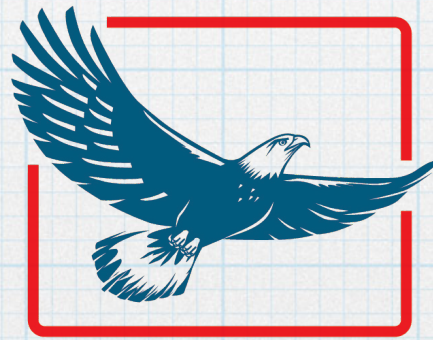




# DEĞİŞKENLER, OPERATÖRLER VE KARAR YAPILARI

ANTALYA SINAV ANADOLU LİSESİ ROBOTİK TOPLULUĞU



## SINAV KOLEJİ



E-posta: [yucelkilog@antalyasinavkoleji.com](mailto:yucelkilog@antalyasinavkoleji.com)



# Değişkenler

Değişkenler bilgisayarın geçici belleğinde bilginin saklandığı gözlemlenebilir verilen sembolik adlardır.



# Değişkenler

(Değişken Türü) (Değişken Adı) = (Değişkenin İlk Değeri);

```
int tamsayi = 345;
```



# Değişken Türleri

Tip	Boyut(byte)	Aralık	Açıklama
int	2	-32768 ile 32767 arasında	işaretli tamsayılar
unsigned int	2	0 ile 65535 arasında	pozitif tamsayılar
long	4	-2147483648 ile 2147483647 arasında	küçük negatif ve büyük pozitif tamsayılar
unsigned long	4	0 ile 4294967295 arasında	büyük pozitif tamsayılar
char	1	-128 ile +127 arasında	işaretli -128 ile +128 arasındaki sayılar veya tek bir karakter
string	boyutu karakter sayısına göre değişkenlik gösterir	karakterlerden oluşan dizi	tırnak işareti(" ") içerisine yazılan karakter dizisidir. Örneğin; "2016 ya Merhaba Dünya!"
byte	1	0 ile 255 arasında	0 ile 255 arasında tam sayılar
float	4	-3.4028235E+38 ile 3.4028235E+38 arasında	ondalık sayılar
double	4	-3.4028235E+38 ile 3.4028235E+38 arasında	ondalık sayılar (arduinoda float ile double aynı aralığa sahiptir)
boolean	1	true(1) ya da false(0)	mantık ifadelerinde kullanılırlar



# Operatörler (aritmetik)

$y = y + 3;$

// Toplama

$x = x - 7;$

// Çıkarma

$i = j * 6;$

// Çarpma

$r = r / 5;$

// Bölme



# Operatörler (birleştirme)

$x++$  //  $x = x + 1$  ile aynı

$x--$  //  $x = x - 1$  ile aynı

$x += y$  //  $x = x + y$  ile aynı

$x -= y$  //  $x = x - y$  ile aynı

$x *= y$  //  $x = x * y$  ile aynı

$x /= y$  //  $x = x / y$  ile aynı



# Operatörler (kıyaslama)

$x == y$	// x, y'ye eşittir
$x != y$	// x, y'ye eşittir değil
$x < y$	// x, y'den küçüktür
$x > y$	// x, y'den büyüktür
$x <= y$	// x, y'den küçüktür veya eşittir
$x >= y$	// x, y'den büyüktür veya eşittir



# Operatörler (mantıksal)

Mantıksal “VE”;

`if (x > 0 && x < 5)`

//x'in 0'dan büyük **VE** x'in 5'ten küçük olduğu durumlar.

Mantıksal “VEYA”;

`if (x > 0 || y > 0)`

//x'in 0'dan büyük **VEYA** y'nin 0'dan küçük olduğu durumlar.

Mantıksal “DEĞİL”;

`if (!x > 0)`

// x 0'dan büyük **DEĞİL**.



# Operatörler (mantıksal)

Mantıksal “DOĞRU/YANLIŞ”;

if (x == TRUE)

//x değişkeni DOĞRU ise

if (x == FALSE)

//x değişkeni YANLIŞ ise



# Koşul ifadeleri

```
if (birDegisken ?? deger)
{
    birseyYap;
}
```



# Koşul ifadeleri

```
if (girisPini == HIGH)
{
    aIsiniYap;
}
else
{
    bIsiniYap;
}
```



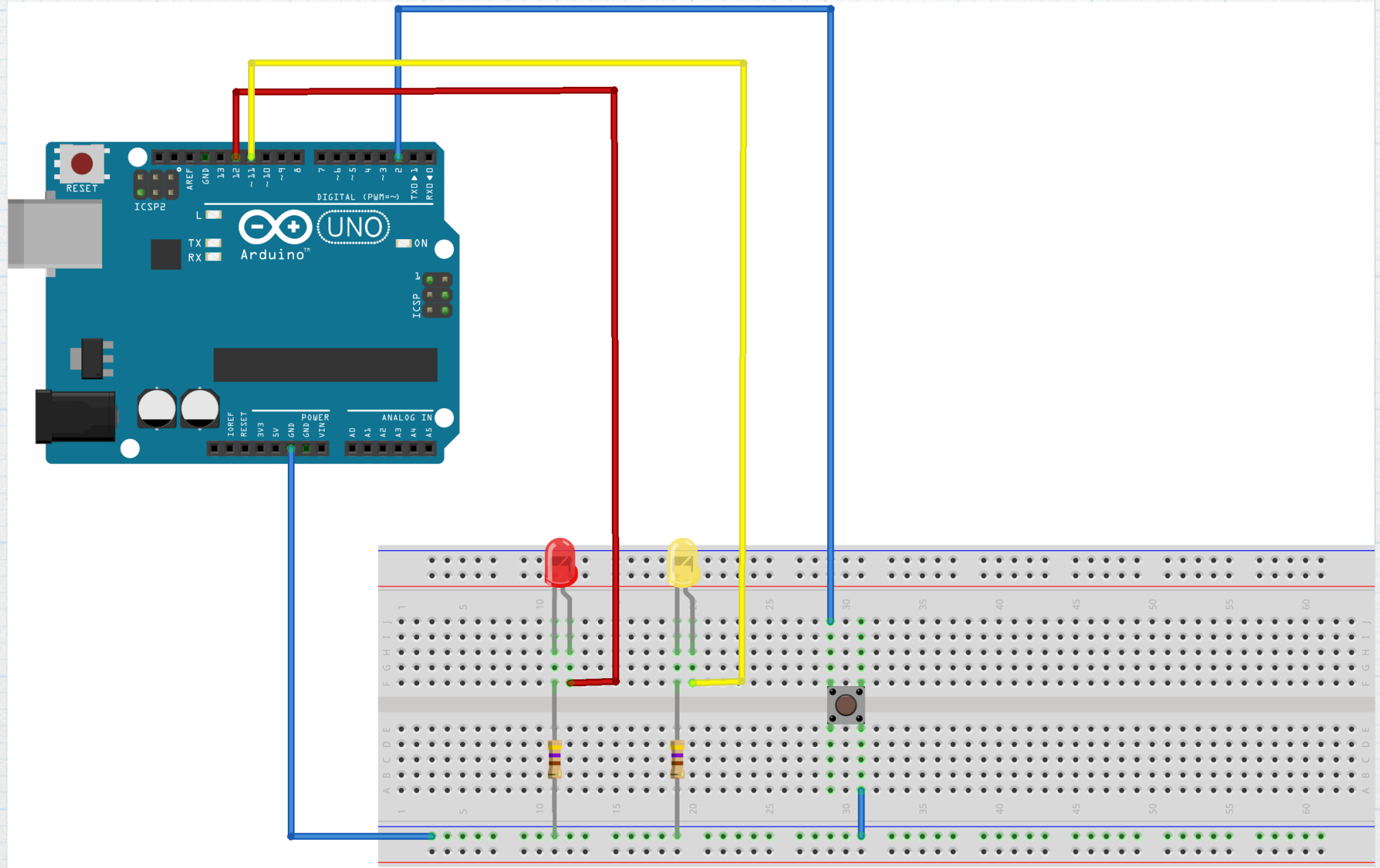
# Koşul ifadeleri

```
if (girisPini < 500)
{
    aIsiniYap;
}
else if (girisPini >= 1000)
{
    bIsiniYap;
}
else
{
    cIsiniYap;
}
```



# Uygulama!

## Buton ile LED Kontrol





# Uygulama!

## Buton ile LED Kontrol

Gerekenler;

- \* Arduino UNO
- \* 2 adet LED
- \* 2 adet 470 Ohm Direnç
- \* Jumper kablolar
- \* Breadboard
- \* Bilgisayar! :)
- \* İlgi ve merak!



# Kod!

```
// LED pinlerini ve buton pinlerini tanımlıyoruz.
const int LEDyellowPin = 11;
const int LEDredPin = 12;
const int buttonPin = 2;
int butonDurumu = 0;

void setup() {
// Buton pinini dijital giriş olarak tanımlıyoruz.
pinMode(buttonPin, INPUT);
digitalWrite(buttonPin, HIGH);
// LED pinini dijital çıkış olarak tanımlıyoruz.
pinMode(LEDyellowPin, OUTPUT);
pinMode(LEDredPin, OUTPUT);
// Serial port ile iletişimi başlatıyoruz.
Serial.begin(9600);

}

void loop() {
// Buton durumunu okuyoruz.
butonDurumu = digitalRead(buttonPin);
// Buton durumunu ekrana yazdırıyoruz.
Serial.println(butonDurumu);

/* Butona basıldığında buton durumu digital 0 (LOW) olacaktır.
Bu durumda LED çıkışlarını digital 1 (HIGH) yapıyoruz.

*/
if (butonDurumu == LOW)
{
digitalWrite(LEDredPin, HIGH);
digitalWrite(LEDyellowPin, LOW);
}
else
{
digitalWrite(LEDredPin, LOW);
digitalWrite(LEDyellowPin, HIGH);
}
}
```



**\* To be continued...**