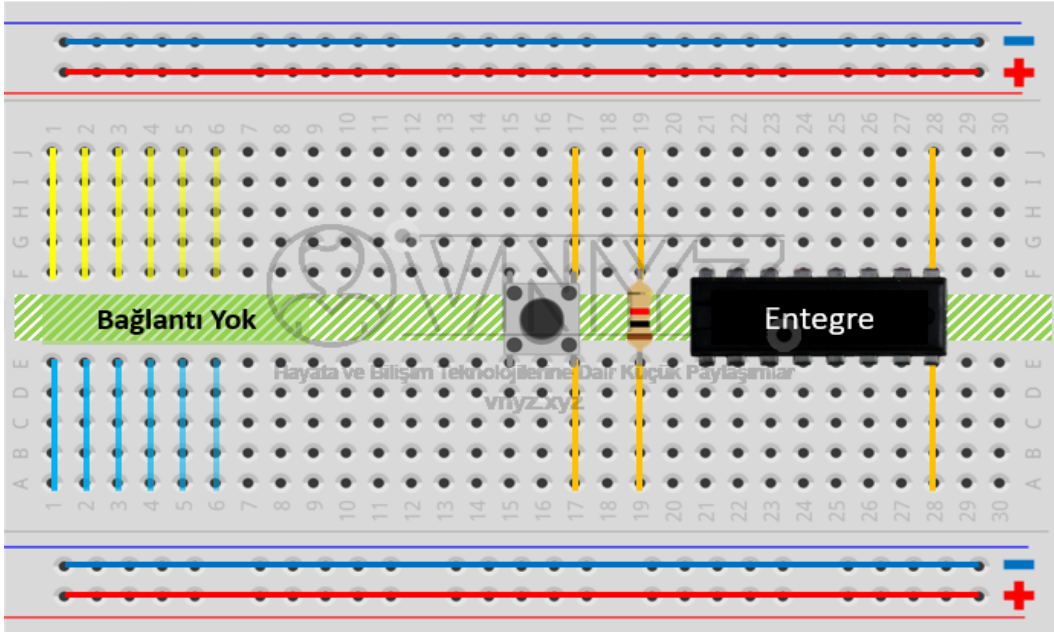
Kurulacak Devre :



HATIRLATMA – LED'in kısa bacağı GND'ye(-, eksi), uzun bacağı dirençle birlikte herhangi bir dijital giriş-çıkışa bağlanır.

Devrede gördüğümüz gibi ledin uzun bacağı direncin bir bacağı ile aynı beşli sütuna bağlanmış. Breadboard'un yapısı gereği bunlar uç uca bağlanmış gibi olur. Direncin diğer ayağında herhangi bir dijital giriş-çıkışa bağladık.

“LED Blink” devresinde LED'i direkt olarak GND'ye bağlamıştık. Ama burda birden fazla LED'imiz ve az sayıda GND'miz var. Bu yüzden GND'yi bir kablo yardımıyla Breadboard'un -(eksi) satırına bağladık. LED'lerin kısa bacaklarında -(eksi) satırına bağlayarak tüm LED'leri GND'ye bağlamış olduk.Devremizi tam olarak anladysak



kodlamaya geçebiliriz.

Breadboard'un Yapısı : + ve - satırları ayrı ayrı, ortadaki beşerli sütunlar ise ayrı ayrı birbirlerine bağlıdır. Bağlantı olmayan kısımda araya başka elemanlar bağlayarak alttaki ve üstteki beşerli sütunları birbirlerine bağlamış oluruz. Yani üstteki devremizi şu şekilde de bağlayabilirdik.

1- mBlock



Tinkercad için

Daha önce kullandığımız bloklardan farklı bir blok kullanmadık. Kodları inceleyecek olursak;

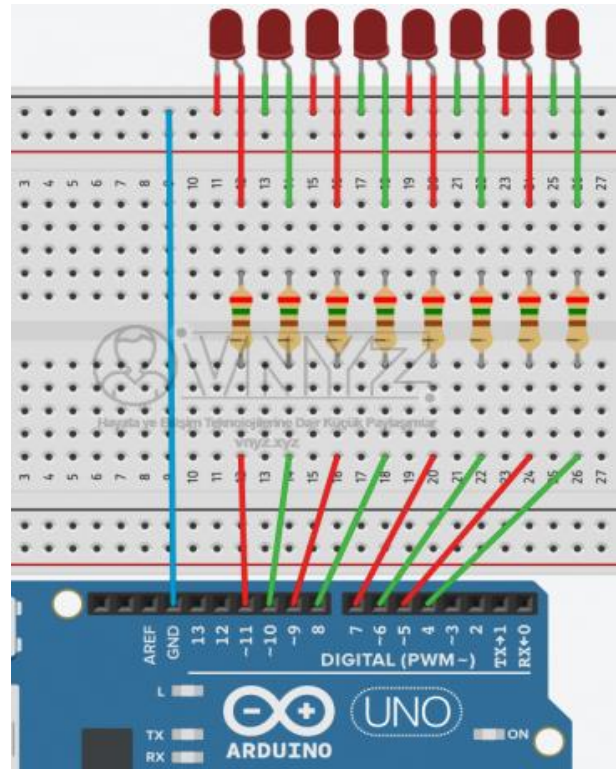
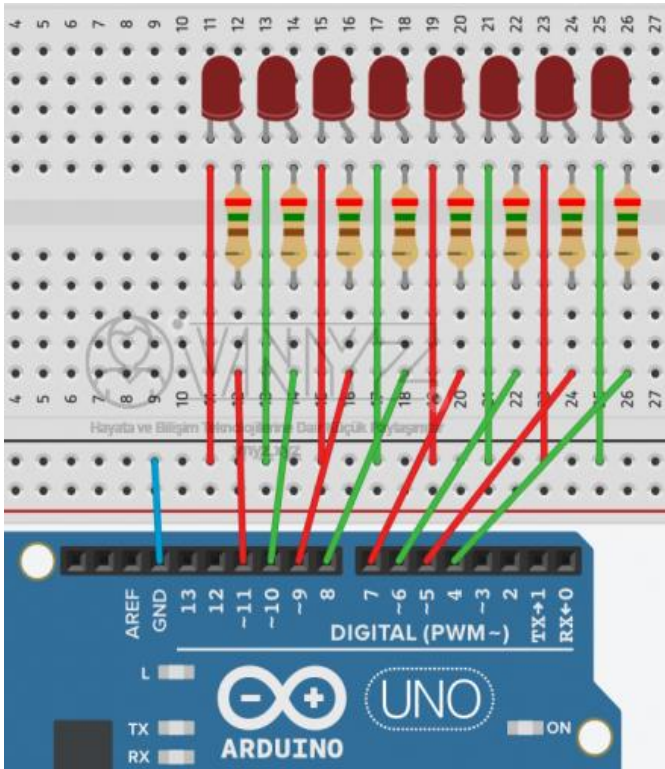
- 12 nolu pine bağlı olan kırmızı LED yanıyor.
- 7 saniye bekliyor
- 12 nolu pine bağlı olan kırmızı LED sönüyor.
- 8 nolu pine bağlı olan sarı LED yanıyor.
- 1 saniye bekliyor
- 8 nolu pine bağlı olan kırmızı LED sönüyor.
- 5 nolu pine bağlı olan sarı LED yanıyor.
- 5 saniye bekliyor
- 5 nolu pine bağlı olan kırmızı LED sönüyor.

2- ARDUINO

```
1. void setup()
2. {
3. pinMode(12, OUTPUT);
4. pinMode(8, OUTPUT);
5. pinMode(5, OUTPUT);
6. }
7.
8. void loop()
9. {
10. digitalWrite(12, HIGH);
11. delay(7000);
12. digitalWrite(12, LOW);
13. digitalWrite(8, HIGH);
14. delay(1000);
15. digitalWrite(8, LOW);
16. digitalWrite(5, HIGH);
17. delay(5000);
18. digitalWrite(5, LOW);
19. }
```

- LED Animasyon (Çoklu Led Yakma)

Kurulacak Devre:



Malzemeler: Arduino UNO, Breadboard LED x 8, Direnç x 8 (250ohm), Jumper Kablolar
HATIRLATMA – LED'in kısa bacağı GND'ye(-, eksi), uzun bacağı dirençle birlikte herhangi bir dijital giriş-çıkışa bağlanır.

1- mBlock



mBlock - LED Animation

- 11 ve 4 nolu pine bağlı LED lambalar yanıyor
- 0.1 saniye bekliyor
- 10 ve 5 nolu pine bağlı LED lambalar yanıyor
- 0.1 saniye bekliyor
- 9 ve 6 nolu pine bağlı LED lambalar yanıyor
- 0.1 saniye bekliyor
- 8 ve 7 nolu pine bağlı LED lambalar yanıyor
- 0.1 saniye bekliyor
- 8 ve 7 nolu pine bağlı LED lambalar sönüyor
- 0.1 saniye bekliyor
- 9 ve 6 nolu pine bağlı LED lambalar sönüyor
- 0.1 saniye bekliyor
- 10 ve 5 nolu pine bağlı LED lambalar sönüyor
- 0.1 saniye bekliyor
- 11 ve 4 nolu pine bağlı LED lambalar sönüyor
- 0.1 saniye bekliyor

2- ARDUINO IDE

```
1. void setup(){
2. pinMode(11,OUTPUT);
3. pinMode(10,OUTPUT);
4. pinMode(9,OUTPUT);
5. pinMode(8,OUTPUT);
6. pinMode(7,OUTPUT);
7. pinMode(6,OUTPUT);
8. pinMode(5,OUTPUT);
9. pinMode(4,OUTPUT);
10. }
11.
12. void loop(){
13. //LED lambalar içe doğru sırasıyla yanıyor
14. digitalWrite(11,HIGH);
15. digitalWrite(4,HIGH);
16. delay(100);
17. digitalWrite(10,HIGH);
18. digitalWrite(5,HIGH);
19. delay(100);
20. digitalWrite(9,HIGH);
21. digitalWrite(6,HIGH);
22. delay(100);
23. digitalWrite(8,HIGH);
24. digitalWrite(7,HIGH);
25. delay(100);
26.
27. //LED lambalar bu sefer tersten sırasıyla sönüyor.
28. digitalWrite(8,LOW);
29. digitalWrite(7,LOW);
30. delay(100);
31. digitalWrite(9,LOW);
32. digitalWrite(6,LOW);
33. delay(100);
34. digitalWrite(10,LOW);
35. digitalWrite(5,LOW);
36. delay(100);
37. digitalWrite(11,LOW);
38. digitalWrite(4,LOW);
39. delay(100);
40. }
```