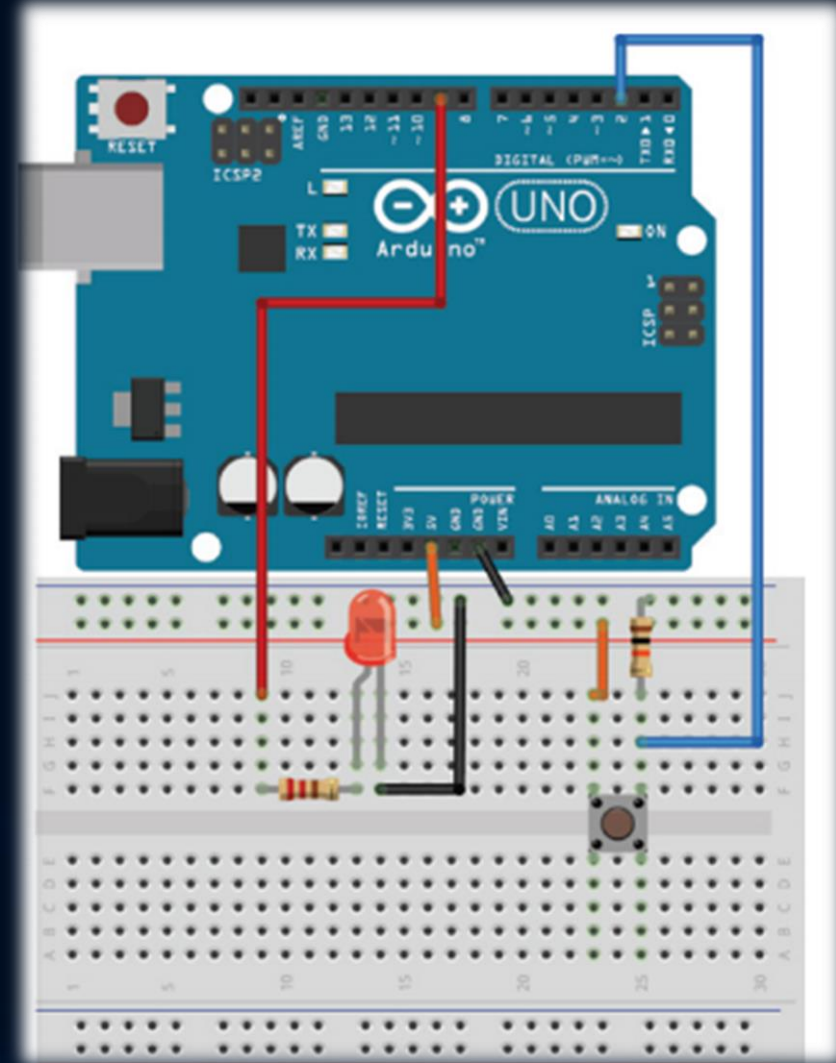


ARDUİNO VE UYGULAMALAR

Dijital Girişlerin Okunması

Ayrıca devre tahtasının güç ve topraklama bağlantılarını Arduino'ya bağladığınızdan emin olun.

Artık breadboard üzerinde birden fazla cihaz kullandığınıza göre, bu kullanışlı olacaktır.



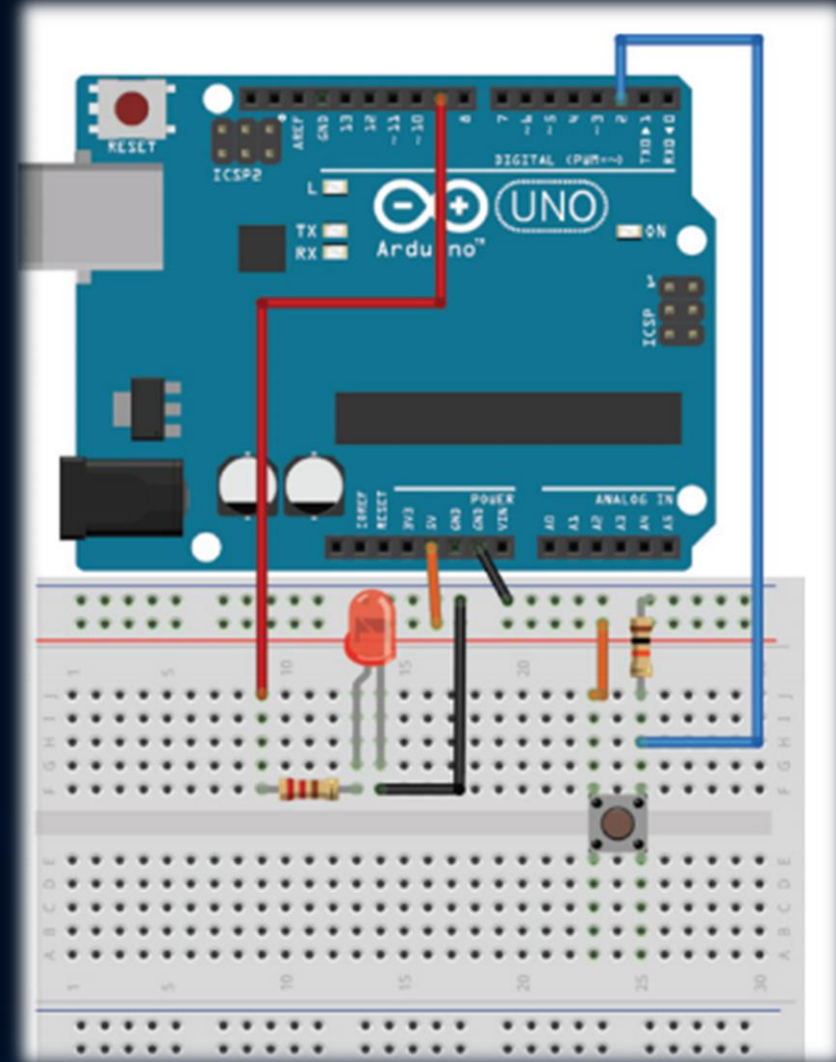
ARDUİNO VE UYGULAMALAR

Dijital Girişlerin Okunması

Düğmeye basıldığında, giriş pini düğme aracılığıyla doğrudan 5V'a bağlanır.

Şimdi, akımın iki seçeneği var:

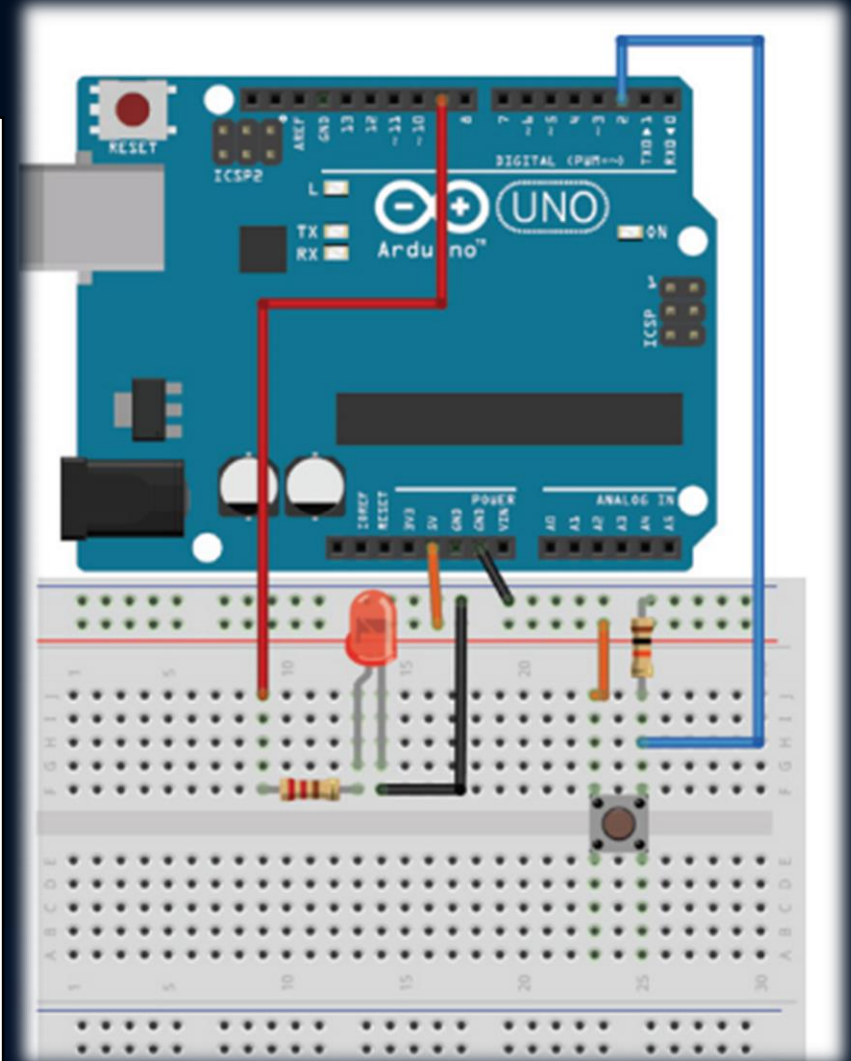
1. 5V doğrudan pin üzerinden bağlanır ve dijital pin enerjilenebilir.
2. Düğmeye basılmadığı zaman devre toprak potansiyeline bağlanır.



ARDUİNO VE UYGULAMALAR

Dijital Girişlerin Okunması

```
const int LED=9; // Pin 9 LED pini
const int BUTTON=2; // Pin 2 buton pini
void setup()
{
  pinMode (LED, OUTPUT); // Pin 2 çıkış olarak tanımla
  pinMode (BUTTON, INPUT); // Pin 2 yi giriş olarak tanımla (gerek yok)
}
void loop()
{
  if (digitalRead(BUTTON) == LOW)
  {
    digitalWrite(LED, LOW);
  }
  else
  {
    digitalWrite(LED, HIGH);
  }
}
```



ARDUİNO VE UYGULAMALAR

Dijital Girişlerin Okunması

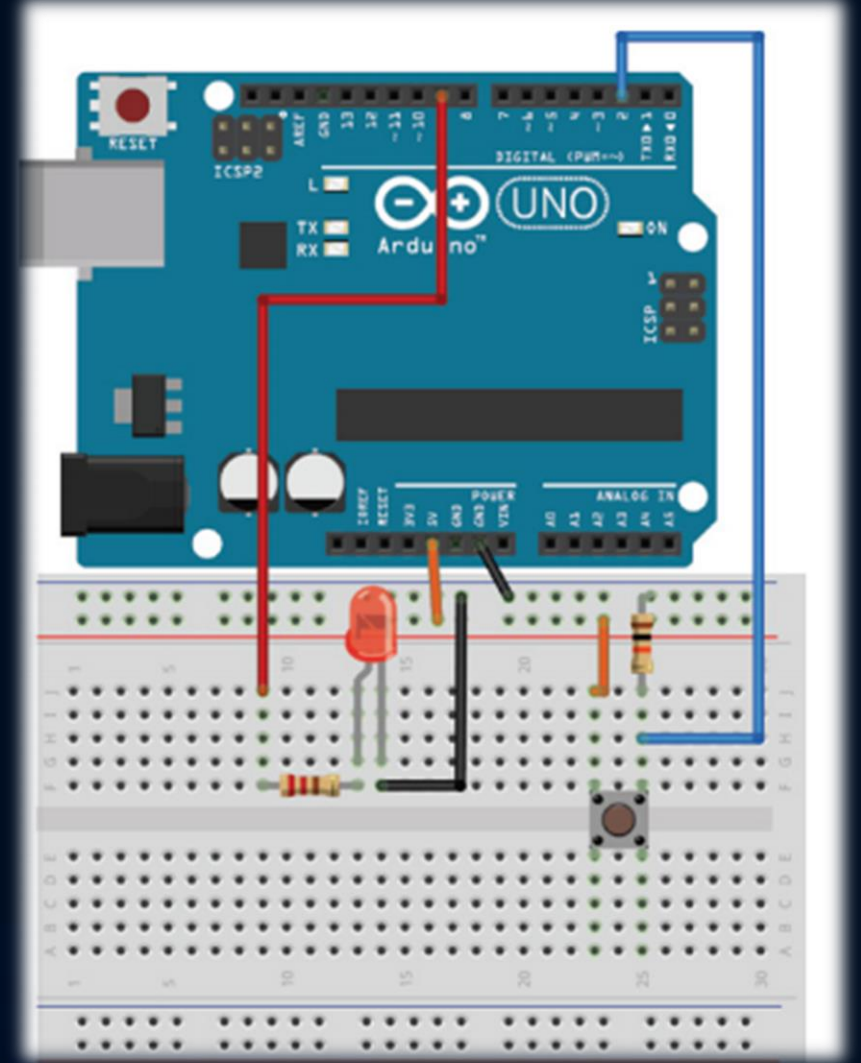
En son ne zaman ışık tutmak için bir düğmeyi basılı tutmanız gerekti? Muhtemelen asla.

Düğmeyi açmak için bir kez tıklamak ve kapatmak için düğmeye tekrar tıklamak daha mantıklıdır.

Bu şekilde, düğmesini basılı tutmanıza gerek kalmaz.

Düğmeler, yaylı amortisör sistemi olarak çalışan mekanik cihazlardır.

Başka bir deyişle, bir düğmeye bastığınızda, okuduğunuz sinyal sadece alçaktan yükseğe gitmez; yerleşmeden önce bu iki durum arasında birkaç milisaniye boyunca yukarı ve aşağı sıçrar.



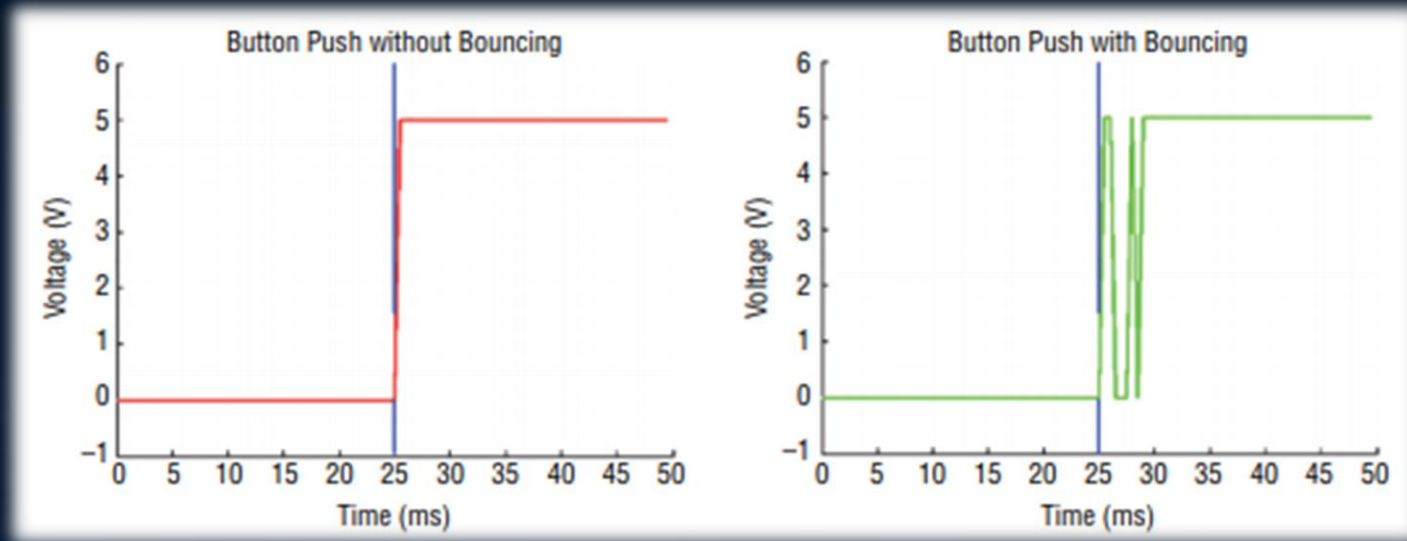
ARDUİNO VE UYGULAMALAR

Dijital Girişlerin Okunması ve Düğme salınımları

Düğme 25 msn işaretinde fiziksel olarak basılır.

Fakat bu durumda düğmede bazı istenmeyen sinyaller olabilir.

Anahtar açma yazılımı bir düğme durumu değişikliği arayabilir, sıçramanın bitmesini bekleyebilir ve ardından anahtar durumunu tekrar okuyabilir.



ARDUİNO VE UYGULAMALAR

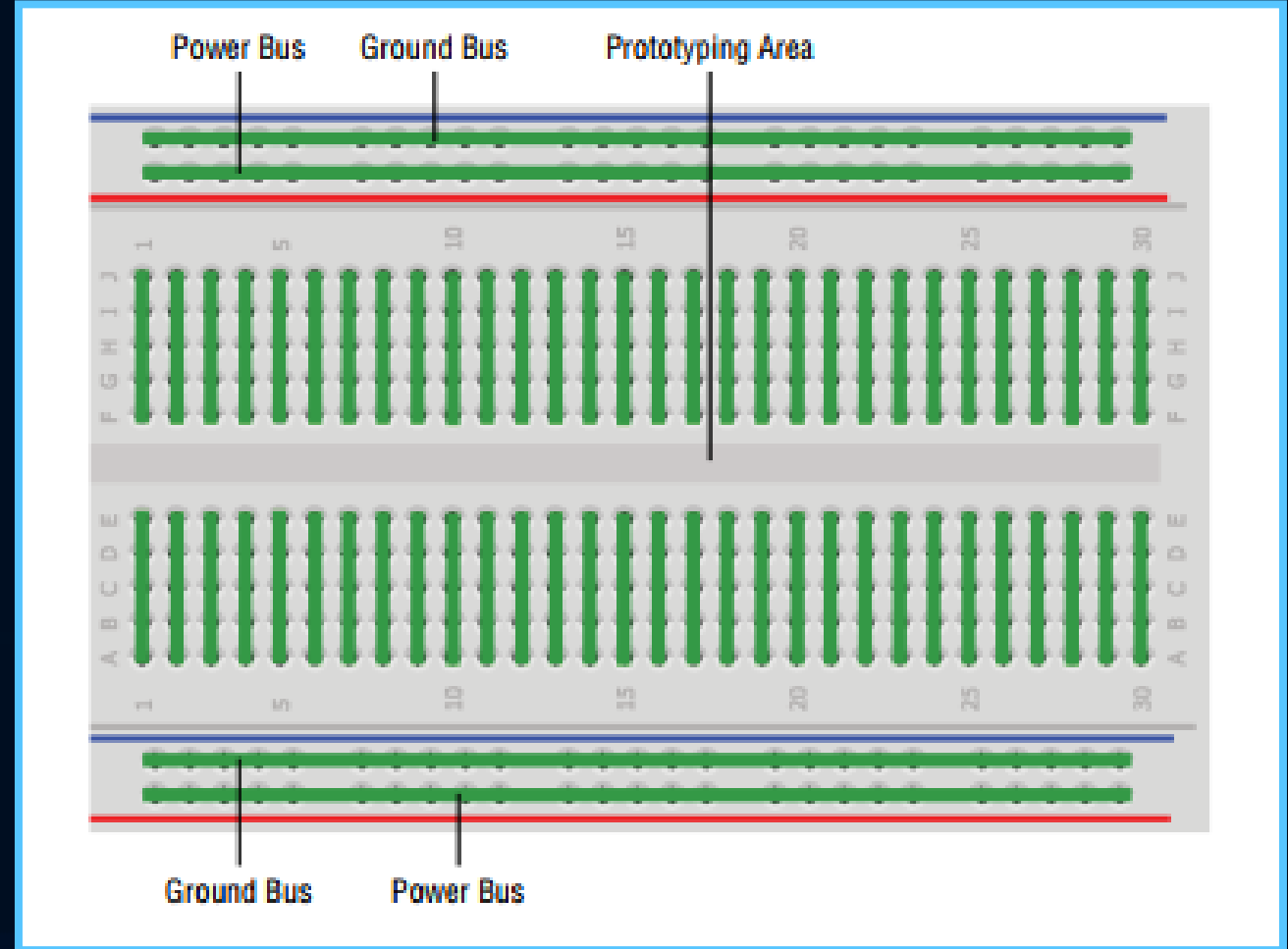
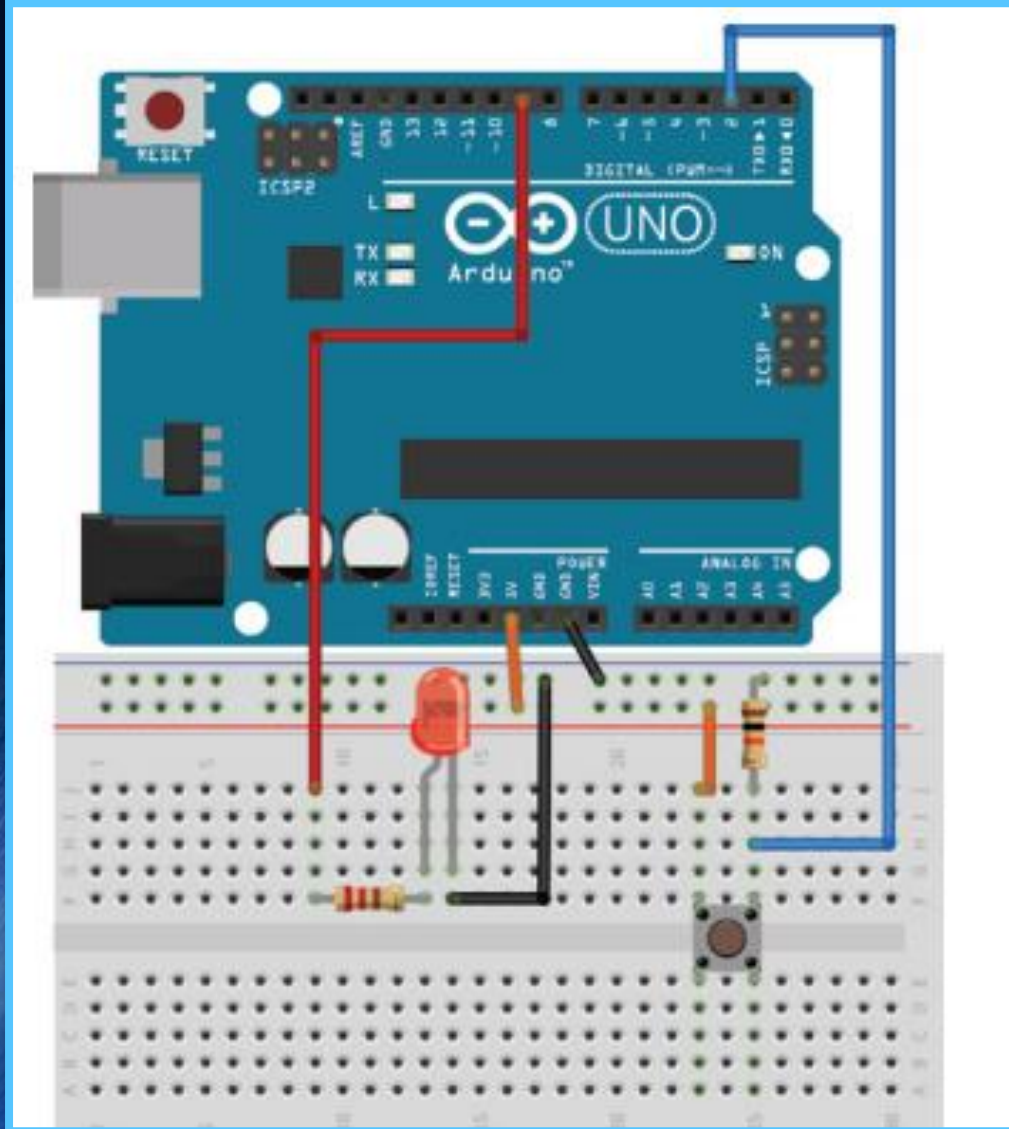
Dijital Girişlerin Okunması ve Düğme salınımları

Bu program mantık aşağıdaki gibi ifade edilebilir:

1. Önceki bir düğme durumunu ve geçerli bir düğme durumunu (DÜŞÜK olarak ayarlanmış) kaydedin.
2. Geçerli düğme durumunu okuyun.
3. Geçerli düğme durumu önceki düğme durumundan farklıysa, 5 ms bekleyin çünkü düğmenin durumu değişmiş olmalıdır.
4. 5 ms sonra, düğme durumunu yeniden okuyun ve geçerli düğme durumu olarak kullanın.
5. Önceki düğme durumu düşükse ve geçerli düğme durumu yüksekse, geçiş yapın LED durumu.
6. Önceki düğme durumunu geçerli düğme durumuna ayarlayın.
7. 2. adıma dönün.

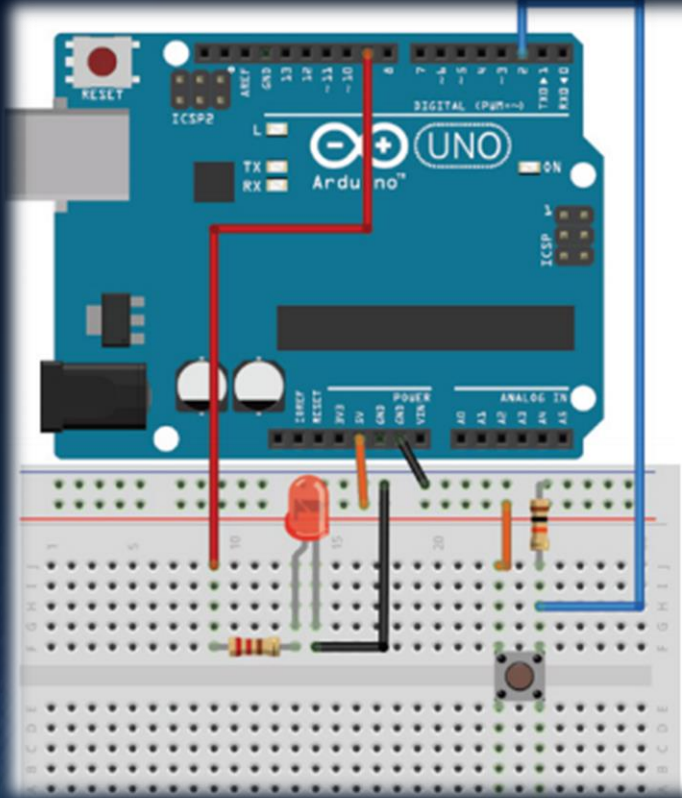
ARDUİNO VE UYGULAMALAR

Dijital Girişlerin Okunması ve Board Bağlantıları Hatırlatma



ARDUİNO VE UYGULAMALAR

Dijital Girişlerin Okunması ve Düğme salınımları



Böylece geçişler şöyle görünecek:

last = LOW, geçerli = LOW -> başlangıç durumu

last = LOW, akım = HIGH -> anahtarına basıldı ve LED durumu değiştirildi

last = HIGH, akım = DÜŞÜK -> anahtar bırakıldı, hiçbir şey yapma

last = LOW, akım = LOW -> başlangıç durumuna geri dön

```
const int LED=9; // The LED is connected to pin 9
const int BUTTON=2; // The Button is connected to pin 2
boolean lastButton = LOW; // Variable containing the previous
// button state
boolean currentButton = LOW; // Variable containing the current
// button state
boolean ledOn = false; // The present state of the LED (on/off)
void setup()
{
  pinMode (LED, OUTPUT); // Set the LED pin as an output
  pinMode (BUTTON, INPUT); // Set button as input (not required)
}
boolean debounce(boolean last)
{
  boolean current = digitalRead(BUTTON); // Read the button state
  if (last != current) // if it's different...
  {
    delay(5); //Wait 5ms
    current = digitalRead(BUTTON); //Read it again
  }
  return current; //Return the current value
}
void loop()
{
  currentButton = debounce(lastButton); //Read debounced state
  if (lastButton == LOW && currentButton == HIGH) //if it was pressed...
  {
    ledOn = !ledOn; //Toggle the LED value
  }
  lastButton = currentButton; //Reset button value
  digitalWrite(LED, ledOn); //Change the LED state
}
```