

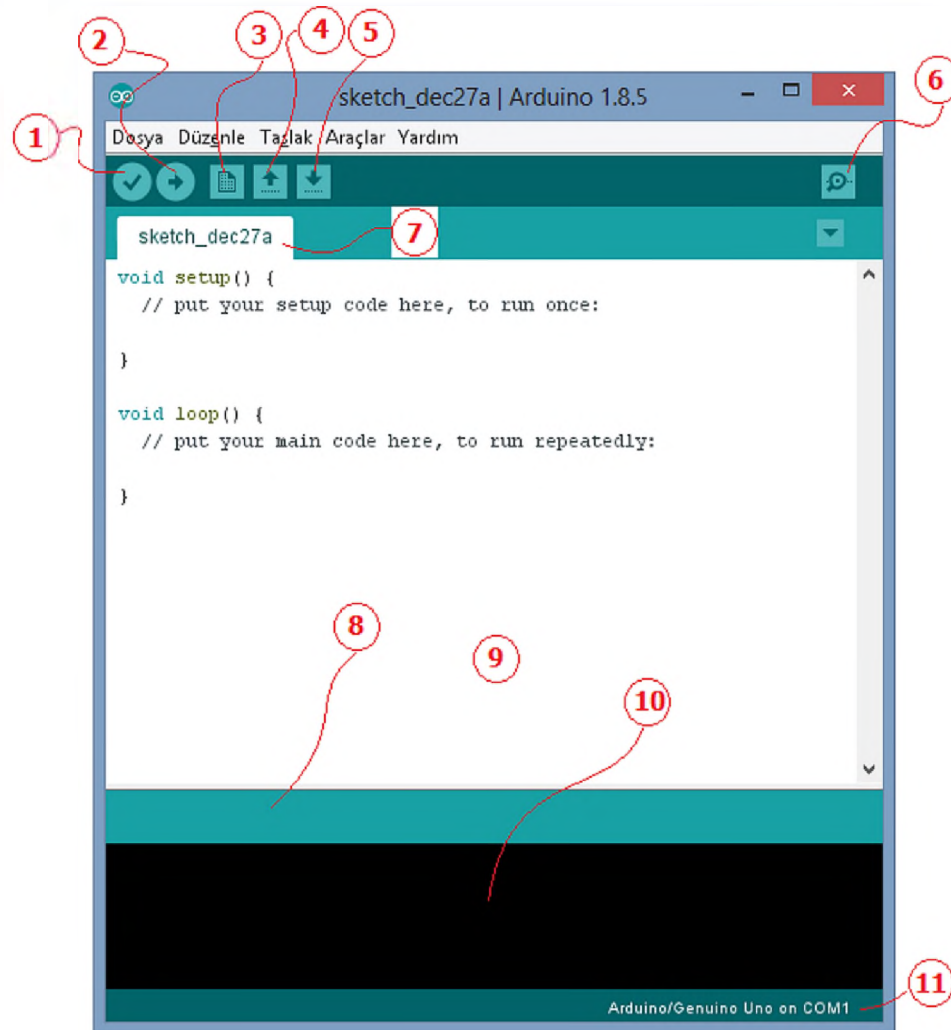
# METİN TABANLI ROBOT PROGRAMLAMADA KULLANILAN YAPILAR



---

**Mustafa NUMANOĞLU**

# Arduino IDE'nin Yapısı



# Arduino IDE'nin Yapısı

- **1. Kontrol Et:** Yazılan kodları derler ve hataları bulur.
- **2. Yükle:** Yazılan programı Arduino kartına yükler.
- **3. Yeni:** Yeni çalışma sayfası açar.
- **4. Aç:** Kayıtlı bir programı açar.
- **5. Kaydet:** Yazılan programı kaydeder.
- **6. Seri Port Ekranı:** Arduino ile seri iletişim yaparak ekran açar.
- **7. Sketch:** Yazılan programın dosya ismini gösterir.
- **8. Gösterge:** Yaptığı işlemin ilerleme durumunu gösterir.
- **9. Boş alan:** Yazılacak program alanıdır.

# Arduino IDE'nin Yapısı

- **10. Rapor:** Derleme sonucu varsa yapılan hataları, yoksa programın yükleme sonrası mikrodenetleyicide kapladığı alanı gösterir.
- **11. Gösterge:** Bilgisayara USB ile bağlanan Arduino'nun bağlandığı portu ve hangi Arduino modeli ile çalışılıyorsa onu gösterir.

# Arduino Tümlleşik Geliştirme Ortamının Temel Özellikleri

- Arduino IDE tümlleşik geliştirme ortamında basitleştirilmiş C++ kullanılır.
- Arduino programları genellikle tanımlamalar, kurulum ve ana program bloğu olmak üzere uç bolümden oluşur.
- Program yazımı belirli kalıpta, bloklar halinde gerçekleştirilir.
- Program kodları renkli olarak gösterilir. Kodların bulunduğu yerlerde gri renkte olan yazılar kodun ne işe yaradığı hakkında bilgi vermek için kullanılır.
- Arduino'ya yüklenen programlar kaldırılana kadar Arduino içinde kalır. Yüklemeden sonra bağımsız olarak çalıştırılabilir.
- Bloklar { } parantezleri ile oluşturulur.

# Arduino Tümlleşik Geliştirme Ortamının Temel Özellikleri

- Komutlar aynı veya alt alta satırlara yazılabilir. Fakat programın anlaşılabilirliği açısından alt alta yazmak daha uygundur.
- Tüm komutlar noktalı virgül (;) ile biter. Fakat blok başlatan ifadelerden sonra noktalı virgül kullanılmaz.
- Programda kullanılan tüm değişkenler ve bilgi tipleri bildirilir.
- Programın başında kullanılacak kütüphaneler aktifleştirilir /çağrılır.
- Açıklamalar “//” ve “/\* \*/” (birden fazla satır için) ile yazılır.
- Türkçe karakter kullanılmamalıdır. Fakat açıklama satırları içerisinde (derleme işlemine dahil edilmediğinden) kullanılabilir.
- Eşdeğer ifadeler `#define` ile atanır.
- Kütüphaneler `#include` ile çağrılır.

# Arduino IDE'nin Program Yapısı

A. SÖZ DİZİMİ VE OPERATÖRLER	
<b>1. Söz Dizimi</b>	<b>5. İşaretçi Operatörler</b>
<ul style="list-style-type: none"><li>; Noktalı Virgül</li><li>{ } Süslü Parantez</li><li>// Çift Slash</li><li>/**/ Yıldızlı Slash</li><li>#define</li><li>#include</li></ul>	<ul style="list-style-type: none"><li>* Referans dışı operatör</li><li>&amp; Referans operatörü</li></ul>
<b>2. Aritmetik Operatörler</b>	<b>6. Bitisel Operatörler</b>
<ul style="list-style-type: none"><li>= Atama İşleci</li><li>+ Toplama</li><li>- Çıkarma</li><li>* Çarpma</li><li>/ Bölme</li><li>% Modulo</li></ul>	<ul style="list-style-type: none"><li>&amp; (Bitisel Ve)</li><li>   (Bitisel Veya)</li><li>^ (Bitisel Xor)</li><li>~ (Bitisel Değil)</li><li>&lt;&lt; (Bitshift Sol)</li><li>&gt;&gt; (Bitshift Sağ)</li></ul>
<b>3. Karşılaştırma Operatörleri</b>	<b>7. Birleşik Operatörler</b>
<ul style="list-style-type: none"><li>== (eşit eşit)</li><li>!= (eşit değil)</li><li>&lt; (küçük)</li><li>&gt; (büyük)</li><li>&lt;= (küçük eşit)</li><li>&gt;= (büyük eşit)</li></ul>	<ul style="list-style-type: none"><li>++ (arttırma)</li><li>-- (azaltma)</li><li>+= (Birleşik artırma)</li><li>-= (Birleşik çıkarma)</li><li>*= (Birleşik çarpma)</li><li>/= (Birleşik bölme)</li><li>%= (Birleşik mod)</li><li>&amp;= (Bitisel Lojik Ve)</li><li> = (Bitisel Lojik Veya)</li></ul>
<b>4. Boolean Operatörleri</b>	
<ul style="list-style-type: none"><li>&amp;&amp; (ve)</li><li>   (veya)</li><li>! (değil)</li></ul>	

# Arduino IDE'nin Program Yapısı

<b>B. DEĞİŞKENLER</b>	
<b>1. Sabitler</b>	<b>3. Dönüşümler</b>
HIGH   LOW INPUT   OUTPUT   INPUT_PULLUP LED_BUILTIN true   false integer constants floating point constants	char() byte() int() word() long() float()
<b>2. Veri Tipleri</b>	<b>4. Değişken Kapsamları</b>
void boolean char unsigned char byte int unsigned int word long unsigned long short float double string - char array substring - object array	static volatile const
	<b>5. Yardımcılar</b>
	PROGMEM sizeof()



# Arduino IDE'nin Program Yapısı

## C. PROGRAM YAPISI

### 1. Temel Yapı

```
void setup()  
void loop()
```

### 2. Kontrol Yapıları

```
if  
if/else  
for  
switch/case  
while  
do/while  
break  
continue  
return  
goto
```

# Arduino IDE'nin Program Yapısı

D. FONKSİYONLAR		
<b>1. Dijital Giriş Çıkışlar</b> pinMode(pin,mod) digitalWrite(pin,değer) digitalRead(pin)	<b>6. Trigonometri İşlevleri</b> sin(rad) cos(rad) tan(rad)	<b>10. Harici İnterruptlar (Kesmeler)</b> attachInterrupt(interrupt, function, mode) detachInterrupt(interrupt)
<b>2. Analog Giriş Çıkışlar</b> analogRead(pin,mod) analogWrite(pin,değer) analogReference(tip) analogReadResolution () analogWriteResolution ()	<b>7. Karakterler</b> isAlphaNumeric() isAlpha() isAscii() isWhiteSpace() isControl() isDigit() isGraph() isPrintable() isPunct() isSpace() isUpperCase() isHexadecimalDigit()	<b>11. İnterruptlar (Kesmeler)</b> interrupts() noInterrupts()
<b>3. Gelişmiş Giriş Çıkışlar</b> tone() noTone() shiftOut() shiftIn() pulseIn()		<b>12. Seri Haberleşme</b> Serial.begin(hız) int Serial.available() int Serial.read() Serial.flush() Serial.print(data) Serial.println(data)
<b>4. Gecikmeler</b> delay(milisaniye) unsigned long millis() delayMicroseconds(mikrosaniye)	<b>8. Rasgele Sayılar</b> randomSeed() random()	<b>13. Haberleşme Protokolleri</b> I2C Veri Yolu SPI Veri Yolu
<b>5. Matematiksel İşlevler</b> min(x,y) max(x,y) abs(x) constrain(x, a, b) map(value, fromLow, fromHigh, toLow, toHigh) pow(base, exponent) sqrt(x)	<b>9. Bit ve Bayt'lar</b> lowByte() highByte() bitRead() bitWrite() bitSet() bitClear() bit()	