

EĐİTSEL ROBOT PROGRAMLAMA YAZILIMLARI VE ORTAMLARI



Mustafa NUMANOĐLU

Blok Tabanlı Robot Programlama Yazılımları ve Ortamları

- Robot programlama için oluşturulmuş metin tabanlı programlama yazılımlarına göre daha kullanıcı dostu, öğrenici ve hobi kullanıcılarına hitap eden diller ve ortamlar da bulunmaktadır. Birçoğu ücretsiz olan bu araçlarla hiçbir kod kullanmadan, sürükle bırak veya yapboz oynar gibi programlar oluşturmak olanaklı hale gelmiştir. Bu tür ortamlara blok programlama ortamları adı verilmektedir.
- Lego NXT-G, EV3, **Enchanting**, Robo Pro, **Modkit**, **miniBlog**, **Ardublock**, **Snap4Arduino**, **Srach for Arduino (S4A)** ve **mBlock** bu ortamlara örnek olarak verilebilir.
- Bu dersimizde blok tabanlı programlama yazılımı ve ortamı olarak mBlock tercih edilmiştir.

Metin Tabanlı Robot Programlama Yazılımları ve Ortamları

- Robot programlamak amacıyla kullanılan pek çok programlama yazılımı ve ortamı bulunmaktadır. Robot programlama için kullanılan diller incelendiğinde geleneksel dillere robotik kontrolleri kolaylaştıran yapıların eklenmesiyle oluşturduğu görülür. **Robot C** ve **Parallax Propeller C** bu alanda dikkat çeken C dilinin robotik programlamaya uyarlanmış metin temelli sürümleridir.
- Burada metin tabanlı programlama yazılımı olarak **Arduino IDE** tercih edilmiştir. **Arduino IDE** robotik programlamada oldukça yaygın kullanılan tümleşik geliştirme ortamıdır.

Blok Tabanlı Eğitsel Robot Programlama Ortamları ve Dilleri

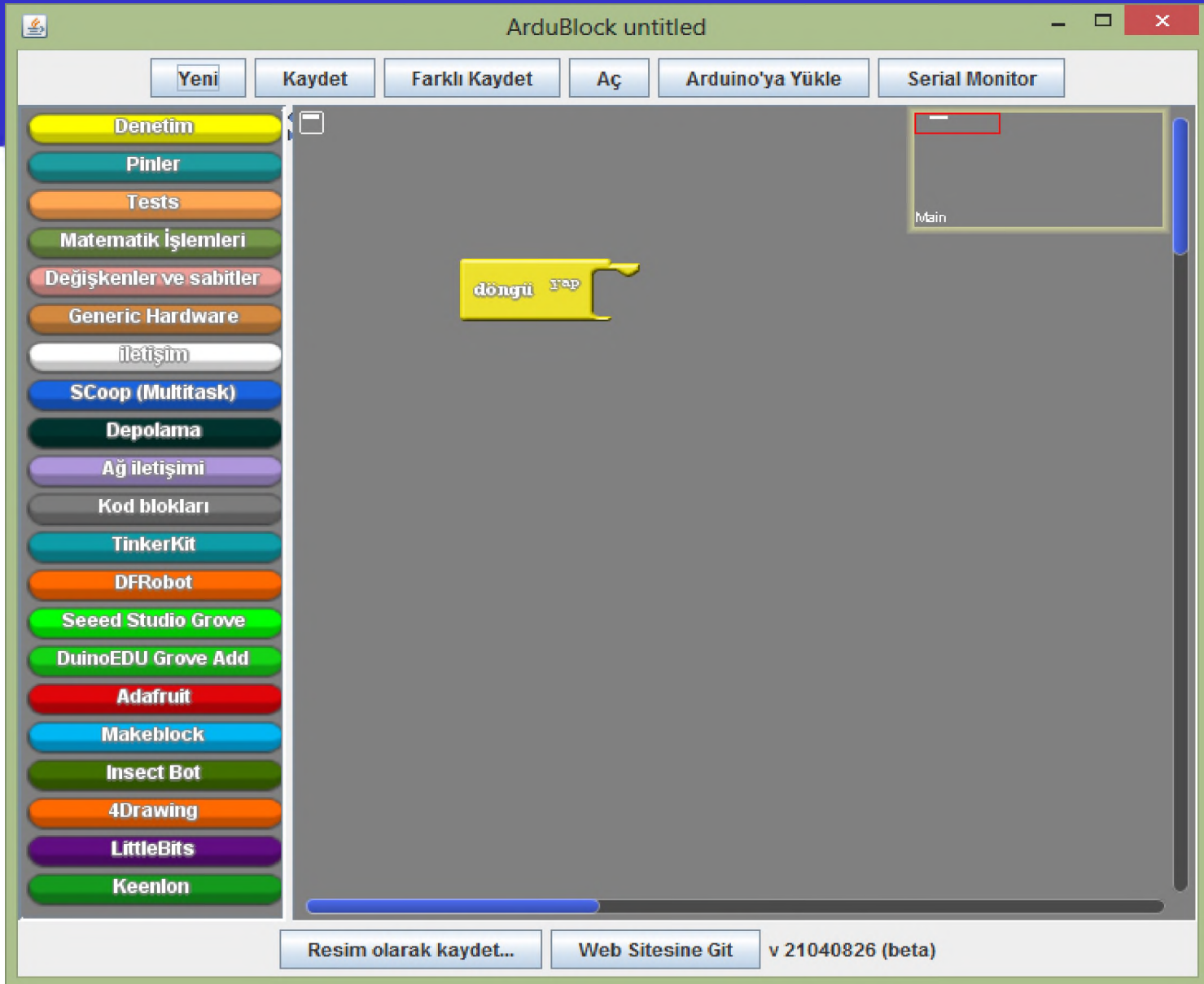
- Bu çalışmanın amacı bilgisayar bilimi dersinin robot programlama ünitesinde kullanılabilecek blok tabanlı eğitsel robot programlama ortam ve dillerini ortak bir uygulama örneğiyle tanıtmaktır.
- Bu amaçla çalışmanın ilk aşamasında Ortaöğretim Bilgisayar Bilimi Dersi Robot Programlama Ünitesi (Kur 2) Öğretim Programı incelenmiştir.
- Programdan beklenen öğrenme çıktıları ve alan yazındaki çalışmalar incelenerek bu dersin öğrenme-öğretme sürecinde kullanılabilecek blok tabanlı eğitsel robot programlama ortam ve dilleri belirlenmeye çalışılmıştır.

Seçilen Blok Tabanlı Eğitsel Robot Programlama Ortam ve Dillerin Yapısı ve Genel Özellikleri

- İncelemeler sonucunda ders için kullanılabilecek blok tabanlı eğitsel robot programlama ortam ve dilleri olarak Ardublock, mBlock, miniBloq, Scratch for Arduino (S4A) ve Snap4Arduino seçilmiştir.
- Seçilen ortam ve dillerin kısa tanıtımı yapılmış, çalışmanın sonraki aşamasında bu ortam ve dillerin programlama yapısı ve kullanım şekilleri geliştirilen ortak bir uygulama programı ile tanıtılmıştır.
- Bu amaçla robot programlama çalışmalarında temel uygulamalardan biri olan ve “if/else” yapısını öğretmeyi amaçlayan örnek bir program, seçilen her ortam için ayrı ayrı hazırlanmıştır.

Seçilen Blok Tabanlı Eğitsel Robot Programlama Ortam ve Dillerin Yapısı ve Genel Özellikleri

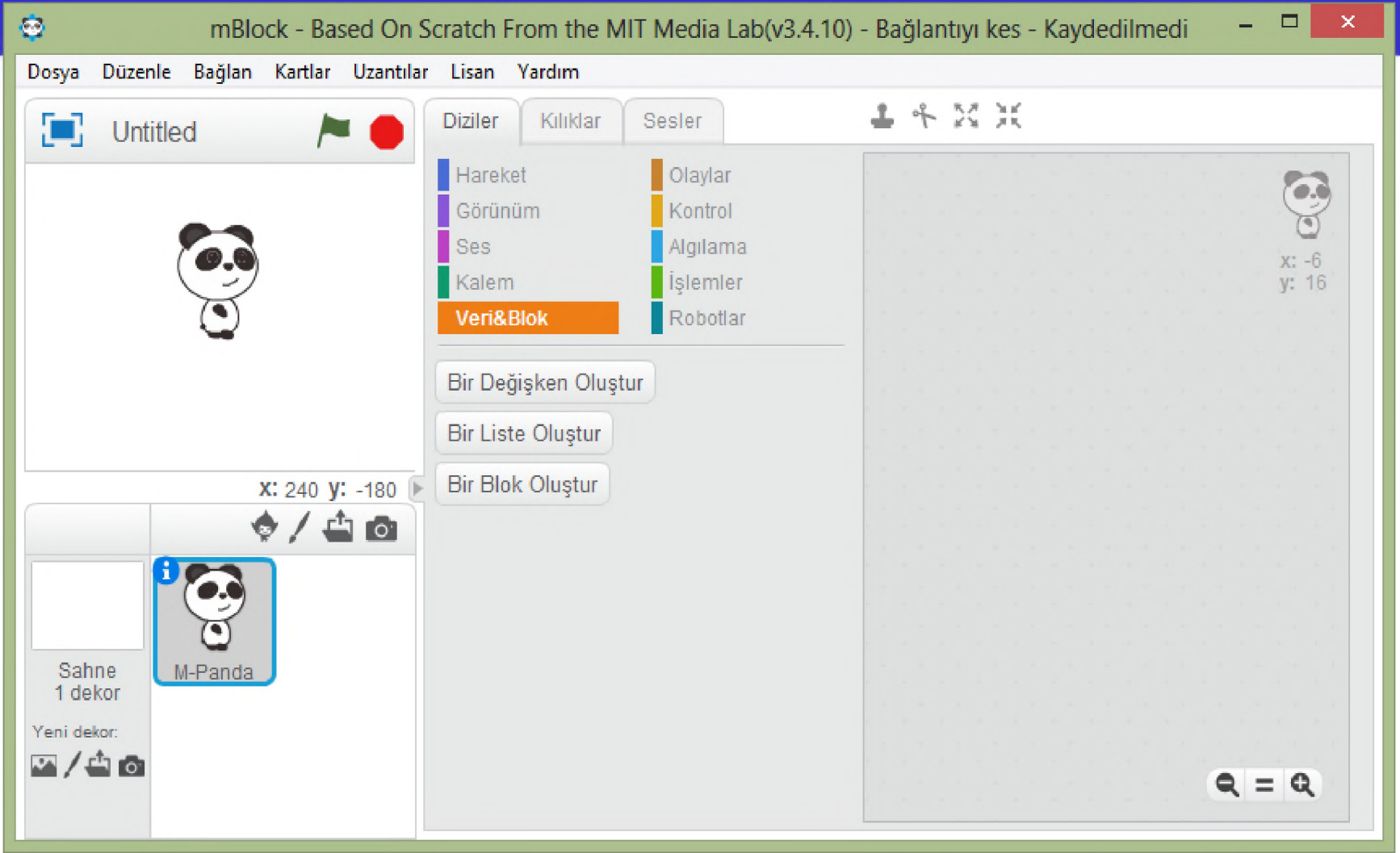
- **Ardublock:** Ardunio için grafik ve blok tabanlı açık kaynak kodlu görsel programlama dili eklentisidir.
- Scratch mantığıyla çalışmaktadır. Arduino ve Ardunio temelli kartlar (Scoop, Adafruit, DFrobot, TinkerKit vb. üreticilerin kartları) ve Ardunio temelli robotlar programlanabilmektedir.
- Programın kullanılabilmesi için Ardublock ile birlikte Ardunio IDE'nin de kurulması gerekmektedir.
- <http://blog.ardublock.com/>



Şekil 1: Ardublock Uygulama Ortamı

Seçilen Blok Tabanlı Eğitsel Robot Programlama Ortam ve Dillerin Yapısı ve Genel Özellikleri

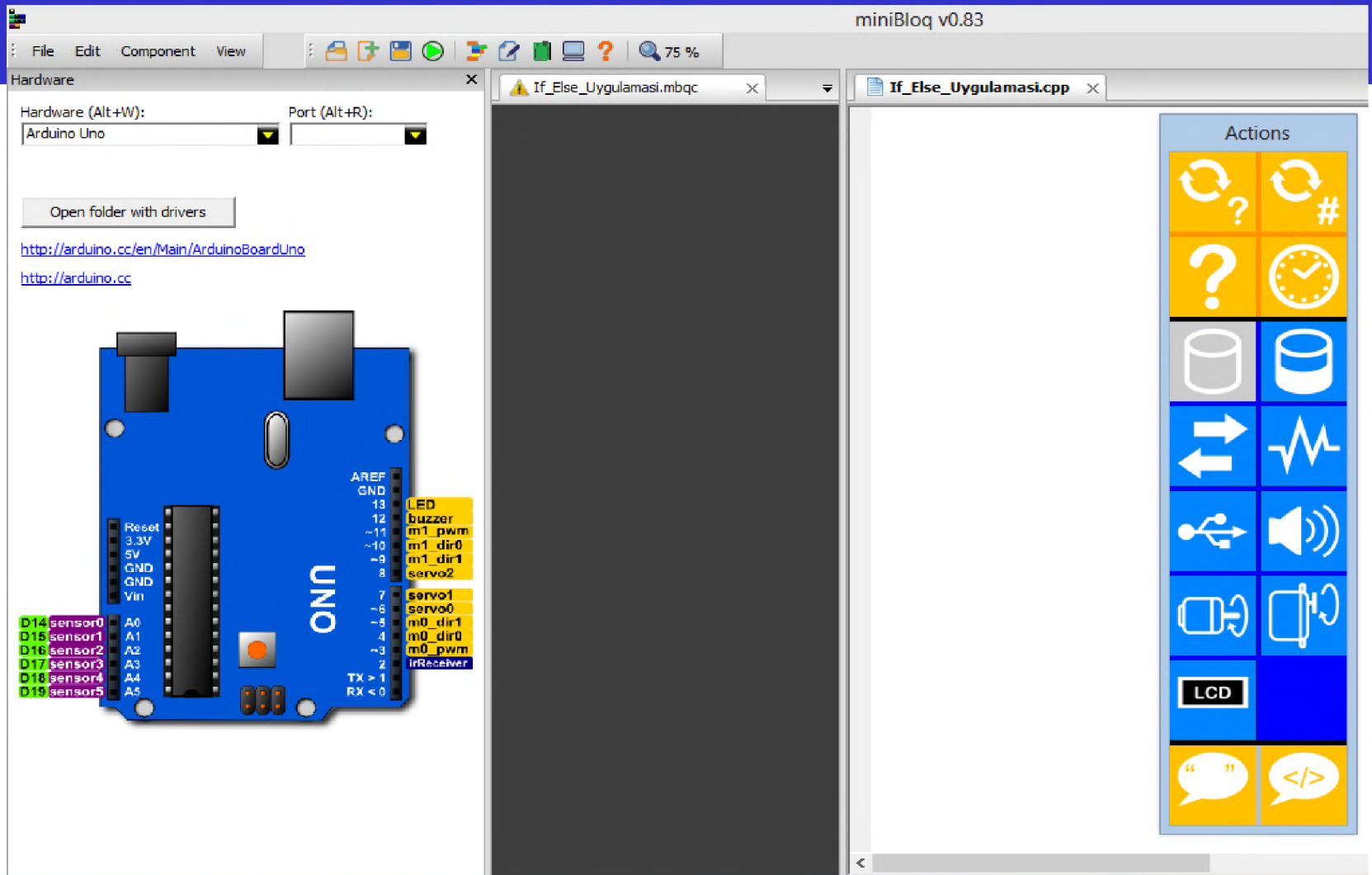
- **mBlock:** Makeblock tarafından geliştirilen ve kendi geliştirdiği eğitsel robotların programlanması için kullanılan mBlock, grafik ara yüzü görsel programlama ortamıdır.
- Fiziksel dünya ile etkileşim içinde interaktif uygulamalar (oyun, hikâye, animasyon) ve kablosuz programlama robotları oluşturmak için modüler ve genişletilebilir şekilde tasarlanmıştır.
- Arduino temelli robotların ve kartların programlanmasında da kullanabilmektedir. Bu amaçla Arduino Board Standartlarını desteklemekte ve Arduino uyumlu kartlarla kullanılabilir.
- <http://www.mblock.cc/>



Şekil 2: mBlock Uygulama Ortamı

Seçilen Blok Tabanlı Eğitsel Robot Programlama Ortam ve Dillerin Yapısı ve Genel Özellikleri

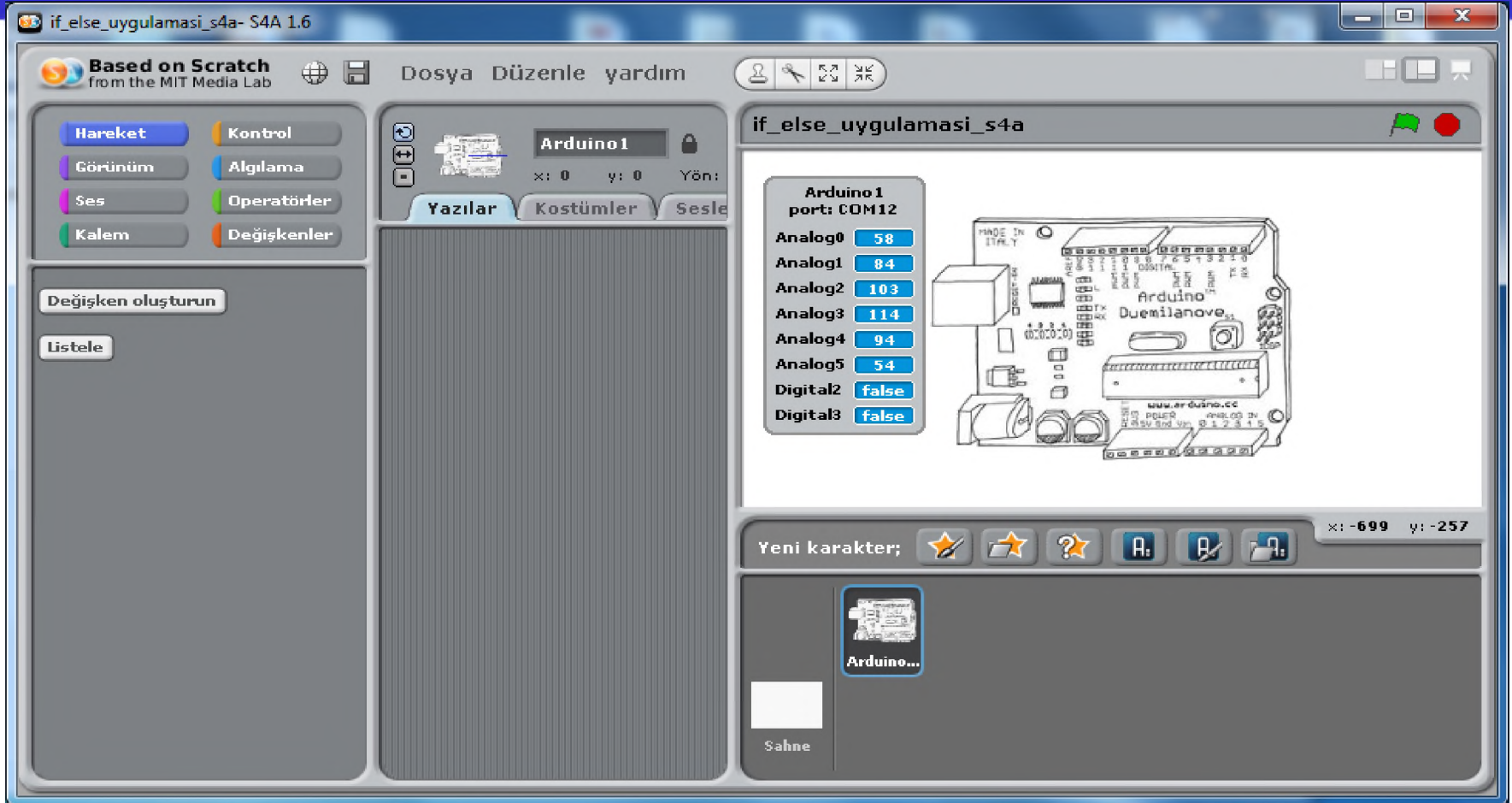
- **miniBloq**: Arduino türleri ve Maple gibi mikrodenetleyici kartlar ve Multiplo ve Duino gibi eğitim robotlar için açık kaynaklı, sezgisel grafik programlama ortamıdır.
- Gerçek zamanlı kod üretici ve hata kontrolü içermektedir. Çoğu hatayı gerçek zamanlı olarak doğrulamakta ve kırmızı renk ile işaretlemektedir.
- Otomatik değişken ad yönetimi sayesinde çok fazla sözdizimsel hatayı önleme özelliği bulunmaktadır. Kullanıcının bir bloğa yanlış veri türü parametreleri eklemesi mümkün değildir.
- Blok eklerken veya parametre değerlerini değiştirirken kod oluşturulmakta ve kod sözdizimi renkli olarak bir pencerede gösterilmektedir.
- <http://blog.minibloq.org/>



Şekil 3: miniBloq Uygulama Ortamı

Seçilen Blok Tabanlı Eğitsel Robot Programlama Ortam ve Dillerin Yapısı ve Genel Özellikleri

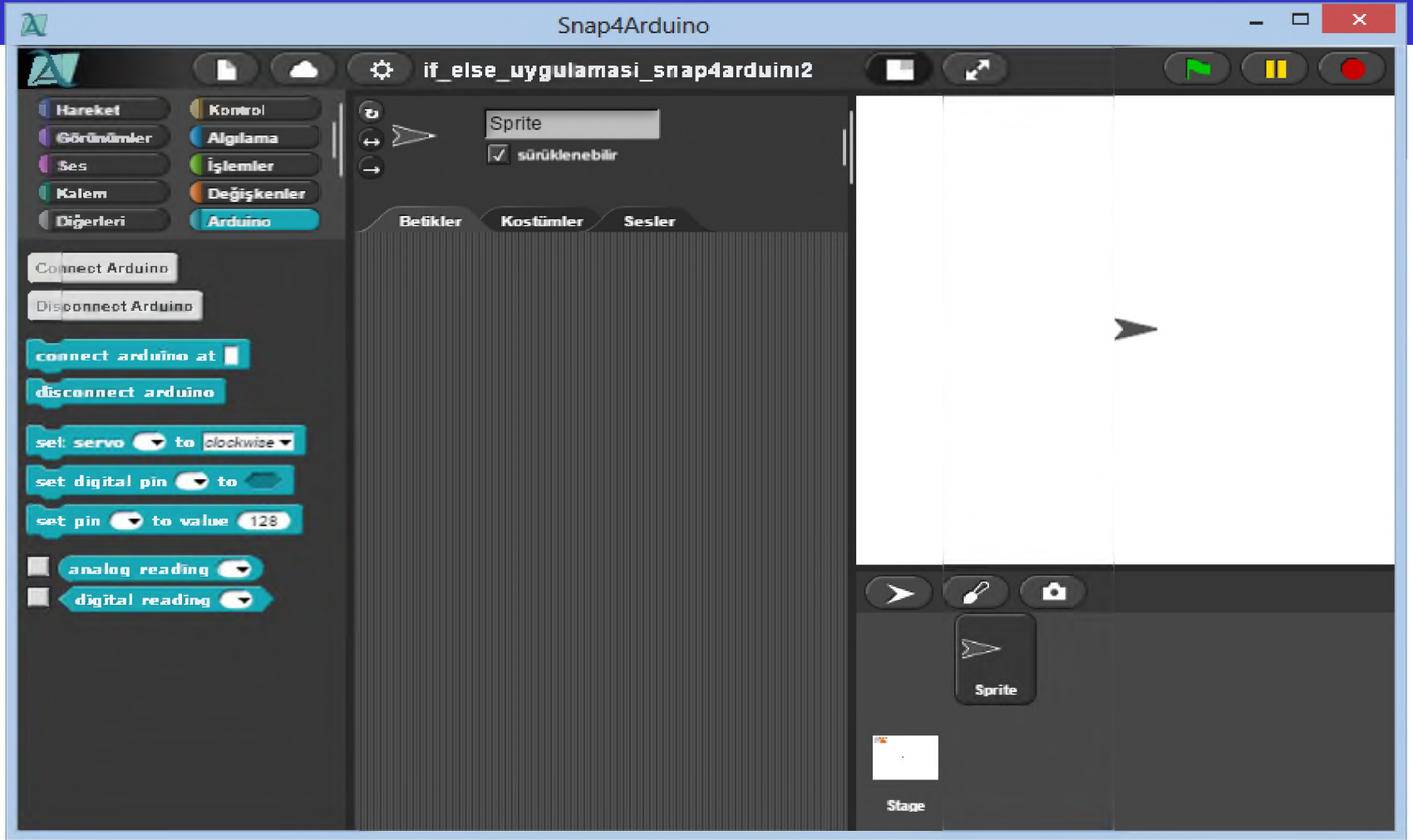
- **Scratch for Arduino (S4A):** S4A yazılımı Scratch tabanlı açık kaynaklı bir görsel programlama dilidir.
- İçerisinde Arduino'yu kontrol etmeyi, algılayıcıları ve aktüatörleri yönetmeyi sağlayan belirli programlama blokları bulunmaktadır.
- Arduino ve Ardunio temelli kart ve robotların programlanması mümkündür.
- Arduino'yu programlamadan önce, S4AFirmware16.ino isimli programın Arduino kartına yüklenmesi gerekmektedir.
- Program S4A ile Arduino arasında iletişim kurularak Arduino temelli kart ve robotların tanınmasını sağlamaktadır.
- <http://s4a.cat/>



Şekil 4: Scratch for Arduino (SA4) Ortamı

Seçilen Blok Tabanlı Eğitsel Robot Programlama Ortam ve Dillerin Yapısı ve Genel Özellikleri

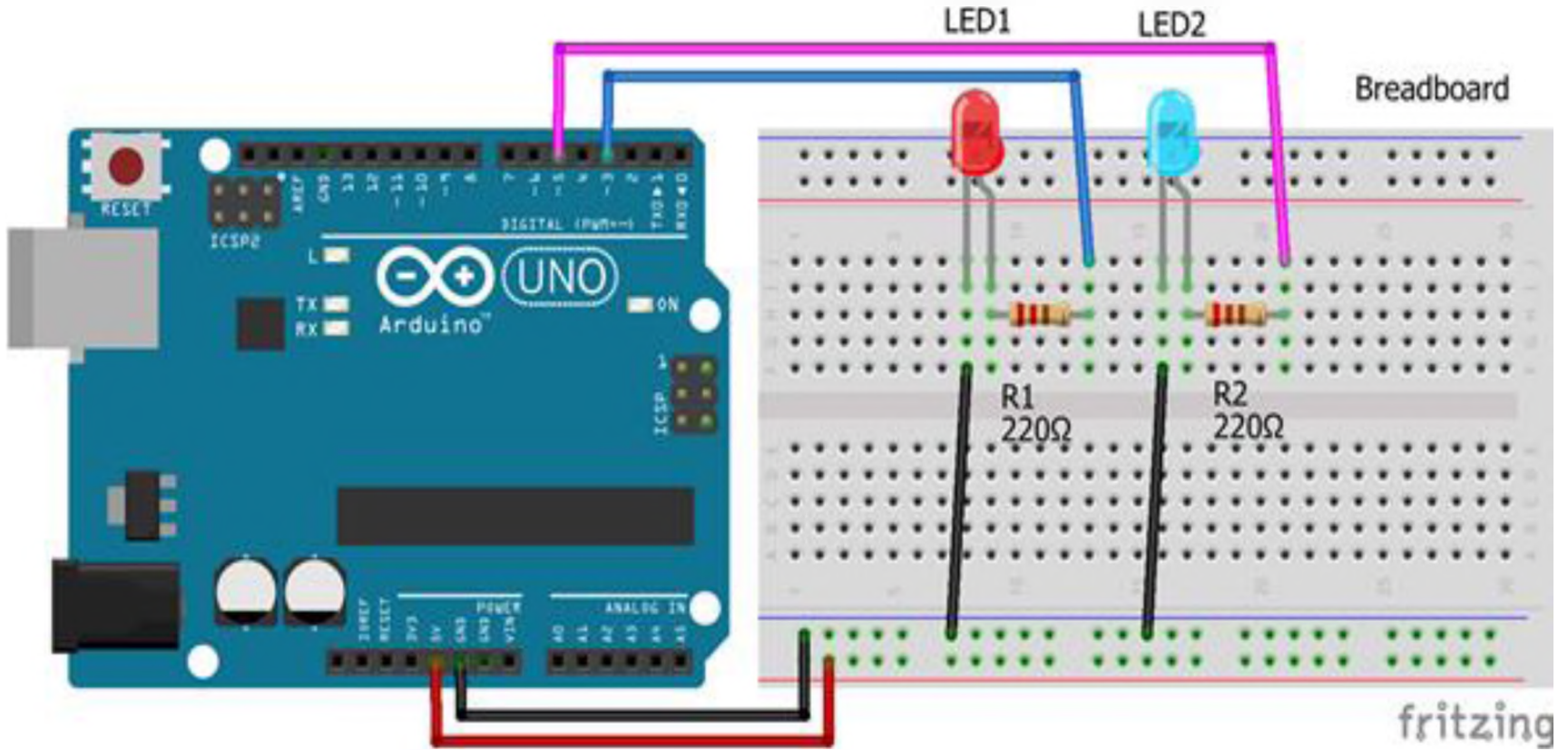
- **Snap4Arduino:** Snap4Arduino bir görsel programlama dilidir.
- Kullanıcının kendi bloklarını oluşturmasını sağlayan bir yapısı bulunmaktadır.
- Ayrıca, birinci sınıf heterojen listeleri, birinci sınıf prosedürleri ve süreklilikleri içermektedir. Bu ek yetenekler, lise veya üniversite öğrencileri için bilgisayar bilimlerine girişte onu önemli bir araç haline getirmektedir.
- Hemen hemen tüm Arduino kartları tarafından desteklenmektedir.
- Aynı anda birden fazla Arduino kartları ile etkileşime geçebilmektedir.
- Arduino kartlarıyla iletişim için Standart Firmata yazılımı kullanılmaktadır.
- <http://snap4arduino.rocks/>



Şekil 5: Snap4Arduino Ortamı

Hazırlanan Blok Tabanlı Eğitsel Robot Programlama Uygulamaları

- Seçilen robot programlama ortamlarının programlama yapısını, aralarındaki benzerlik ve farklılıkları göstermek amacıyla iki LED'in yanma kontrolünü sağlayan "if/else" uygulaması için bir devre tasarlanmıştır.
- Bu amaçla Arduino UNO'nun 3 ve 5 numaralı pwm pinlerine bağlı 2 LED ve 2 adet 220 Ω direnç kullanılmıştır.
- Devrenin çizimi için fritzing yazılımı kullanılmıştır.
- Bu devre için geliştirilen uygulamalarının tümünde birinci LED'in ışık gücü giderek artırılarak en parlak noktaya ulaşırken, diğeri en parlak noktada yanarken ışık gücü giderek düşürülmektedir.



Şekil 6: if/else Uygulaması Breadboard Çizimi



Şekil 7: Ardublock Uygulaması

Arduino Programı

nLed , 0 olsun

mLed , 100 olsun

sürekli tekrarla

eğer nLed < 100 ise

nLed 'i 5 arttır

mLed 'i -5 arttır

3 pwm pini nLed yap

5 pwm pini mLed yap

0.1 saniye bekle

değilse

nLed , 0 olsun

mLed , 100 olsun

geri al Arduinoya Yükle Arduino BGO'yla düzenle

```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 double angle_rad = PI/180.0;
6 double angle_deg = 180.0/PI;
7 double nLed;
8 double mLed;
9
10 void setup() {
11     nLed = 0;
12     mLed = 100;
13     pinMode(3,OUTPUT);
14     pinMode(5,OUTPUT);
15 }
16
17 void loop() {
18     if((nLed) < (100)){
19         nLed += 5;
20         mLed += -5;
21         analogWrite(3,nLed);
22         analogWrite(5,mLed);
23         _delay(0.1);
24     }else{
25         nLed = 0;
26         mLed = 100;
27     }
28     _loop();
29 }
30
31 void _delay(float seconds){
32     long endTime = millis() + seconds * 1000;
33     while(millis() < endTime)_loop();
34 }
35
36 void _loop() {
37 }
```

Şekil 8: mBlock Uygulaması

The image displays a software development environment with two windows. The left window, titled 'If_Else_Uygulaması.mbqc', shows a miniBloq block-based programming interface. The right window, titled 'If_Else_Uygulaması.cpp', shows the corresponding C++ code.

miniBloq Blocks (Left Window):

- initBoard** block: Initializes the board.
- nLed** variable block: Set to 0.
- mLed** variable block: Set to 100.
- Loop** block: A yellow block with a question mark and a checkmark, indicating a loop.
- Comparison** block: A black block with a less-than sign (<), comparing nLed to 100.
- Arithmetic** blocks: A blue block with a plus sign (+) adding 5 to nLed, and a black block with a minus sign (-) subtracting 5 from mLed.
- Comparison** block: A black block with an equals sign (=), comparing nLed to 0.
- Comparison** block: A black block with a greater-than sign (>), comparing mLed to 100.
- PWM** blocks: Two blue blocks labeled PWM3 and PWM5, both connected to nLed and mLed variables respectively.
- Delay** block: A yellow block with a clock icon, set to 100.
- Loop** block: A yellow block with a question mark and an 'x', indicating the end of the loop.

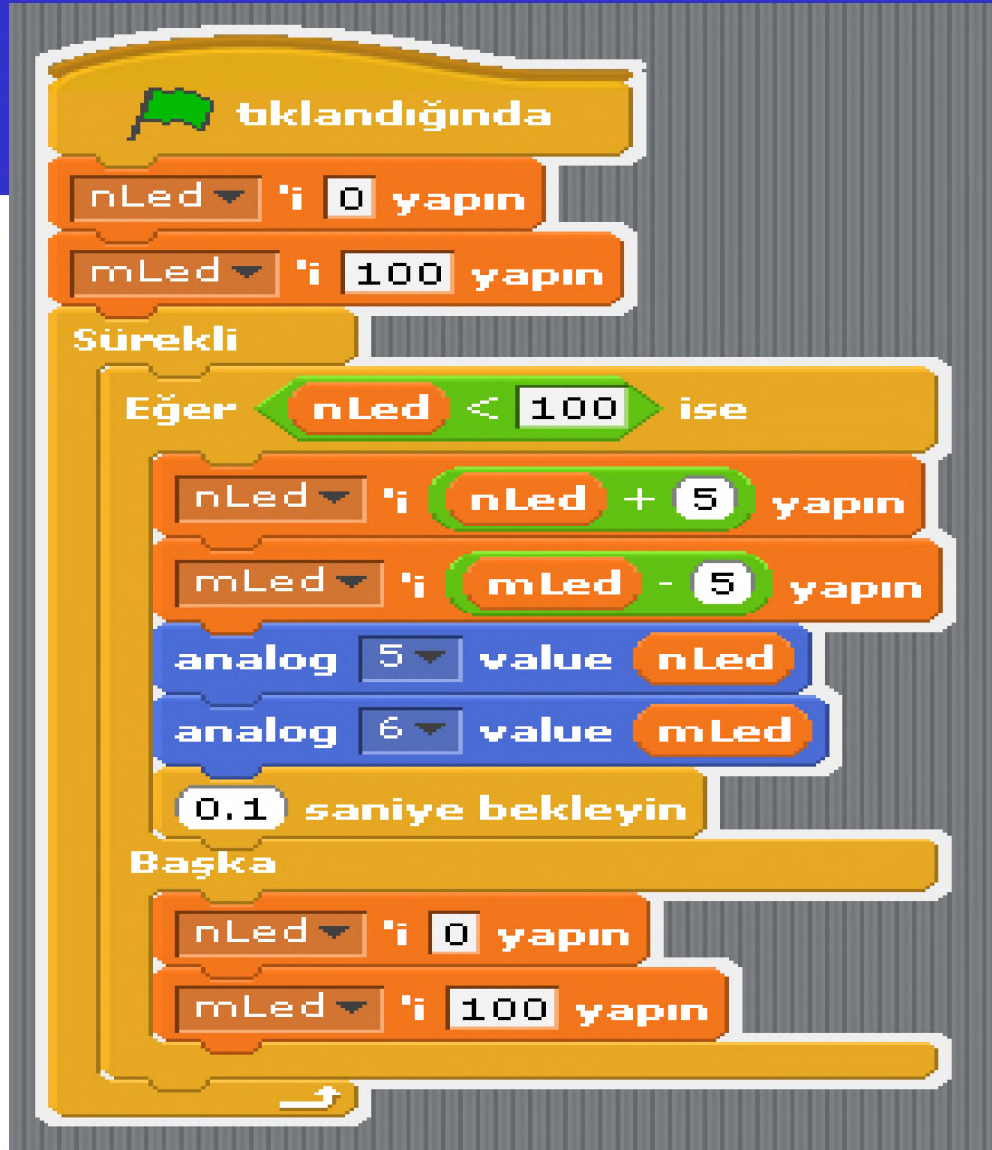
C++ Code (Right Window):

```
#include <mbq.h>
#include <PingIRReceiver.h>

void setup()
{
  initBoard();
  float nLed = 0;
  float mLed = 100;
  while(true)
  {
    if((nLed<100))
    {
      nLed = (nLed+5);
      mLed = (mLed-5);
    }
    else
    {
      nLed = 0;
      mLed = 100;
    }
    AnalogWrite(PWM3, nLed);
    AnalogWrite(PWM5, mLed);
    delay(100);
  }
}

void loop()
{
}
```

Şekil 9: miniBloq Uygulaması



Şekil 10: Scratch for Arduino (SA4)
Uygulamas1



Şekil 11: Snap4Arduino Uygulaması

SONUÇ VE ÖNERİLER

- Burada açıklanan robot programlama ortam ve dillerini genel olarak değerlendirdiğimizde; Ardublock ve miniBloq'un blok yapısı ve tasarımı olarak farklılaştığını, mBlock, Scratch for Arduino (S4A) ve Snap4Arduino'nun Scratch temelli olmasından dolayı oldukça benzer yapıda olduğu ortaya çıkmaktadır.
- Blok tabanlı programlama dersi almamış öğrenciler için miniBloq daha uygun bir seçim olabilirken, ders almış olan öğrenciler için mBlock ve Snap4Arduino daha uygun bir seçim olabilir. İstenirse bu araçların herhangi birisinin Kur 2'de bulunan Robot Programlama dersinde kullanılabileceği düşünülmektedir.

Sıra No	BLOK TABANLI EĞİTSEL ROBOT PROGRAMLAMA ORTAMLARI VE DİLLERİ ÖZELLİKLERİ	ÖZELLİKLERİ				
		ArduBlock	mBlock	miniBloq	Scratch for Arduino	Snap4 Arduino
1	Üzerinde uygulama ve program geliştirmek oldukça kolay ve hızlıdır.	+	+	+	+	+
2	Yaygın kullanılan ve yeni Arduino IDE sürümlerinde çalışmaktadır.	-	+	-	+	+
3	Derse uygun robotik işlem ve uygulamaların gerektirdiği bütün bloklara sahiptir.	-	+	-	+	+
4	Sanal robot kullanımına destek sağlamaktadır.	-	+	-	-	+
5	Kullanılan robotun ve algılayıcıların sahne, dekor ile etkileşiminin sağlanması mümkündür.	-	+	-	+	+
6	Gerekli olabilecek farklı blokların kullanıcı tarafından oluşturmasına olanak sağlanmaktadır.	-	+	-	-	+
7	Hazırlanan uygulama hem blok hem de metin görünümünde gösterebilmektedir.	+	+	+	-	-
8	Geliştirilen uygulama metin tabanlı olarak Arduino IDE ile düzenlenebilmektedir.	+	+	+	-	-
9	Gerçek zamanlı kod üretici ve hata kontrolü içermektedir.	-	-	+	-	-
10	Sözdizimsel hatayı önleme özelliği bulunmaktadır.	-	-	+	-	-
11	Kullanıcının bir bloğa yanlış veri türü parametreleri eklemesi mümkün değildir.	-	-	+	-	-
12	Blok yapısı sezgisel olduğu için kullanılacak bloğun ne yaptığının önceden bilinmesi gerekli değildir.	+	+	-	+	+
13	Geliştirilen uygulamalar Arduino kartına yüklenmeden gerçek zamanlı olarak çalıştırılabilmektedir.	-	+	-	+	+
14	Arduino Board Standartlarını desteklemektedir.	+	+	+	+	+
15	Çok fazla sayıda robotik ve fiziksel donanıma destek sağlamaktadır.	+	Sınırlı	Sınırlı	Sınırlı	Sınırlı
16	Robot hareketi/navigasyonu için hazır bloklara sahiptir.	-	+	-	-	-
17	Farklı firmaların hazır robot ve robotik donanımları kullanılabilir.	+	-	-	-	-
18	Arduino kartların bütün pinlerini kullanmak mümkün olmaktadır.	+	+	+	-	+
19	Birinci sınıf heterojen listeleri, birinci sınıf prosedürleri ve süreklilikleri içermektedir.	-	-	-	-	+
20	Otomatik olarak yapılandırılabilir.	-	-	-	-	+
21	Pin çıkışı sağlamakta ve üst düzey donanım soyutlamaları gerçekleştirilebilmektedir.	-	-	-	-	+
22	Çalışması için Arduino kartına özel bir uygulamanın yüklenmesi gerekmektedir.	-	-	-	+	+
23	Uzaktan kontrol ve gerçek zamanlı veri akışı için HTTP protokol desteği bulunmaktadır.	-	-	-	-	+
24	Kablosuz programlama desteği bulunmaktadır.	+	+	-	+	+
25	Kullanıcı yorumları bloklar arasına eklenebilmektedir.	-	+	+	+	+
26	Sürekli olarak güncelleştirilmektedir.	-	+	-	-	-
27	Microsoft Windows, Mac OS ve Linux işletim sistemleri üzerinde çalışmaktadır.	+	+	+	+	+
28	Web tabanlı bir sürümü bulunmaktadır.	-	+	-	-	+
29	Taşınabilir yapıdadır, kurulum gerektirmemektedir.	-	-	+	-	-
30	Türkçe dil desteği bulunmamaktadır.	+	+	-	+	+

ROBOT PROGRAMLAMA ÖĞRETİMİNDE ARDUİNO IDE KULLANIMI



Mustafa NUMANOĞLU

Robot Programlama Öğretiminde Arduino IDE Kullanımı

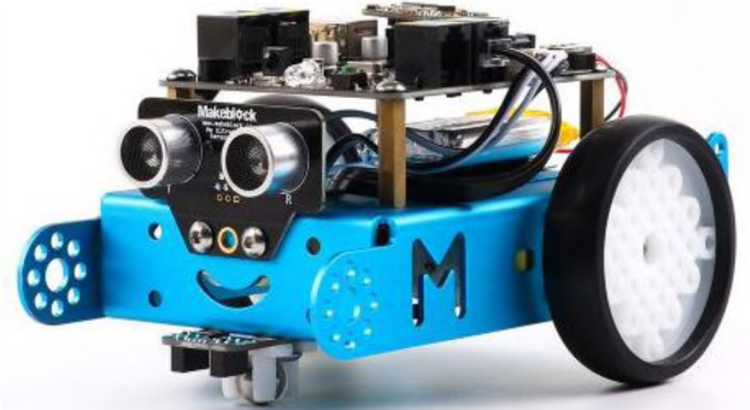
- Bu çalışmanın amacı; ortaöğretim bilgisayar bilimi dersinin robot programlama ünitesinde kullanılabilecek metin tabanlı eğitsel robot programlama ortam ve dillerden biri olarak Arduino IDE'yi uygulama örnekleriyle tanıtmaktır.
- Bu amaçla çalışmanın ilk aşamasında Ortaöğretim Bilgisayar Bilimi Dersi Robot Programlama Ünitesi (Kur 2) Öğretim Programı incelenmiştir.
- Programdan beklenen öğrenme çıktıları ve alan yazındaki çalışmalar incelenerek bu dersin öğrenme-öğretme sürecinde kullanılıp kullanılmayacağı belirlenmeye çalışılmıştır.

Robot Programlama Öğretiminde Arduino IDE Kullanımı

- Bu çalışma yapılırken geliştirilmeye açık bir sisteme sahip olması, açık kaynak donanım ve yazılım desteği sağlanması, tümleşik bir geliştirme ortamı olması, tercihen Türkçe dil desteğinin bulunması, tercih edilebilecek farklı programlama donanımları için kullanılabilmesi gibi ayrıntılar da dikkate alınmıştır.
- İncelemeler sonucunda Arduino IDE'nin bu ders için kullanılacak metin tabanlı eğitsel robot programlama ortam ve dillerinden biri olabileceği tespit edilmiştir.
- Bütünleşik geliştirme ortamının kısa tanıtımı yapılmış, çalışmanın sonraki aşamasında bu ortam ve dilin programlama yapısı ve kullanım şekilleri geliştirilen uygulama programları ile gösterilmiştir.

Robot Programlama Öğretiminde Arduino IDE Kullanımı

- Bu amaçla robot programlama çalışmalarında temel uygulamalardan olan döngü, koşul yapıları, fonksiyon, değişken ve dizi yapısını öğretmeyi amaçlayan örnek programlar hazırlanmıştır.
- Hazırlanan bu programlar mBot eğitsel robot üzerinde denenmiştir.



Arduino IDE (Tümleşik Geliştirme Ortamı- Integrated Development Environment)

- Arduino IDE; tümleşik bir derleyici, yorumlayıcı ve hata ayıklayıcı olarak görev yapan, aynı zamanda derlenen programı karta yükleme işlemini de yapabilen, her platformda çalışabilen Java programlama dilinde yazılmış bir uygulamadır.
- Ortam; Arduino Bootloader (Optiboot), Arduino kütüphaneleri, AVRDUde (Arduino üzerindeki mikrodenetleyiciyi programlayan, derlenen kodları programlamak için kullanılan yazılım) ve derleyiciden (AVR-GCC) oluşmaktadır.
- Bu araçlar yazılımın derlenmesi, bağlanması, çalışmaya tümüyle hazır hale gelmesi ve daha birçok ek işi otomatik olarak yapabilmek amacıyla kullanılmaktadır.

Arduino IDE (Tümleşik Geliştirme Ortamı- Integrated Development Environment)

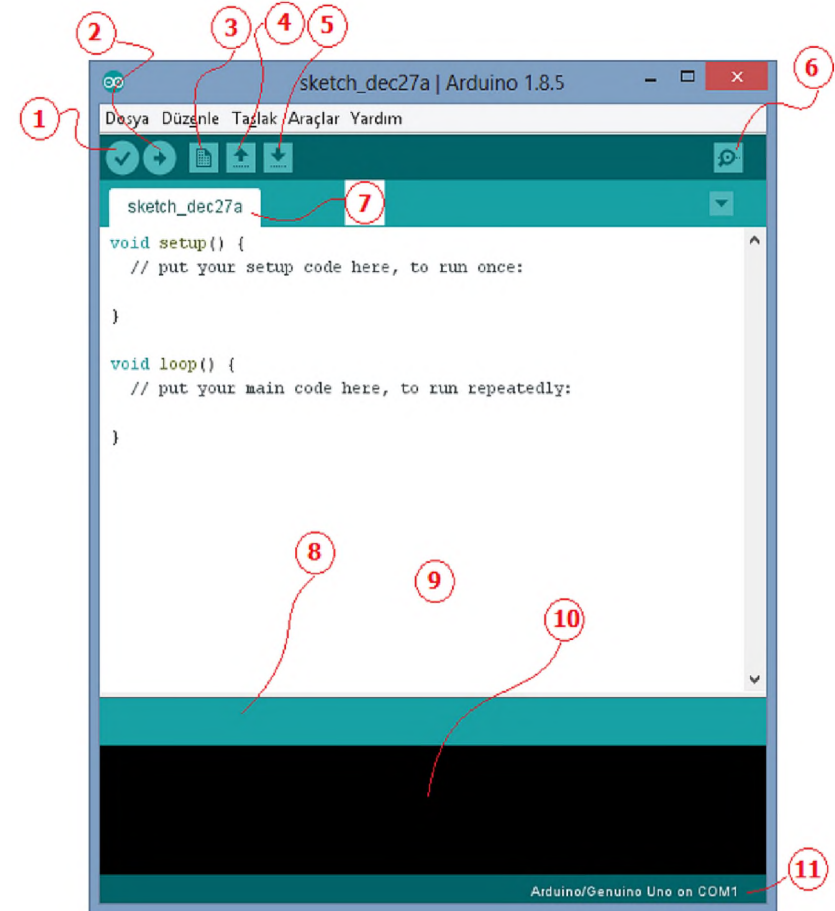
- Arduino tümleşik geliştirme ortamı (IDE) mikrodenetleyici ve elektronik konusunda detaylı bilgi sahibi olmayı gerektirmeden herkesin programlama yapabilmesini sağlayan kütüphanelerden oluşmaktadır.
- Arduino kütüphaneleri, geliştirme ortamı ile birlikte gelmekte ve "libraries" klasörünün altında bulunmaktadır.
- IDE, Java dilinde yazılmıştır ve Processing adlı dilin ortamına dayanmaktadır.
- Arduino C++ dili ile kolayca programlanabilmektedir.
- Kütüphaneler yardımıyla uygulama geliştirilmesi de oldukça kolay ve hızlıdır.

Arduino IDE (Tümleşik Geliştirme Ortamı- Integrated Development Environment)

- Arduino IDE yazılımının son versiyonu <https://www.arduino.cc/en/main/software> sitesinden, kullanılacak işletim sistemi seçilerek ücretsiz olarak indirilebilmektedir.
- Çalışmada Arduino IDE'nin 1.8.5 sürümü Microsoft Windows 10 Pro işletim sistemi üzerinde kullanılmıştır.
- mBot eğitsel robot üzerinde yapılan denemeler için gerekli olan kütüphaneler <https://github.com/Makeblock-official/Makeblock-Libraries> adresinden indirilmiştir. Makeblock Library v3.24 sürümü kullanılmıştır.

Arduino IDE Yapısı ve Genel Özellikleri

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.



Arduino IDE Yapısı ve Genel Özellikleri

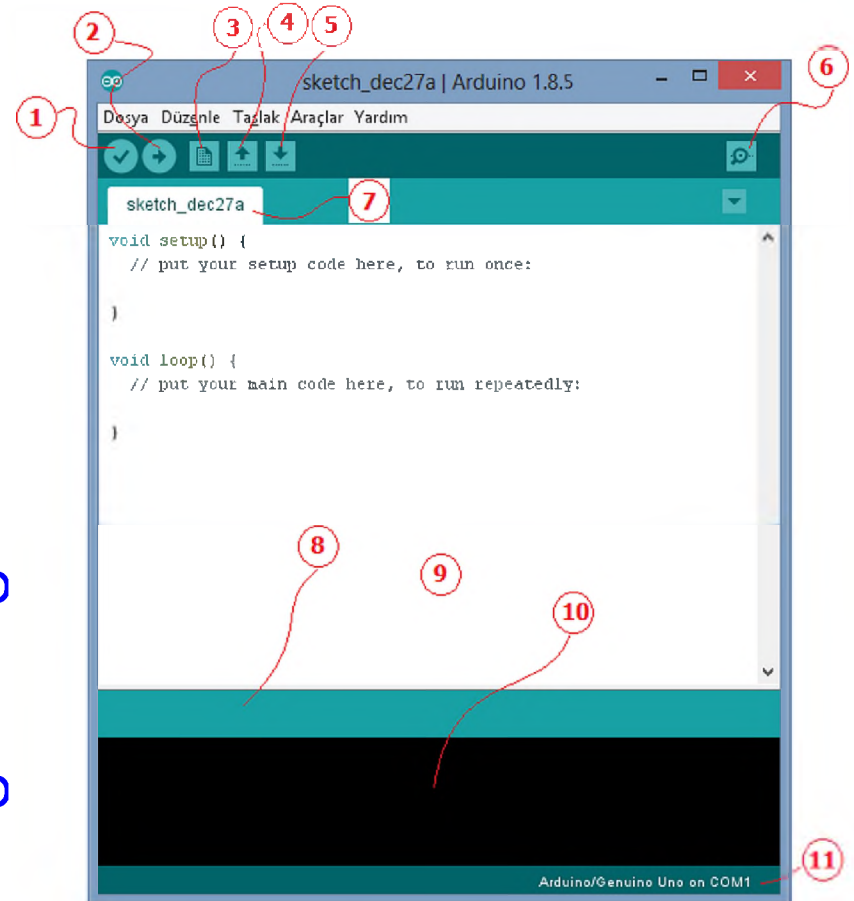
7. Sketch: Yazılan programın dosya ismini gösterir.

8. Gösterge: Yaptığı işlemin ilerleme durumunu gösterir.

9. Boş alan: Yazılacak program alanıdır.

10. Rapor: Varsa derleme sonucu yapılan hataları, yoksa programın yükleme sonrası mikro denetleyicide kapladığı alanı gösterir.

11. Gösterge: Bağlı olan Arduino modeli ve USB portu gösterir.



Arduino IDE'de Kullanılan Yapılar, Değişkenler ve Fonksiyonlar

A. SÖZ DİZİMİ VE OPERATÖRLER		B. DEĞİŞKENLER
1. Söz Dizimi ; Noktalı Virgül { Süslü Parantez // Çift Slash /**/ Yıldızlı Slash #define #include	5. İşaretçi Operatörler * Referans dışı operatör & Referans operatörü	1. Sabitler HIGH LOW INPUT OUTPUT INPUT_PULLUP LED_BUILTIN true false integer constants floating point constants
2. Aritmetik Operatörler "= Atama İşleci + Toplama - Çıkarma * Çarpma / Bölme % Modulo	6. Bitisel Operatörler & (Bitisel Ve) (Bitisel Veya) ^ (Bitisel Xor) ~ (Bitisel Değil) << (Bitshift Sol) >> (Bitshift Sağ)	2. Veri Tipleri void boolean char unsigned char byte int unsigned int word long unsigned long short float double string - char array substring - object array
3. Karşılaştırma Operatörleri == (eşit eşit) != (eşit değil) < (küçük) > (büyük) <= (küçük eşit) >= (büyük eşit)	7. Birleşik Operatörler ++ (arttırma) -- (azaltma) += (Birleşik arttırma) -= (Birleşik çıkarma) *= (Birleşik çarpma) /= (Birleşik bölme) %= (Birleşik mod) &= (Bitisel Lojik Ve) = (Bitisel Lojik Veya)	3. Dönüşümler char() byte() int() word() long() float()
4. Boolean Operatörleri && (ve) (veya) ! (değil)		4. Değişken Kapsamları static volatile const
		5. Yardımcılar PROGMEM sizeof()

Arduino IDE'de Kullanılan Yapılar, Değişkenler ve Fonksiyonlar

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

Hazırlanan Döngü Örnekleri

- İşlemi Sürekli Tekrarlayan Döngü Örneği: Bu örnekte mBot robot hızını 100 yaparak 1 saniye boyunca ileri, 1 saniye boyunca da geri hareket ettikten sonra durmaktadır. Bu işlemi sürekli olarak tekrarlamaktadır.

```
1 /* Döngü Örneği (Sürekli Tekrarla)*/
2 #include "MeMCore.h"
3
4 MeDCMotor motor1(M1);
5 MeDCMotor motor2(M2);
6
7 int motorHiz = 100;
8
9 void setup() {
10 }
11
12 void loop() {
13     motor1.run(-motorHiz);
14     motor2.run(motorHiz);
15     delay(1000);
16
17     motor1.stop();
18     motor2.stop();
19     delay(1000);
20
21     motor1.run(motorHiz);
22     motor2.run(-motorHiz);
23     delay(1000);
24
25     motor1.stop();
26     motor2.stop();
27     delay(1000);
28 }
```

Hazırlanan Döngü Örnekleri

- **Verilen Sayı Kadar İşlemi Tekrarlayan Döngü**
Örneği: Bu örnekte robot hızını 100 yaparak 1 saniye boyunca ileri, 1 saniye boyunca da geri hareket etmektedir. Toplamda bunu 2 defa tekrarlamaktadır. Tekrarın sonunda hızını sıfırlayarak durmaktadır.

```
1 /* Döngü Örneği (Verilen Sayı Kadar Tekrarla)*/
2 #include "MeMCore.h"
3
4 MeDCMotor motor1(M1);
5 MeDCMotor motor2(M2);
6
7 int motorHiz = 100;
8 int donguSayisi = 2;
9
10 void setup() {
11     for (int i = 0; i < donguSayisi; ++i)
12     {
13         motor1.run(-motorHiz);
14         motor2.run(motorHiz);
15         delay(1000);
16
17         motor1.stop();
18         motor2.stop();
19         delay(1000);
20
21         motor1.run(motorHiz);
22         motor2.run(-motorHiz);
23         delay(1000);
24
25         motor1.stop();
26         motor2.stop();
27         delay(1000);
28     }
29 }
30 void loop() {
31 }
```

Hazırlanan Koşul Örnekleri

- **Olumlu Koşul İfadesi “Eğer” “İse”**
Örneği: Bu örnekte robotun ultrasonik algılayıcısı kullanılarak önündeki engele olan uzaklık ölçülmektedir. Engele olan uzaklık 20 cm’den büyük ise robot hareket etmemektedir. Eğer engele olan uzaklık 20 cm’den küçük ise koşul gerçekleşecek ve hızını 100 yaparak 1 saniye boyunca engelden geriye doğru giderek uzaklaşacaktır. 1 saniye sonra M1 ve M2 motorlarının hızlarını sıfıra düşürerek duracaktır. Engele olan uzaklığı tekrar ölçecek yine 20 cm’den küçük ise hızını 100 yaparak 1 saniye boyunca engelden geriye doğru giderek uzaklaşacaktır. Bu durum engele olan uzaklık 20 cm’den büyük oluncaya kadar devam edecektir.

```
1 /* Koşul Örneği (İse)*/
2 #include "MeMCore.h"
3
4 MeDCMotor motor1(M1);
5 MeDCMotor motor2(M2);
6 MeUltrasonicSensor ultrasonic(3);
7
8 int motorHiz = 100;
9
10 void setup() {
11     Serial.begin(9600);
12 }
13
14 void loop() {
15     if ({ultrasonic.distanceCm()} < {20}) {
16         Serial.print("Mesafe : ");
17         Serial.print(ultrasonic.distanceCm());
18         Serial.println(" cm");
19         delay(100);
20
21         motor1.run {motorHiz};
22         motor2.run {-motorHiz};
23         delay(1000);
24
25         motor1.stop();
26         motor2.stop();
27         delay(1000);
28     }
29 }
```

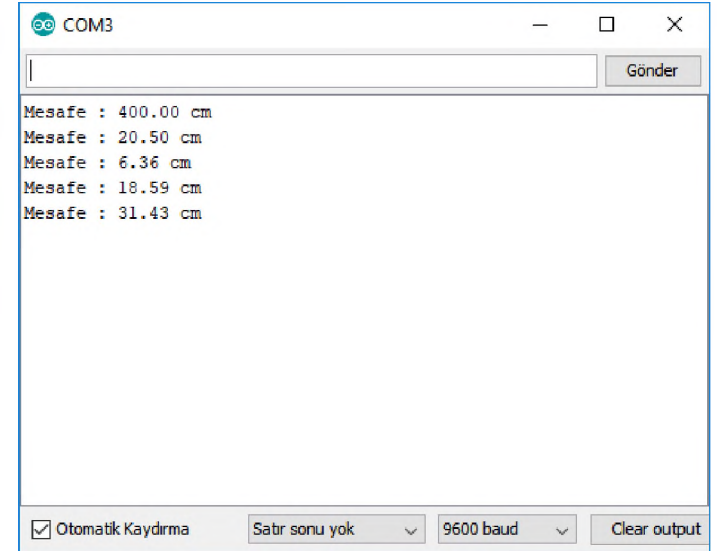
Hazırlanan Koşul Örnekleri

- **Olumlu Koşul İfadesi “Eğer” “İse” ve Olumsuz Koşul İfadesi “Değilse” Örneği:**
Bu örnekte de robotun ultrasonik algılayıcısı kullanılarak önündeki engele olan uzaklık ölçülmektedir. Eğer engele olan uzaklık 20 cm’den büyük ise hızını 100 yaparak 1 saniye boyunca engele doğru ilerleyecektir. 1 saniye sonra M1 ve M2 motorlarının hızlarını sıfıra düşürerek duracaktır. Engel olan uzaklığı tekrar ölçecek yine 20 cm’den büyük ise hızını 100 yaparak 1 saniye boyunca engele doğru ilerleyecektir. Bu ilerleyip durma durumu engele olan uzaklık 20 cm’den küçük olana kadar devam edecektir.

```
1 /* Koşul Örneği (Değilse) */
2 #include "MeMCore.h"
3
4 MeDCMotor motor1(M1);
5 MeDCMotor motor2(M2);
6 MeUltrasonicSensor ultrasonic(3);
7
8 int motorHiz = 100;
9
10 void setup() {
11     Serial.begin(9600);
12 }
13
14 void loop() {
15     if (ultrasonic.distanceCm() > {20}) {
16         Serial.print("Mesafe : ");
17         Serial.print(ultrasonic.distanceCm());
18         Serial.println(" cm");
19         delay(100);
20
21         motor1.run(-motorHiz);
22         motor2.run(motorHiz);
23         delay(1000);
24
25         motor1.stop();
26         motor2.stop();
27         delay(1000);
28     }
29
30     else {
31         Serial.print("Mesafe : ");
32         Serial.print(ultrasonic.distanceCm());
33         Serial.println(" cm");
34         delay(100);
35
36         motor1.run(motorHiz);
37         motor2.run(-motorHiz);
38         delay(1000);
39
40         motor1.stop();
41         motor2.stop();
42         delay(1000);
43     }
44 }
```

Hazırlanan Koşul Örnekleri

- **Olumlu Koşul İfadesi “Eğer” “İse” ve Olumsuz Koşul İfadesi “Değilse” Örneği:** Eğer engele olan uzaklık 20 cm’den küçük ise hızını 100 yaparak 1 saniye boyunca engelden geriye doğru giderek uzaklaşacaktır. 1 saniye sonra M1 ve M2 motorlarının hızlarını sıfıra düşürerek duracaktır. Engel olan uzaklığı tekrar ölçecek yine 20 cm’den küçük ise hızını 100 yaparak 1 saniye boyunca engelden geriye doğru giderek uzaklaşacaktır. Bu durum engele olan uzaklık 20 cm’den büyük oluncaya kadar devam edecektir. Uzaklık bilgisi seri port ekranına yazılmaktadır.



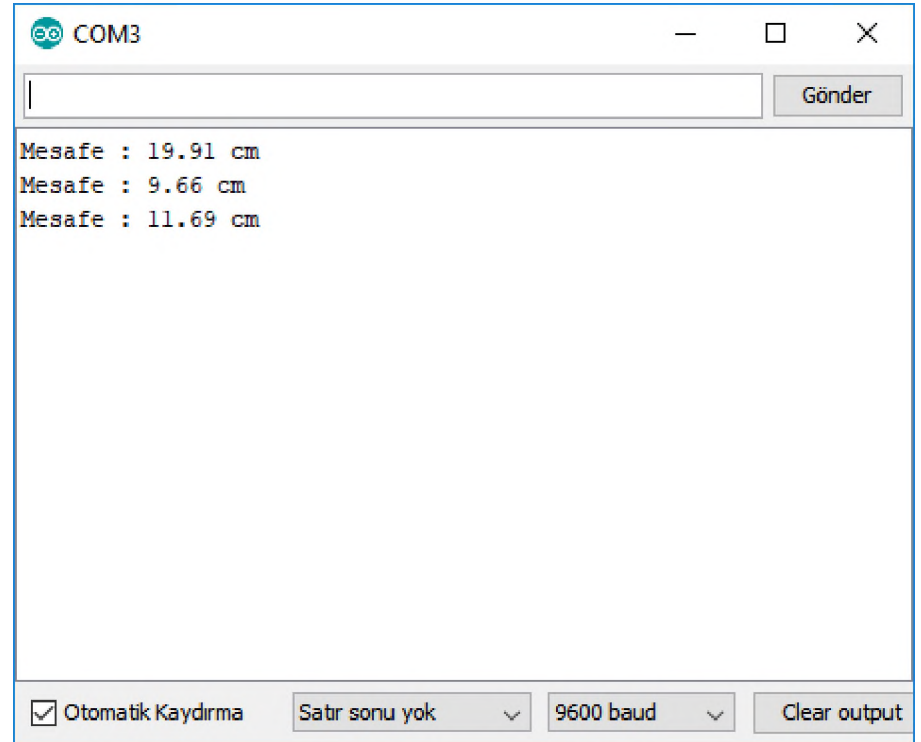
Hazırlanan Koşul Örnekleri

- **Belirli Bir Şart Gerçekleşene Kadar Bekleme Örneği:** Bu örnekte robotun ultrasonik algılayıcısı kullanılarak önündeki engele olan uzaklık ölçülmektedir. Engele olan uzaklık 20 cm'den büyük ise robot beklemeye geçmektedir. Eğer engele olan uzaklık 20 cm'den küçük ise (örneğin elinizi robotun ultrasonik algılayıcısına yaklaştırırsanız) koşul gerçekleşecek ve hızını 100 yaparak 1 saniye boyunca engele doğru ilerleyecektir. 1 saniye sonra M1 ve M2 motorlarının hızlarını sıfıra düşürerek duracaktır.

```
1 /* Koşul Örneği (Şart Gerçekleşene Kadar Bekle)*/
2 #include "MeMCore.h"
3
4 MeDCMotor motor1(M1);
5 MeDCMotor motor2(M2);
6 MeUltrasonicSensor ultrasonic(3);
7
8 int motorHiz = 100;
9
10 void setup() {
11     Serial.begin(9600);
12 }
13
14 void loop() {
15     while (!(ultrasonic.distanceCm() < {20}));
16     Serial.print("Mesafe : ");
17     Serial.print(ultrasonic.distanceCm());
18     Serial.println(" cm");
19     delay(100);
20
21     motor1.run(-motorHiz);
22     motor2.run(motorHiz);
23     delay(1000);
24
25     motor1.stop();
26     motor2.stop();
27     delay(1000);
28 }
```

Hazırlanan Koşul Örnekleri

- **Belirli Bir Şart Gerçekleşene Kadar Bekleme Örneği:** Engel olan uzaklığı tekrar ölçecek 20 cm'den küçük olduğu için hızını 100 yaparak 1 saniye boyunca yine engele doğru ilerleyecektir. Bu ilerleyip durma durumu engele olan uzaklık sıfırlanıncaya kadar devam edecektir.



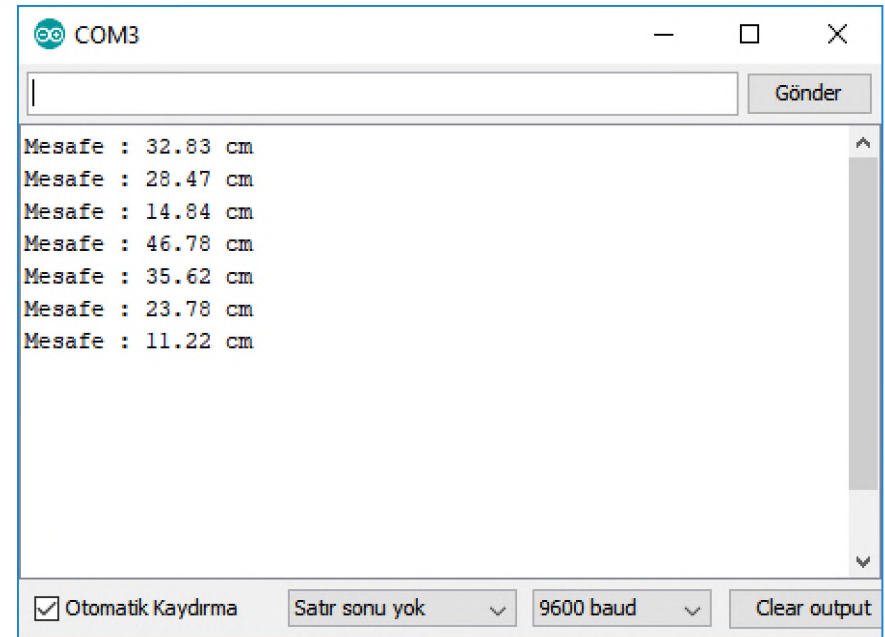
Hazırlanan Koşul Örnekleri

- **Belirli Bir Şart Gerçekleşene Kadar Döngü Örneği:** Bu örnekte robotun ultrasonik algılayıcısı kullanılarak önündeki engele olan uzaklık ölçülmektedir. Engele olan uzaklık 20 cm'den küçük ise robot beklemeye geçmektedir. Eğer engele olan uzaklık 20 cm'den büyük ise koşul gerçekleşecek ve hızını 100 yaparak 1 saniye boyunca engele doğru ilerleyecektir. 1 saniye sonra M1 ve M2 motorlarının hızlarını sıfıra düşürerek duracaktır.

```
1 /* Koşul Örneği (Şart Gerçekleşene Kadar Tekrarla) */
2 #include "MeMCore.h"
3
4 MeDCMotor motor1(M1);
5 MeDCMotor motor2(M2);
6 MeUltrasonicSensor ultrasonic(3);
7
8 int motorHiz = 100;
9
10 void setup() {
11     Serial.begin(9600);
12 }
13
14 void loop() {
15     while (!(ultrasonic.distanceCm() < {20}))
16     {
17         motor1.run(-motorHiz);
18         motor2.run(motorHiz);
19         delay(1000);
20
21         motor1.stop();
22         motor2.stop();
23         delay(1000);
24
25         Serial.print("Mesafe : ");
26         Serial.print(ultrasonic.distanceCm());
27         Serial.println(" cm");
28         delay(100);
29     }
30 }
```

Hazırlanan Koşul Örnekleri

- **Belirli Bir Şart Gerçekleşene Kadar Döngü Örneği:** Engel olan uzaklığı tekrar ölçecek 20 cm'den büyük ise hızını 100 yaparak 1 saniye boyunca yine engele doğru ilerleyecektir. Bu ilerleyip durma durumu engele olan uzaklık 20 cm'den küçük olana kadar devam edecektir. 20 cm'den küçük olunca robot artık hareket etmeyecektir.



Hazırlanan Prosedür Örneği

- Kodu bir kez yazıp defalarca kullanmak için ortaya konmuş temel yapılardır. Program akışı içinde tekrarlayan ifadelerin her seferinde tekrar tekrar yazılması yerine, bir kere yazılıp tekrarlanan her yerde kullanmak için uygundur. Örnekte “Dur” adında bir prosedür oluşturulma işlemi gösterilmiştir. Burada bu prosedürün oluşturulmasıyla yapılmak istenilen bir işlem grubunun tek bir komut olarak belirlenip her seferinde aynı komutların ayrı ayrı kullanılmasını ortadan kaldırmaktır.

```
1 /* Prosedür Örneği 1 */
2 #include "MeMCore.h"
3
4 MeDCMotor motor1(M1);
5 MeDCMotor motor2(M2);
6 MeUltrasonicSensor ultrasonic(3);
7
8 int motorHiz = 100;
9
10 void Dur();
11 void Dur() {
12     motor1.stop();
13     motor2.stop();
14 }
15
16 void setup() {
17     Serial.begin(9600);
18     if ({ultrasonic.distanceCm()} > {20}) {
19         motor1.run(-motorHiz);
20         motor2.run(motorHiz);
21         delay(1000);
22         Dur ();
23     }
24 }
25 void loop() {
26     Serial.print("Mesafe : ");
27     Serial.print(ultrasonic.distanceCm());
28     Serial.println(" cm");
29     delay(100);
30 }
```

Hazırlanan Dizi Örneği

- Diziler çok sayıda değişkenle çalışmak için oluşturulmuş temel yapılarından biridir. Değişkenlerden farklı olarak köşeli parantez içinde virgülle ayrılmış birden fazla değer taşırlar. Örnekte “koşul” adında bir dizinin oluşturulması işlemi gösterilmiştir.

```
1 /* Dizi Örneği 2 */
2 #include "MeMCore.h"
3
4 MeDCMotor motor1(M1);
5 MeDCMotor motor2(M2);
6
7 int motorHiz = 100;
8 char* kosul[5] = {"İleri git", "Geri git", "Saga don", "Sola don"};
9
10 void Dur();
11 void Dur() {
12     motor1.stop();
13     motor2.stop();
14 }
15
16 void setup() {
17 }
18
19 void loop() {
20     for (int m = 0; m < KosulSayisi; m++) {
21         if (m == (1)) {
22             motor1.run(-motorHiz);
23             motor2.run(motorHiz);
24             delay(1000);
25             Dur();
26         }
27         if (m == (2)) {
28             motor1.run(motorHiz);
29             motor2.run(-motorHiz);
30             delay(1000);
31             Dur();
32         }
33         if (m == (3)) {
34             motor1.run(-0);
35             motor2.run(motorHiz);
36             delay(1000);
37             Dur();
38         }
39         if (m == (4)) {
40             motor1.run(-motorHiz);
41             motor2.run(-0);
42             delay(1000);
43             Dur();
44         }
45     }
46 }
47 }
```

SONUÇ VE ÖNERİLER

- Elde edilen bulgular Arduino IDE'nin robot programlama öğretiminde kolayca kullanılabilceğini; döngüler, koşul yapıları, fonksiyonlar ve diziler gibi programlamanın temel kavramlarını içeren uygulamaların hızlı ve kolayca oluşturulup test edilebileceğini göstermektedir.
- İstenirse Arduino IDE'nin Kur 2'de bulunan Robot Programlama dersinde kolaylıkla kullanılabilceği düşünülmektedir.