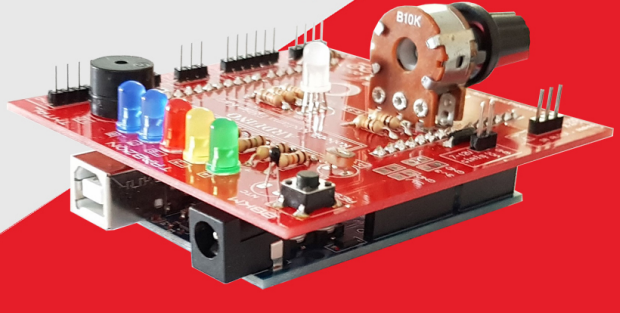
BİLGİSAYAR BİLİMLERİ KODLAMA EĞİTİM MERKEZİ

TRABZON

MESLEKİ VE TEKNİK ANADOLU LİSESİ



ROBOTİK KODLAMA

EĞİTİM SETİ UYGULAMA KLAVUZU

* TEMEL KONU ANLATIM
* BLOK VE METİN TABANLI 100 UYGULAMA ÖRNEĞİ

TRABZON BİLGİSAYAR BİLİMLERİ VE KODLAMA EĞİTİMİ MERKEZİ

TRABZON MESLEKİ VE TEKNİK ANADOLU LİSESİ

ARDUINO ROBOTİK KODLAMA EĞİTİM KARTI UYGULAMA KLAVUZU

HAZIRLAYANLAR

Emre UZUN
(Uzman Teknik Öğretmen - Bilişim Teknolojileri Alanı)

Mustafa GÜNEŞ
(Uzman Teknik Öğretmen - Elektrik-Elektronik Teknolojileri Alanı)

EYLÜL - 2023

İÇİNDEKİLER

A. ARDUINO TANITIM

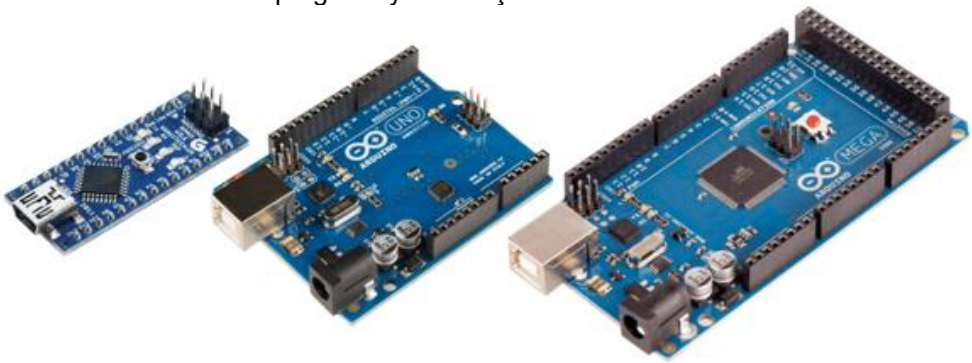
A.1. Genel Tanıtım

ARDUINO, üzerinde **OSİLATÖR**, **REGÜLATÖR**, **RESET**, **USB PROGRAMLAMA PORTU** ve **GİRİŞ/ÇIKIŞ BİRİMLERİ** bulunan bir Mikrodenetleyici geliştirme kartıdır.

ARDUINO, kullanıcıyı elektronik devre ile fazla uğraştırmadan kontrol yeteneği, kolay programlanabilmesi ve bolca kaynak bulunabilmesi sebebiyle günümüzde çok tercih edilir.

ARDUINO ÜRETİCİLERİ, ürettikleri arduinoların devre şemalarını paylaşıma açmıştır. Bu sayede ARDUINO'nun klonları üretilmiştir. Fakat programlanması ve kullanımı aynıdır. **Bu sebeple internette çok farklı fiyatlarda ARDUINO'lar bulabilirsiniz.**

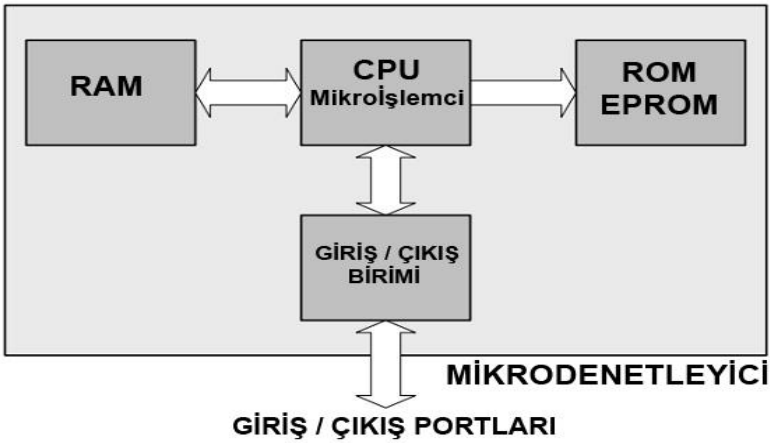
ARDUINO kartlarının donanımında bir adet Atmel AVR mikrodenetleyici (ATmega328, ATmega2560 gibi) ve programlama ile diğer devrelere bağlantı için gerekli yan elemanlar bulunur. Her Arduino kartında en azından bir 5 voltluk regüle ünitesi ve bir 16MHz osilator vardır. Arduino kartlarında programlama için harici bir programlayıcıya ihtiyaç duyulmaz, çünkü karttaki mikrodenetleyiciye önceden bir bootloader programı yüklenmiştir.



ARDUINO kütüphaneleri ile kolaylıkla programlama yapılabilir, analog ve digital sinyalleri işleyebilir, algılayıcılardan gelen sinyalleri kullanarak, çevre birimleri ile etkileşim içerisinde olan robotlar ve sistemler tasarlanabilmektedir. Tasarlanan projeye özgü olarak dış dünyaya hareket, ses, ışık gibi tepkiler oluşturulabilmektedir.

ARDUINO nun farklı ihtiyaçlara çözüm üretebilmek için tasarlanmış çeşitli eklentileri ve modülleri mevcuttur. Bu modüller kullanılarak proje geliştirilebilmektedir.

A.2. İç Yapısı ve Bileşenleri

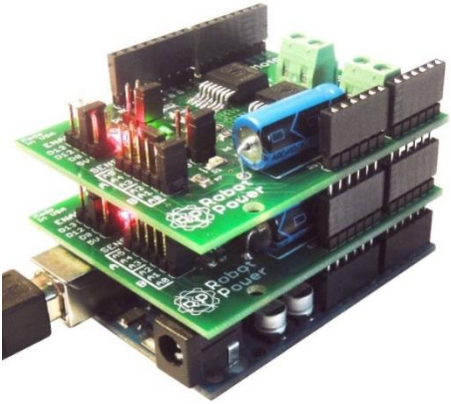


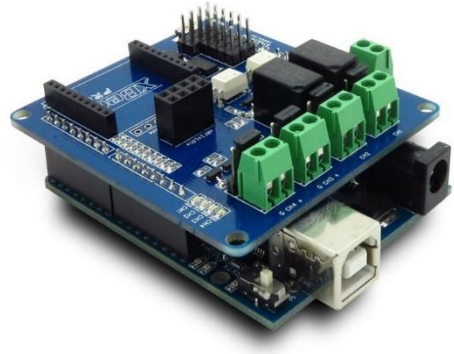
Bu noktada, **Mikroişlemci ve Mikrodenetleyici** kavramlarını detaylandırmak gerekmektedir. Mikroişlemci ile Mikrodenetleyici birbirine karıştırılmamalıdır. Mikrodenetleyiciler, yapılarında Mikroişlemcileri içermektedirler. Mikroişlemcilerin yapısında CPU, ön bellek ve I/O portları olmasına karşın, Mikrodenetleyicilerde ayrıca seri ve paralel portlar, sayıcılar ve çeviriciler bulunmaktadır. EEPROM, RAM ve ROM birimleri, Mikrodenetleyicilerde dahili olasına karşın, Mikroişlemcilerde harici birimlerdir. Mikrodenetleyiciler, gerçek zamanlı uygulamalarda daha başarılı çalışmaktadırlar. Mikrodenetleyiciler çok küçük boyutlarda ve daha az enerji harcayarak çalışabilmektedirler. Bunun yanısıra,Mikroişlemciler aynı anda çoklu işlem yapabilirken,Mikrodenetleyiciler aynı anda tek bir işlem yapabilmektedirler.

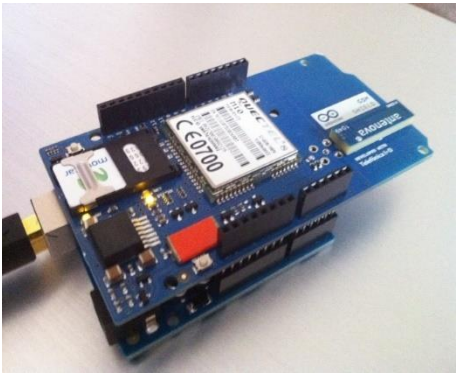
A.3. Genişletme Elemanları

ARDUINO da farklı işlevler için üretilmiş özel geliştirme kartları bulunmaktadır. Bu kartlara **SHIELD** adı verilir. **ARDUINO** ile çalışabilen ve Arduino'yu geliştirmeyi sağlayan ve üzerine takılabilen parçalardır. Diğer bir adıyla modül olarak da bilinir.

ARDUINO'yu geliştirmek ve yeni donanımlar ekleyebilmek için bir adet SHIELD kullanılabileceği gibi, birden çok SHIELD de kullanılabilir. Birden fazla SHIELD kullanılırken, SHIELD lerin pinlerine dikkat etmek gerekmektedir. SHIELD lerin Arduino üzerindeki aynı pini kullanmamasına özen gösterilmelidir. Böyle bir durum varsa, SHIELD ler birlikte kullanılmamalıdır. Aşağıda bazı örnek SHIELD kullanımları gösterilmiştir.







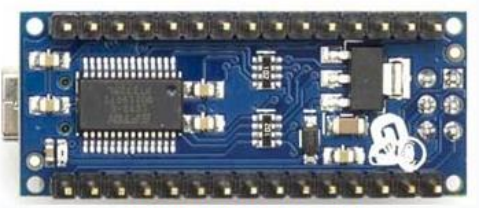
A.4. Çeşitleri ve Özellikleri

ARDUINO kartlarının birçok çeşidi mevcuttur. Temel olarak bütün kartlar benzer bileşenlere sahip olmakla beraber, Mikrodenetleyici modelleri, Giriş/Çıkış pinleri ve dahili modüllerin sayısı, ebat ve çalışma gerilimleri gibi farklılıklara sahiptirler. En çok bilinen ve kullanılan çeşitleri aşağıda belirtilmiştir.

1. Arduino NANO

Arduino NANO, ATmega328 mikrodenetleyici (Nano 3.x) veya Atmega168 (Nano 2.x) mikrodenetleyici barındıran, küçük ve breadboard dostu bir kartdır.

Arduino Nano V3 ön yüz

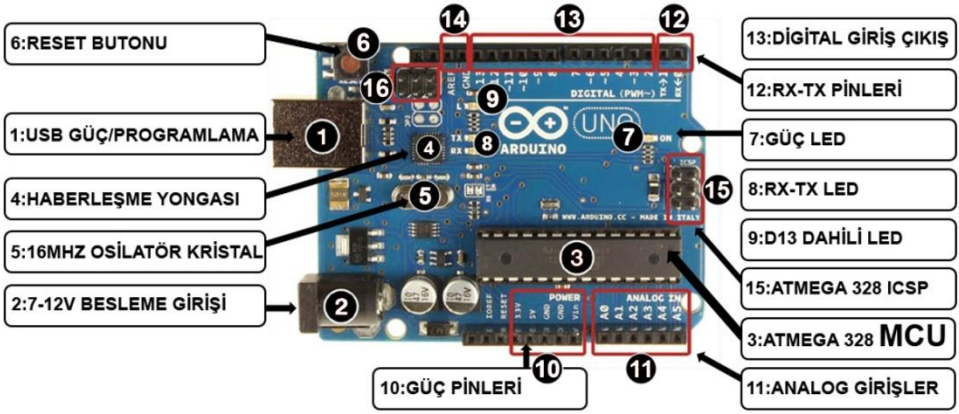

Arduino Nano V3 arka yüz

Teknik Özellikleri

Mikrodenetleyici	Arduino Nano V3 Te Atmega328 Önceki Versiyonlarda Atmega168
Çalışma Gerilimi	+5 V DC
Tavsiye Edilen Besleme Gerilimi	7 - 12 V DC
Dijital Giriş / Çıkış Pinleri	14 ADET/ 6 ADET PWM Çıkış
Analog Giriş Pinleri	8 ADET
Giriş / Çıkış Pini Başına Düşen DC Akım	40 mA
Flash Hafıza	Atmega328 32KB-Atmega168 16KB
SRAM	Atmega328 2KB-Atmega168 1KB
EEPROM	Atmega328 1KB-Atmega168 512B
Saat Frekansı	16 MHz
Boyutları	18 mm X 45 mm
Ağırlık	5 Gram

2. Arduino UNO

Arduino UNO, ATmega328 mikrodenetleyici içeren bir kartdır. Arduino'nun en yaygın kullanılan kartı olduğu söylenebilir. Arduino UNO, ilk modelinden sonra, Arduino UNO R2, Arduino UNO SMD ve son olarak Arduino UNO R3 çıkmıştır. Arduino'nun kardeş markası olan Genuino markasını taşıyan Genuino UNO kartı ile tamamen aynı özelliklere sahiptir.



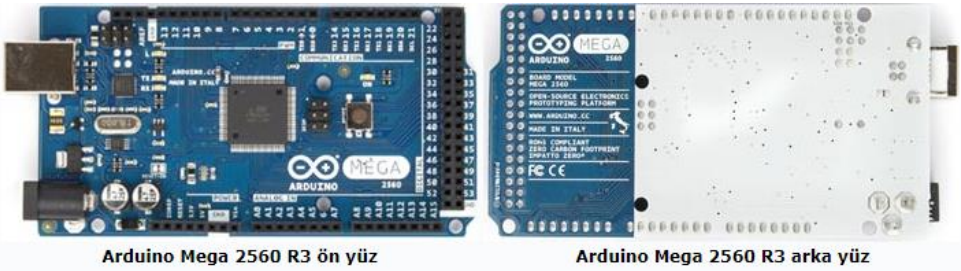
Teknik Özellikleri

Mikrodenetleyici	Atmega328
Çalışma Gerilimi	+5 V DC
Tavsiye Edilen Besleme Gerilimi	7 - 12 V DC
Dijital Giriş / Çıkış Pinleri	14 ADET/ 6 ADET PWM Çıkış
Analog Giriş Pinleri	6ADET
Giriş / Çıkış Pini Başına Düşen DC Akım	40 mA
Flash Hafıza	Atmega328 32KB
SRAM	Atmega328 2KB
EEPROM	Atmega328 1KB
Saat Frekansı	16 MHz

3. Arduino MEGA

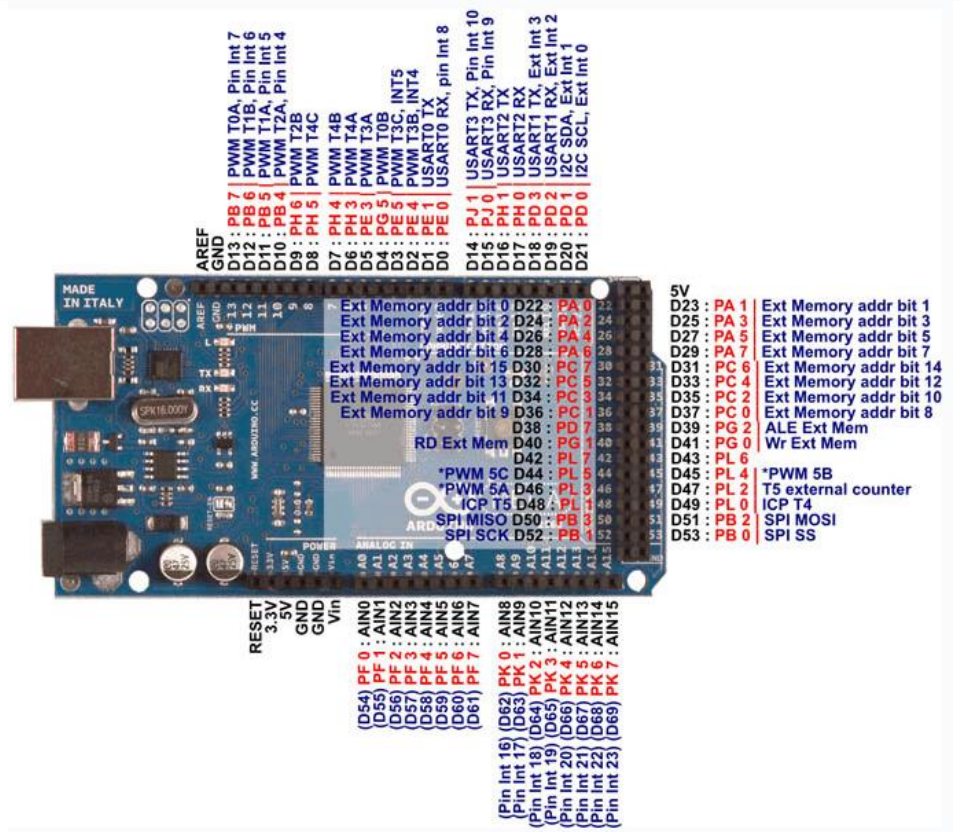
Arduino MEGA, ATmega2560 mikrodenetleyici içeren bir Arduino kartıdır. Arduino Uno 'dan sonra en çok tercih edilen Arduino kartı olduğu söylenebilir. Arduino'nun kardeş markası olan Genuino markasını taşıyan Genuino Mega 2560 kartı ile tamamen aynı özelliklere sahiptir.

Genuino markası, Avrupa'da geçerli olmakla birlikte, Amerika'da Arduino olarak kullanılmaktadır.



Arduino Mega 2560 R3 ön yüz

Arduino Mega 2560 R3 arka yüz

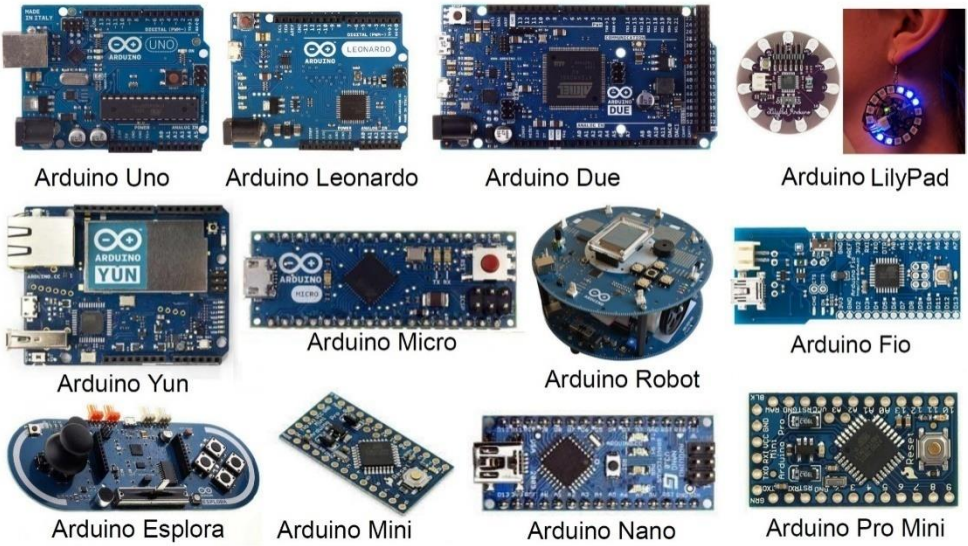


Teknik Özellikleri

Mikrodenetleyici	Atmega2560
Çalışma Gerilimi	+5 V DC
Tavsiye Edilen Besleme Gerilimi	7 - 12 V DC
Dijital Giriş / Çıkış Pinleri	54 ADET/ 15ADET PWM Çıkış
Analog Giriş Pinleri	16ADET
Giriş / Çıkış Pini Başına Düşen DC Akım	40 mA
Flash Hafıza	Atmega2560 256KB
SRAM	Atmega25608KB
EEPROM	Atmega25604KB
Saat Frekansı	16 MHz

---

4. Diğer Arduino Çeşitleri



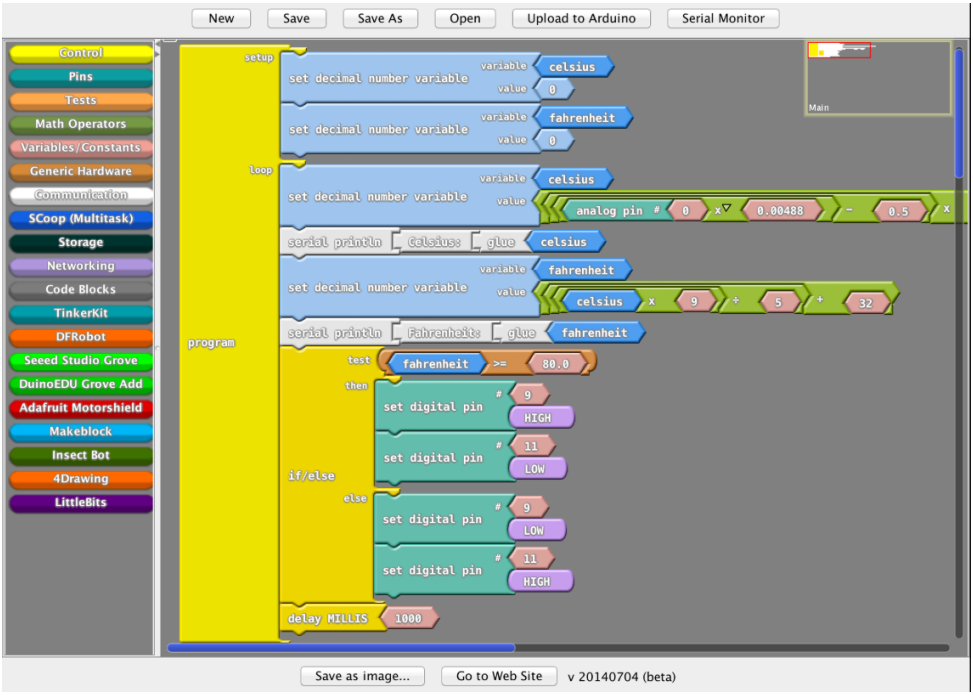
A.5. Programlama Türleri ve Geliştirme Ortamları

ARDUINO'nun programlanması, temel olarak iki ana başlıkta incelenebilir. Kod yazma süreçlerine dahil olmadan, BLOCK tabanlı algoritma geliştirme ortamlarından, ARDUINO programlamaya uyumlu olan araçları kullanılabilir. Eğitim-Öğretim süreçleri bağlamında kullanışlı ve öğrenci veya yeni öğrenen yetişkinlerde ilgi çekici ve pratik olarak hızlı netice vermesi, kolay öğrenilmesi bakımından olumlu olarak değerlendirilmesine rağmen, kullanıcının seviyesi arttıkça ve yapılan uygulamaların karmaşıklık düzeyi yükseldikçe, sorunlar oluşturabilmektedir. Başlangıç düzeyi için uygun olmakla beraber, ileri seviyede kaçınılmaz olan sonuç, diğer tüm Mikrodenetleyici türevlerinde olduğu gibi, üretici tarafından geliştirilen, kod tabanlı editör ortamlarına geçiş yapmaktır.

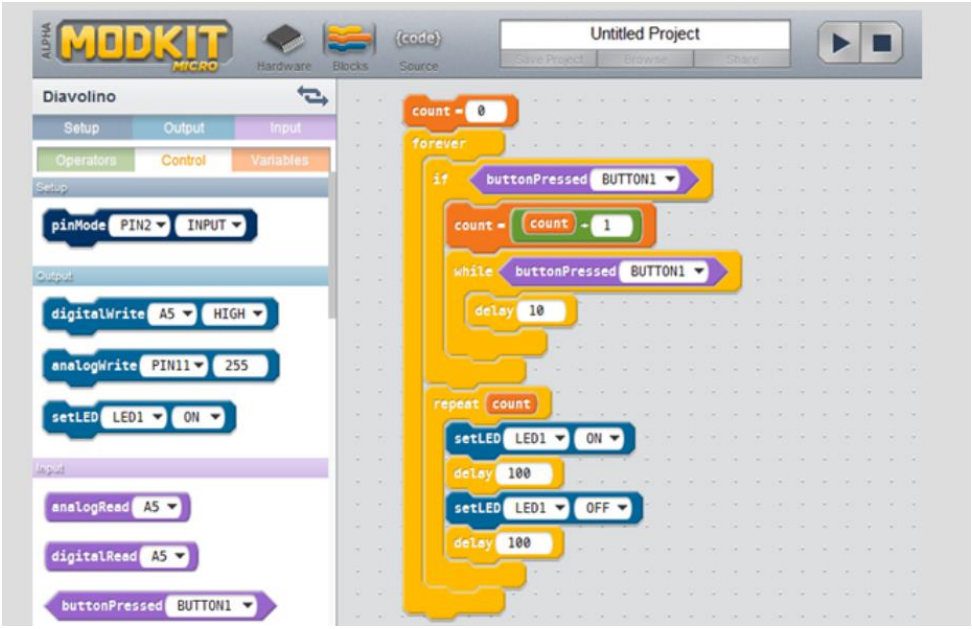
Bu durum, ARDUINO ve diğer tüm Mikrodenetleyici türevlerinde geçerli olan temel programlama yaklaşımıdır.

BLOCK tabanlı ARDUINO programlama araçlarının, tamamında bulunmamakla birlikte, bir diğer dikkat edilesi gereken noktası da, ARDUINO üzerine yazılan program kodlarının kalıcılığıdır. Birçok BLOCK tabanlı geliştirme ortamında yazılan kodlar, PC-ARDUINO iletişimi kesildiğinde, çalışmaya devam etmemektedir. Bu bilgiler ışığında ARDUINO üzerine programlama yapılabilecek birkaç BLOCK tabanlı geliştirme ortamı aşağıda listelenmiştir.

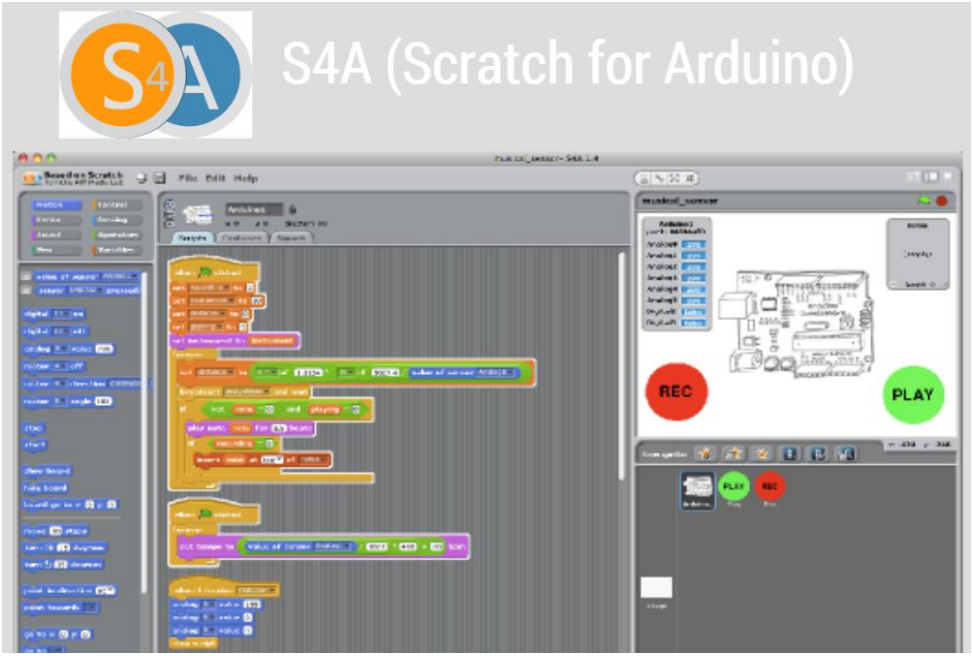
1. ARDUBLOCK



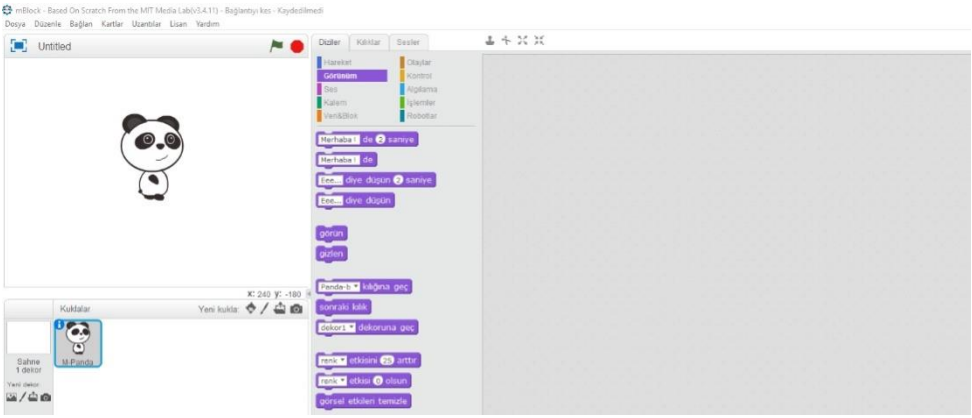
2. MODKIT



3. S4A (SCRATCH FOR ARDUINO)

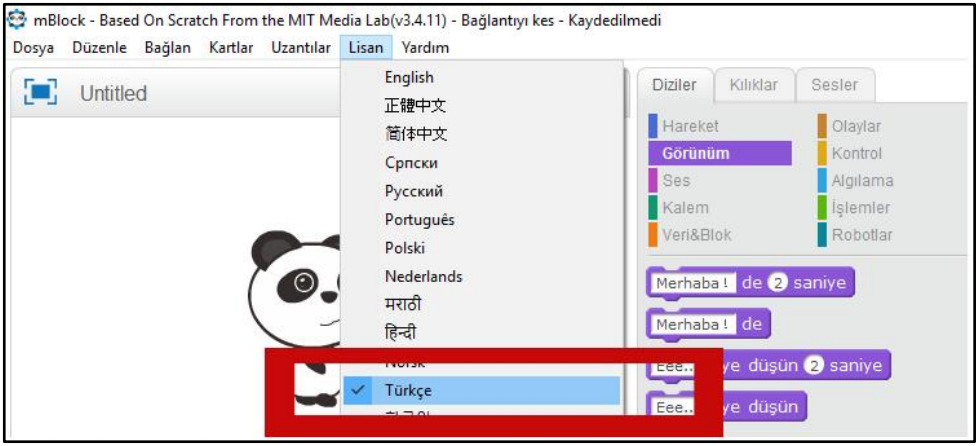


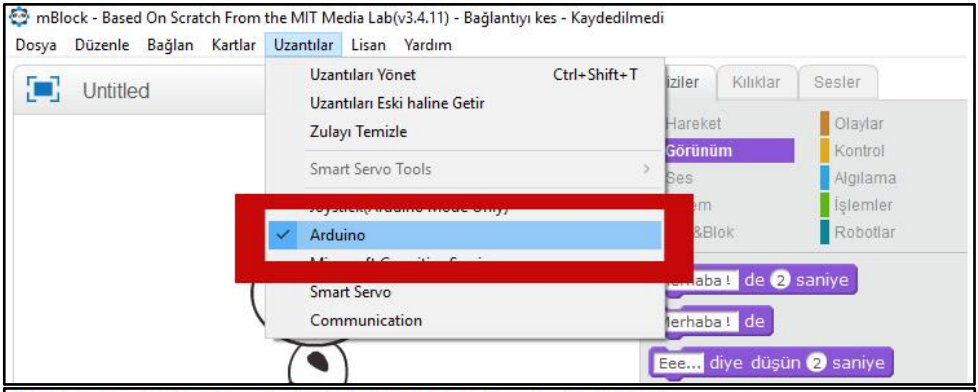
4. MBLOCK

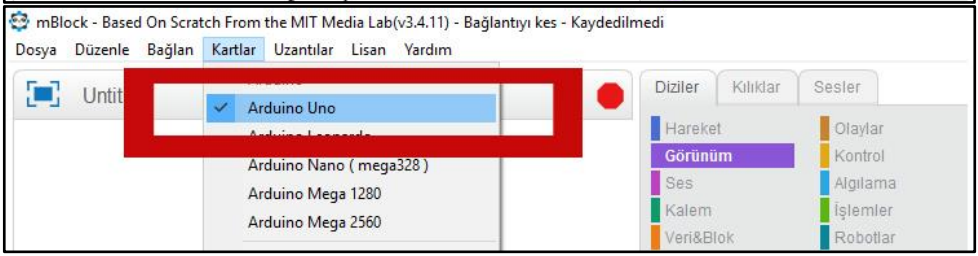


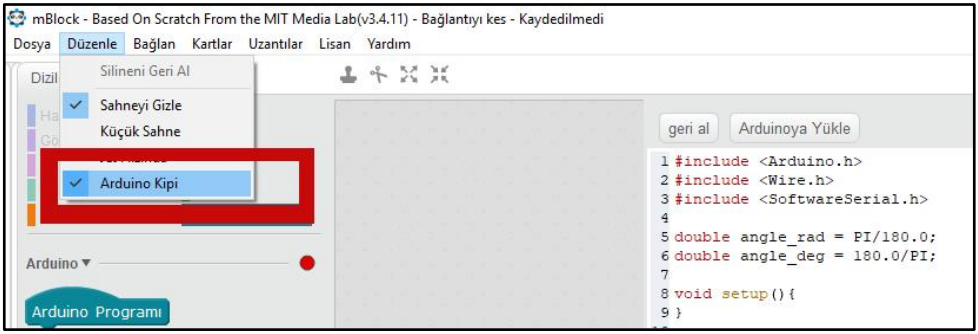
MBLOCK, birkaç farklı robotik ürünü programlayabilmesi, kullanım kolaylığı ve kodlama uyumluluğu bakımından, BLOCK tabanlı ARDUINO programlamada en uygun araç olma özelliğine sahiptir. Ayrıca, ARDUINO uygulamalarına başlamadan, oyun tabanlı temel kod yazma becerisi ve programlama mantığı kazandırmak amacıyla kullanım özellikleri olan bir BLOCK tabanlı geliştirme ortamıdır. Program kurulumu gerçekleştirildikten sonra, ARDUINO bağlantısı ve programlaması için gerekli ayarlamalar, aşağıdaki biçimdedir.

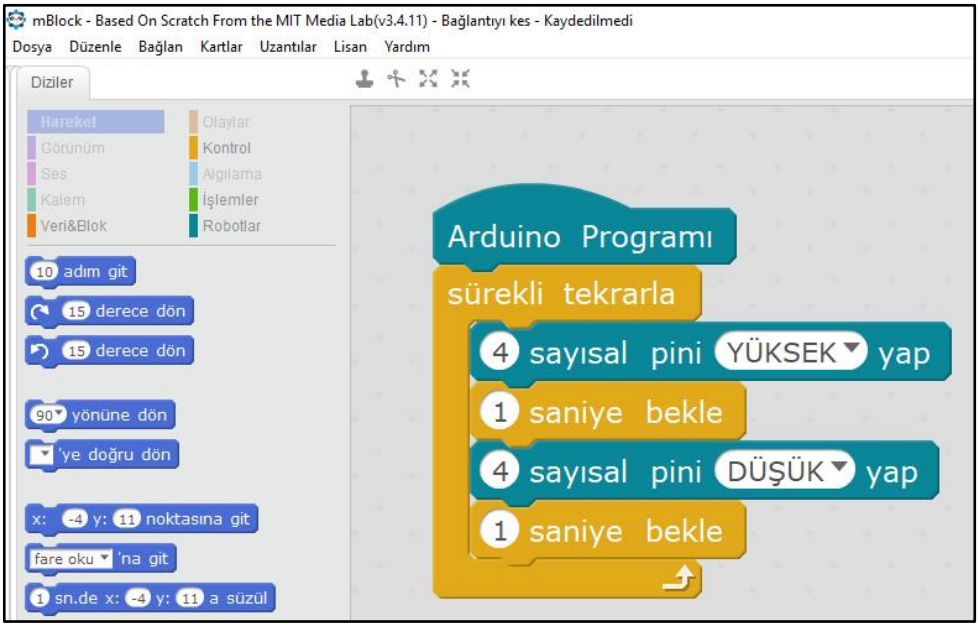
http://www.mblock.cc

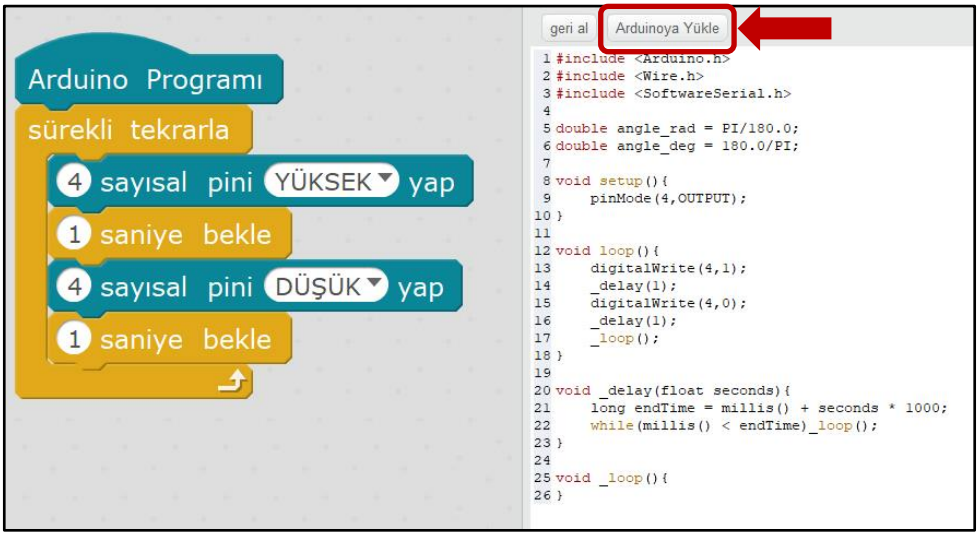

mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Bağlantıyı kes - Kaydedilmedi
Dosya Düzenle Bağlan Kartlar Uzantılar Lisan Yardım
Untitled
English
正體中文
简体中文
Српски
Русский
Português
Polski
Nederlands
मराठी
हिन्दी
Türkçe
Diziler Kılıklar Sesler
Hareket
Görünüm
Ses
Kalem
Veri&Blok
Olaylar
Kontrol
Algılama
İşlemler
Robotlar
Merhaba ! de 2 saniye
Merhaba ! de


mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Bağlantıyı kes - Kaydedilmedi
Dosya Düzenle Bağlan Kartlar Uzantılar Lisan Yardım
Untitled
Uzantıları Yönet Ctrl+Shift+T
Uzantıları Eski haline Getir
Zulayı Temizle
Smart Servo Tools
Arduino
Smart Servo
Communication
Hareket
Görünüm
Olaylar
Kontrol
Algılama
İşlemler
Robotlar
de 2 saniye
de
Eee... diye düşün 2 saniye

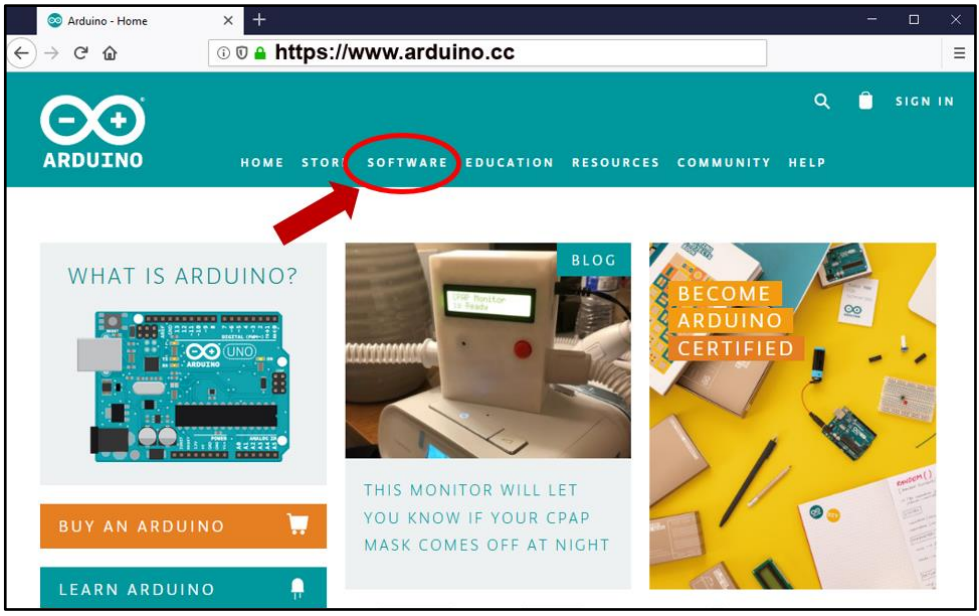

mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Bağlantıyı kes - Kaydedilmedi
Dosya Düzenle Bağlan Kartlar Uzantılar Lisan Yardım
Arduino Uno
Arduino Nano (mega328)
Arduino Mega 1280
Arduino Mega 2560
Diziler Kılıklar Sesler
Hareket
Görünüm
Ses
Kalem
Veri&Blok
Olaylar
Kontrol
Algılama
İşlemler
Robotlar

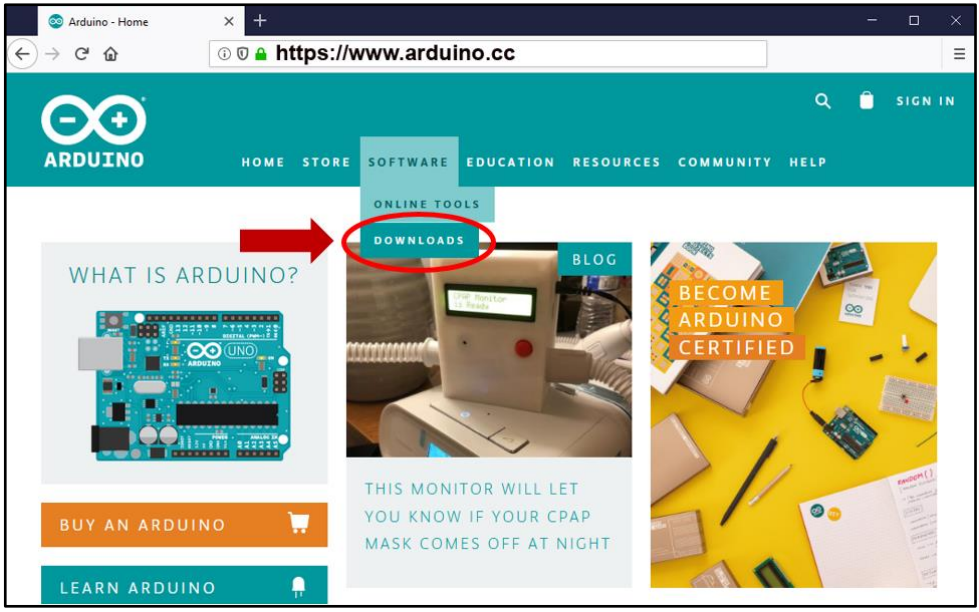

mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Bağlantıyı kes - Kaydedilmedi
Dosya Düzenle Bağlan Kartlar Uzantılar Lisan Yardım
Silineni Geri Al
Sahneyi Gizle
Küçük Sahne
Arduino Kipi
Arduino
Arduino Programı
geri al
Arduinoya Yükle
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 double angle_rad = PI/180.0;
6 double angle_deg = 180.0/PI;
7
8 void setup(){
9 }

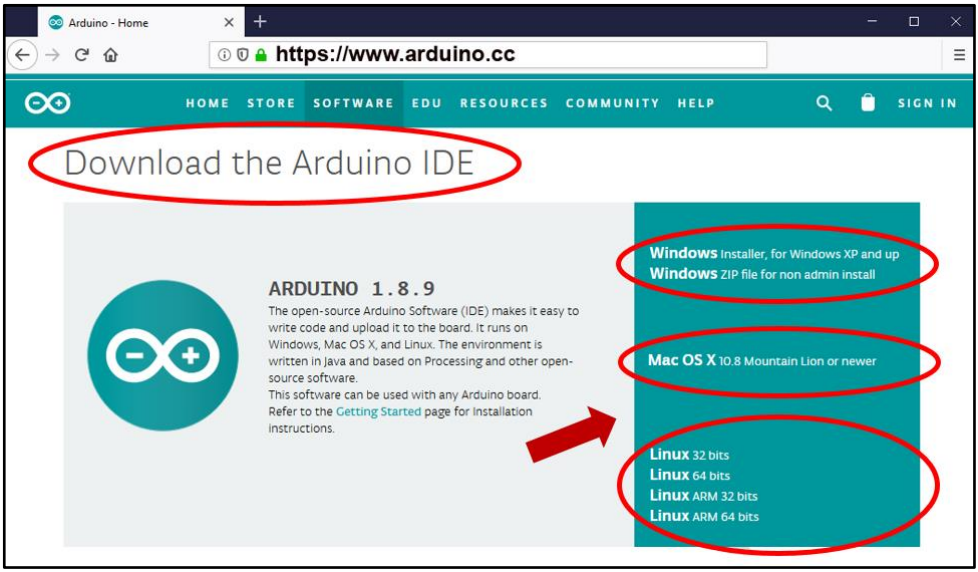

mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Bağlantıyı kes - Kaydedilmedi
Dosya Düzenle Bağlan Kartlar Uzantılar Lisan Yardım
Diziler
Hareket
Görünüm
Ses
Kalem
Veri&Blok
Olaylar
Kontrol
Algılama
İşlemler
Robotlar
10 adım git
15 derece dön
15 derece dön
90 yönüne dön
'ye doğru dön
x: -4 y: 11 noktasına git
fare oku 'na git
1 sn.de x: -4 y: 11 a süzül
Arduino Programı
sürekli tekrarla
4 sayısal pini YÜKSEK yap
1 saniye bekle
4 sayısal pini DÜŞÜK yap
1 saniye bekle

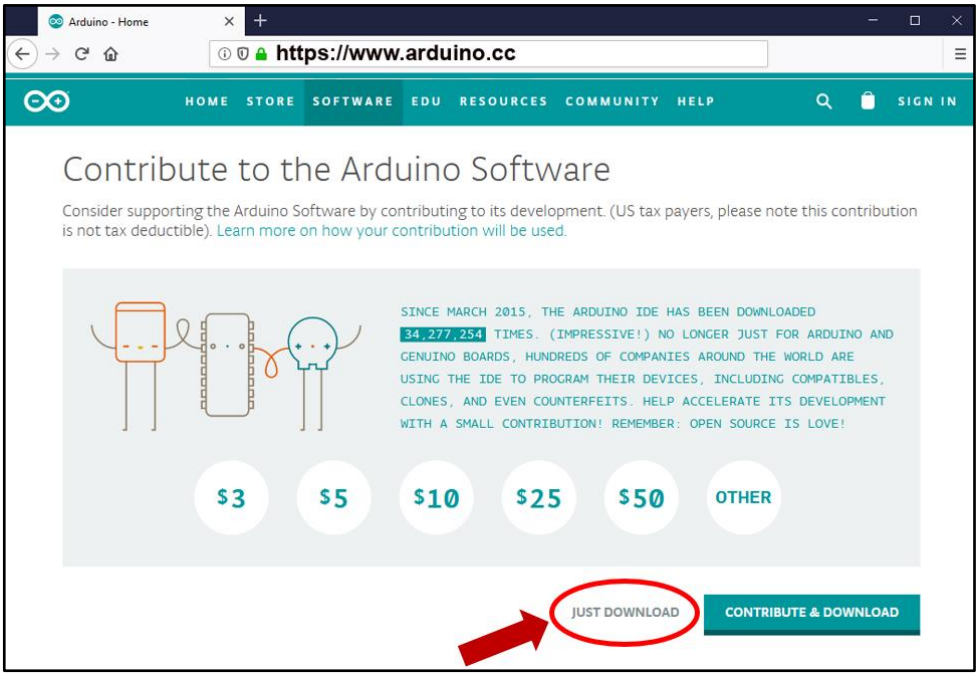

Arduino Programı
sürekli tekrarla
4 sayısal pini YÜKSEK yap
1 saniye bekle
4 sayısal pini DÜŞÜK yap
1 saniye bekle
geri al
Arduinoya Yükle
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
void setup(){
pinMode(4,OUTPUT);
}
void loop(){
digitalWrite(4,1);
_delay(1);
digitalWrite(4,0);
_delay(1);
_loop();
}
void _delay(float seconds){
long endTime = millis() + seconds * 1000;
while(millis() < endTime)_loop();
}
void _loop(){
}

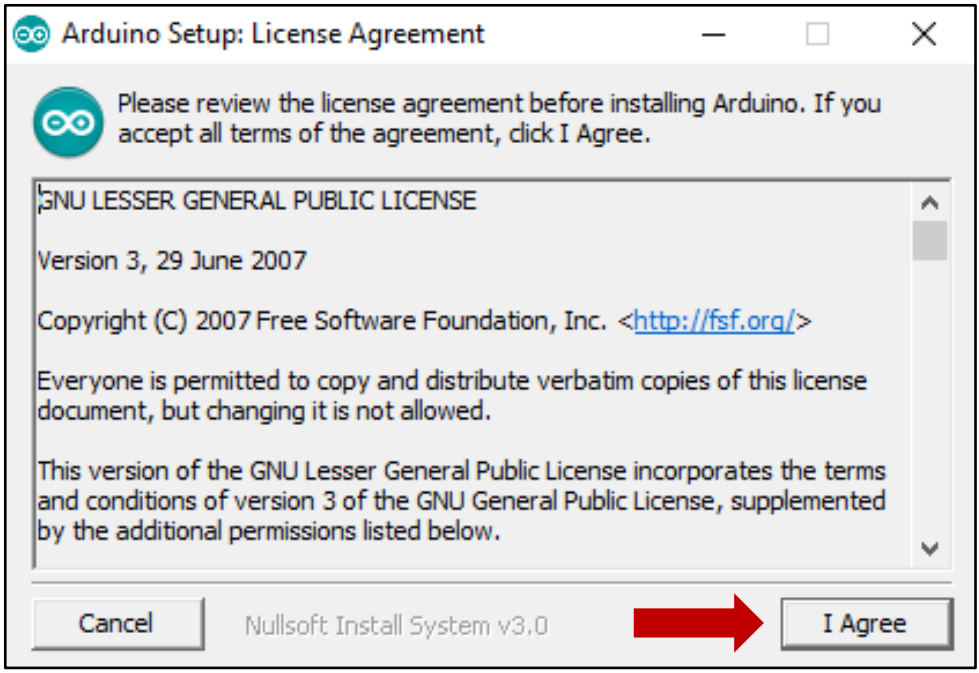
5. ARDUINO IDE

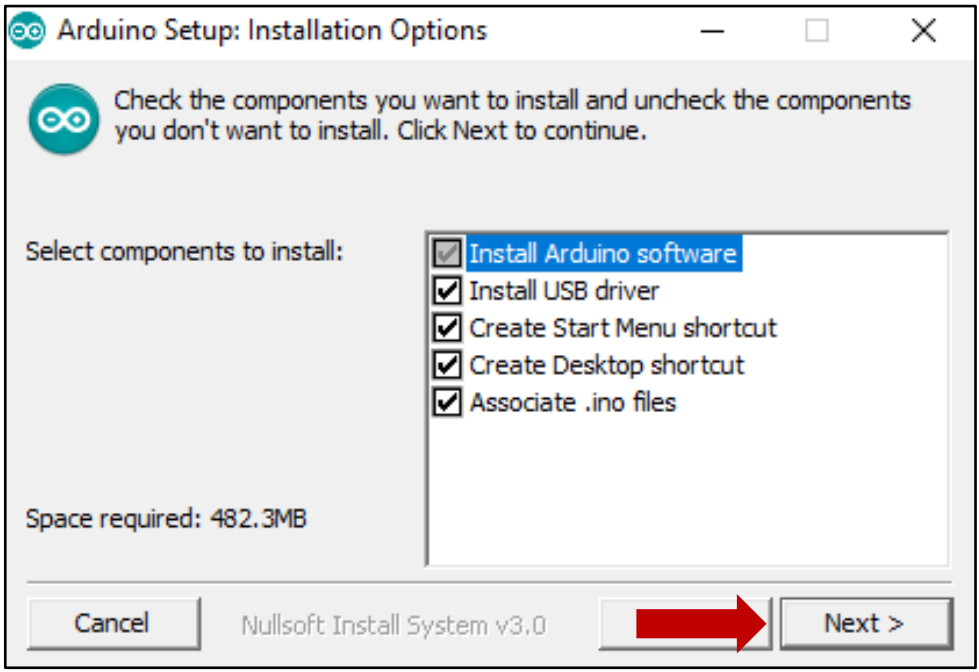


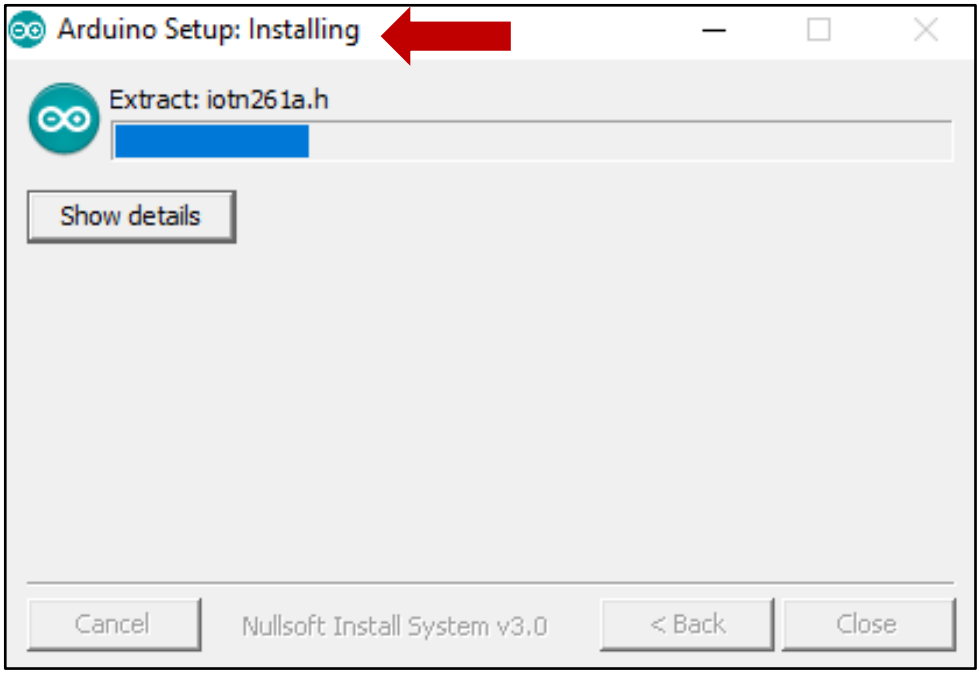


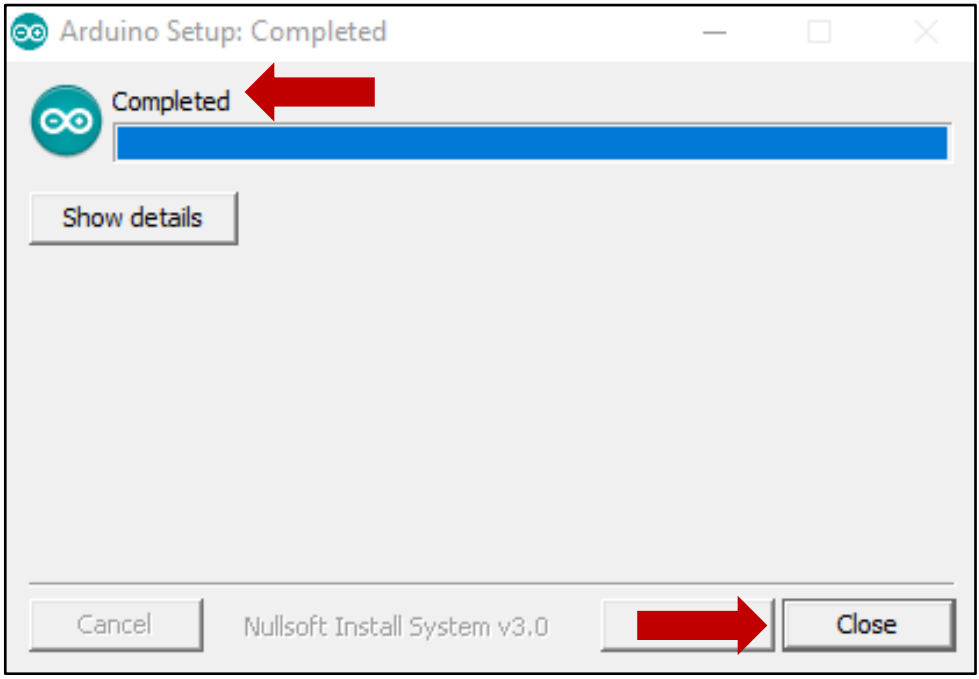

Arduino - Home
https://www.arduino.cc
HOME STORE SOFTWARE EDU RESOURCES COMMUNITY HELP SIGN IN
Download the Arduino IDE
ARDUINO 1.8.9
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for installation instructions.
Windows Installer, for Windows XP and up
Windows ZIP file for non admin install
Mac OS X 10.8 Mountain Lion or newer
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

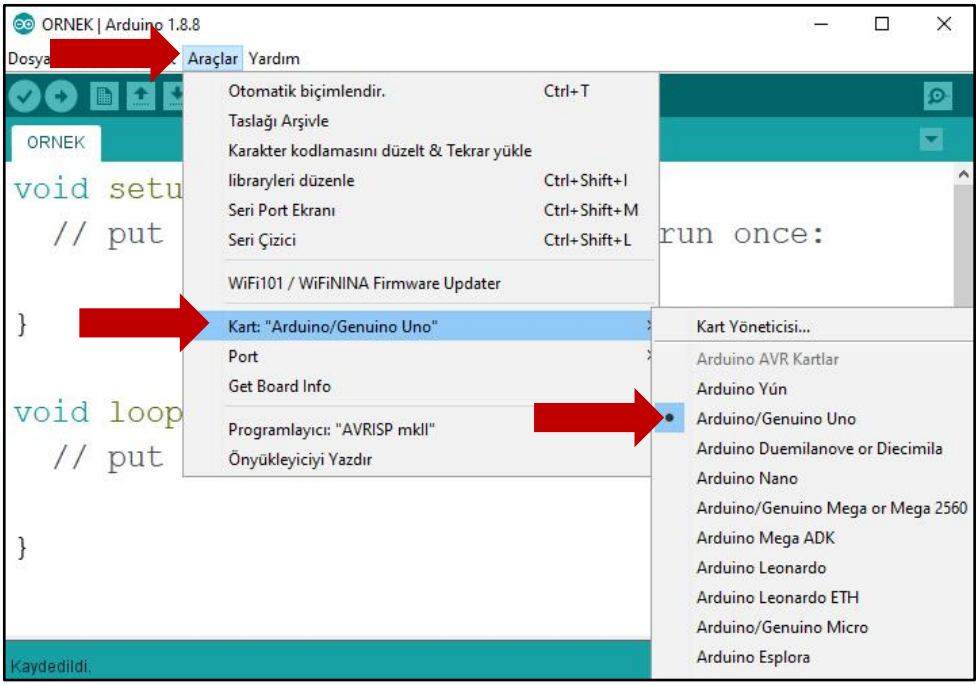

Arduino - Home
https://www.arduino.cc
HOME STORE SOFTWARE EDU RESOURCES COMMUNITY HELP SIGN IN
Contribute to the Arduino Software
Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.
SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED 34,277,254 TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!
$3
$5
$10
$25
$50
OTHER
JUST DOWNLOAD
CONTRIBUTE & DOWNLOAD

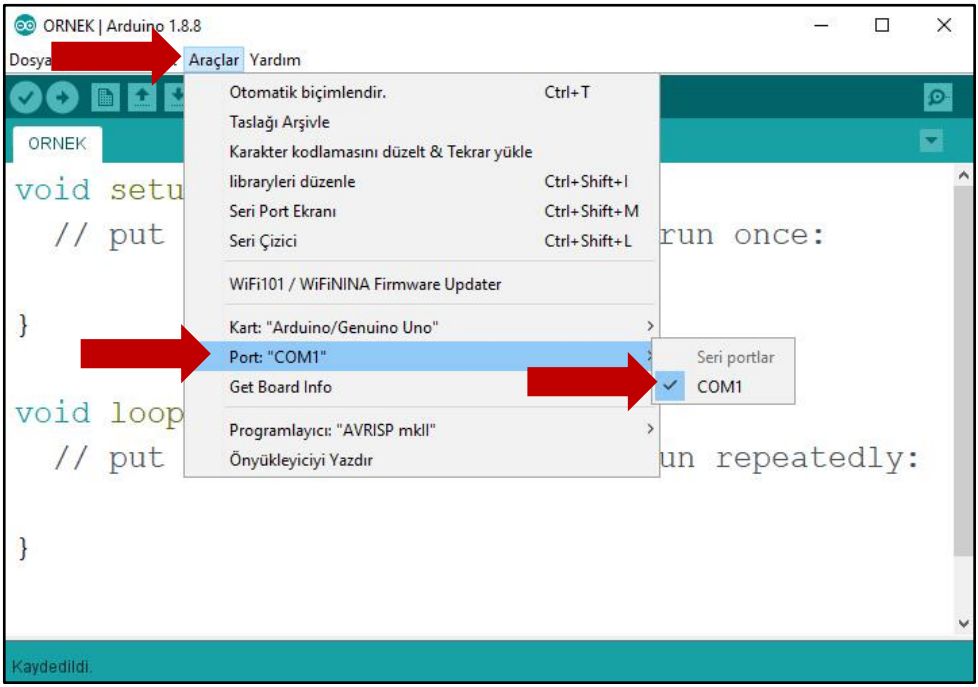

Arduino Setup: License Agreement
Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.
Cancel
Nullsoft Install System v3.0
I Agree


Arduino Setup: Installation Options
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Install Arduino software
Install USB driver
Create Start Menu shortcut
Create Desktop shortcut
Associate .ino files
Space required: 482.3MB
Cancel
Nullsoft Install System v3.0
Next >


Arduino Setup: Installing
Extract: iotn261a.h
Show details
Cancel
Nullsoft Install System v3.0
< Back
Close


Arduino Setup: Completed
Completed
Show details
Cancel
Nullsoft Install System v3.0
Close


ORNEK | Arduino 1.8.8
Dosya
Araçlar
Yardım
Otomatik biçimlendir. Ctrl+T
Taslağı Arşivle
Karakter kodlamasını düzelt & Tekrar yükle
libraryleri düzenle Ctrl+Shift+I
Seri Port Ekranı Ctrl+Shift+M
Seri Çizici Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Kart: "Arduino/Genuino Uno"
Port
Get Board Info
Programlayıcı: "AVRISP mkII"
Önyükleyiciyi Yazdır
Kart Yöneticisi...
Arduino AVR Kartlar
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino/Genuino Micro
Arduino Esplora
ORNEK
void setu
// put
run once:
}
void loop
// put
}
Kaydedildi.


ORNEK | Arduino 1.8.8
Dosya
Araçlar
Yardım
Otomatik biçimlendir. Ctrl+T
Taslağı Arşivle
Karakter kodlamasını düzelt & Tekrar yükle
libraryleri düzenle Ctrl+Shift+I
Seri Port Ekranı Ctrl+Shift+M
Seri Çizici Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Kart: "Arduino/Genuino Uno"
Port: "COM1"
Get Board Info
Programlayıcı: "AVRISP mkII"
Önyükleyiciyi Yazdır
Seri portlar
COM1
ORNEK
void setu
// put
run once:
}
void loop
// put
un repeatedly:
}
Kaydedildi.

ARDUINO IDE «TEMEL C KODLAMA »

setup() : Programın başlangıç ayarları içindir. Başlangıçta bir defa çalışır.
Bu fonksiyon değişkenler, pin modları, seri iletişim ve kütüphaneler içindir.

loop() :« Loop» Türkçesi«Döngü» dür. Adından anlaşılacağı gibi **setup()** fonksiyonundan sonra sonsuz döngü şeklinde sürekli çalışır. Ana program kodları bu fonksiyon içine yazılır.

```
// Tanımlamalar
void setup() {
   // Kurulum kodları buraya yazılır. (Bir defa çalışır)
}
void loop() {
   // Ana program kodları buraya yazılır. (Sürekli çalışır)
}
```

- Program yazımı belirli kalıpta, bloklar halinde olur.
- Bloklar, { } parantezleri ile oluşturulur.
- Komutlar aynı veya alt alta satırlara yazılabilirler.
- Tüm komutlar, noktalı virgül (;) ile biter.
- Programda kullanılan tüm değişkenler ve bilgi tipleri bildirilir.
- Programın başında kütüphaneler aktifleştirilir/çağrılır.
- Açıklamalar “//” ve “/* */ ” (Birden fazla satır için) ile yazılır.
- #define ile eşdeğer ifade atanır.
- #include ile kütüphane çağrılır.

IF / ELSE

ARDUINO programlamada temel karar komutudur. « if » ten sonra verilen koşul doğru ise bu « if » bloğundaki işlemler, yanlış ise “else” bloğundaki işlemler yapılır.

```
if (koşul) {
   // Koşul sağlanıyor ise yapılacak işlem
}
else {
   // Koşul sağlanmıyor ise yapılacak işlem
}
```

FOR

ARDUINOprogramlama dilinde temel döngü komutudur. « for » içinde yer alan koşul doğru olduğu sürece, döngüdeki işlemler gerçekleştirilir.

```
for (başlangıç değeri; koşul; artım)
{
   // İşlemler
}
```

WHILE

« while » ile belirtilen koşul doğru olduğu sürece, döngü içindeki işlemler gerçekleştirilir.

DO – WHILE

« while » ile belirtilen koşul doğru olduğu sürece, döngüdeki işlemler yapılır.

```
while (koşul) {
    // Koşul doğru olduğu sürece yapılacak işlemler
}
do {
    // Koşul doğru olduğu sürece yapılacak işlemler
} while (koşul);
```

SWITCH / CASE

Seçim yapılarak program akışının istenilen bloklara atlamasını sağlar. «switch» teki değişken, «case»deki hangi değeri alırsa, karşılığındaki işlem yapılır.

```
switch (seçim değişkeni) {
    case 1:
        // Seçim değişkeni «1» olduğunda yapılacak işlem
    break;
    case 2:
        // Seçim değişkeni «2» olduğunda yapılacak işlem
    break;
    default:
        // Varsayılan bağımsız işlem (isteğe bağlı)
}
```

ARDUINO IDE « I/O KODLAMA »

pinMode(Pin_No, Pin_Tür);

Pinleri giriş veya çıkış olarak yapılandırma işlemi yapar. (13, INPUT)

digitalWrite(Pin_No, Lojik_Değer);

Çıkış olarak ayarlanan pinlerin değerlerini, HIGH veya LOW olarak ayarlar.

digitalRead(Pin_No);

Belirtilen digital pin değerini okur.

analogRead(Pin_No);

Belirtilen analog pin değerini okur. (0-1023 arasında değerdir. **10BitADC**)

analogWrite(Pin_No, Değer);

Ayarlanan pinden analog çıkış almayı sağlar. LED parlaklığı, motor hızı ayarlama gibi işlemlerde kullanılır. (0-255 arasında değerdir. **8BitDAC**)

delay (Milisaniye);

Milisaniye biriminde zaman gecikmesi verir.

map (Giriş_Değeri, Giriş_Değeri_Minimum, Giriş_Değeri_Maksimum, Çıkış_Değeri_Minimum, Çıkış_Değeri_Maksimum);

map (Giriş_Değeri, 0, 1023, 0, 255);

Giriş_Değeri değişkeni ile gönderdiğimiz sayı 0-1023 arasında değişebilir ve bu değişimle orantılı olarak fonksiyon 0-255 değerleri arasında geri dönüş yapar.

B. ARDUINO İLE DEVRE KURMA

B.1. Elektronik Elemanlar ve Devre Teorisi

Arduino her ne kadar elektronik devre sistemleri ile çok fazla uğraşmadan, birçok uygulama yapılmasını sağlasa da, uygulamalar çeşitlendikçe ek elektronik elemanlar, algılayıcılar ve dönüştürücüler içeren elektronik devreler kurmak ve Arduino üzerinden bu devreleri programlayıp yönetmek ihtiyacı kaçınılmazdır. Bu sebeple, temel elektrik ve elektronik bilgisi yanında, devre elemanları ile temel seviyede devre teorisi ve çözümleme yaklaşımlarına ihtiyaç duyulmaktadır. Bu bölümde, temel seviye elektrik ve elektronik bilgisi,devre elemanları açıklanacak ve bu elemanlar ile devre kurulumu sırasında ihtiyaç duyulacak olan devre teorisi üzerinde durulacaktır.

Elektrik

Elektrik, yüklü parçacıkların hareketi ile ilgili olan fiziksel olaylara verilen genel isimdir. Elektrik, bilinen anlamda iletken maddelerin akım iletmesi ile ilgili olabileceği gibi, statik elektrik, radyo dalgaları (elektromanyetik dalgalar) ve yıldırım gibi daha farklı fiziksel olaylar da elektrik ile ilgilidir.

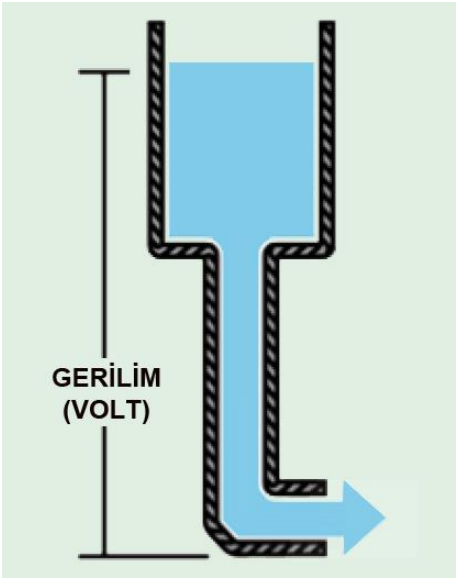
İletken Ve Yalıtkan Maddeler

Elektrik akımını ileten maddelere iletken adı verilir. Metallerin iletken oldukları bilinmektedir (Altın, Bakır, Demir, Gümüş vs). Buna karşılık elektrik akımını taşıyacak serbest elektronlara sahip olmayan, yani elektrik akımını iletmeyen maddelere yalıtkan maddeler adı verilir (Plastik, Kauçuk, Cam, Porselen vs).

Gerilim (Voltaj)

Elektrik gerilimi, akımın oluşması, yani elektronların hareketi için gerekli olan elektrik alan kuvvetidir. Elektrik akımın “akmasını” yani oluşmasını sağlar. Birimi “VOLT” tur. “V” harfi ile gösterilir. Gerilim (Voltaj), elektrik akımını itme gücü olarak tanımlanabilir. Gerilim voltmetre ile ölçülür. Voltmetre ölçüm yapılacak devre elemanına paralel olarak bağlanır.

Kavramları daha kolay anlayabilmek adına elektrik enerjisininsu ile benzerliğinebakılabilir. Gerilimi (Volt) su seviyesinin yüksekliği olarak düşünebiliriz. Su yüksekliği ne kadar fazla olursa, akış o kadar şiddetli olacaktır. Yani gerilim yükseldikçe akacak olan akım şiddeti o kadar artacaktır.



Elektrik Akımı

Güç kaynağı seçimi yaparken, devrenin çalışacağı gerilime uygun bir kaynak kullanılması gerekir. Örneğin 6V motorlara sahip bir robot var ise, bu robotu çalıştırmak için 6V gerilim sağlayabilen bir adaptöre veya bataryaya ihtiyaç olacaktır.

Elektrik akımı, potansiyel fark (gerilim) etkisi sonucunda iletken bir madde üzerinden elektrik yüklerinin hareketi olarak tanımlanabilir. Hareket eden yükler, madde içerisindeki elektronlardır. Birimi "AMPER" dir. "A" harfi ile gösterilir.Denklemlerde "I" harfi ile ifade edilir.

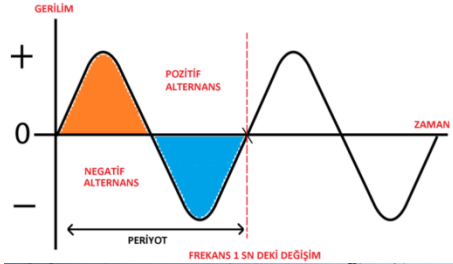
Elektrik devrelerinde asıl işi yapan akımdır. Gerilim akımın akmasını sağlar. Devreyi besleyecek olan pili veya güç kaynağını seçerken devrenin çekeceği en yüksek akımı düşünmek gerekir.

Robot örneğinde olduğu gibi, 6V ile çalışan robotun en yüksek durumda 1A akım çekeceği varsayılırsa, bu durumda en az 6V–1A akım verebilen bir adaptör veya pil kullanılması gerekecektir. Adaptörün verebileceği akımın yüksek olması bir sorun değildir.

Robot ihtiyacı olan 1A akımı çekecek, adaptör ise bu değeri rahatlıkla karşılayabilecektir. 6V–10A bir güç kaynağı da kullanılabilir. Tek fark maliyet ve güç kaynağının boyutu artacaktır. Akım, ampermetre ile ölçülür. Ampermetre devreye seri olarakbağlanır.

AC / DC

AC, Alternating Current (Alternatif Akım) anlamına gelmektedir. İsmini, akımının belirli bir periyot ile pozitif ve negatif alternanslar arasında değişmesinden almaktadır. Üretim ve dağıtımsürecinden, evlere kadar gelen şebeke elektriği, enerji kayıplarının daha az olması sebebiyle alternatif akım şeklindedir.



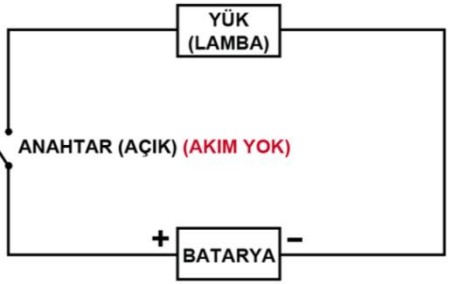
DC ise, Direct Current (Doğru Akım) kelimelerinin kısaltmasından oluşur. Kalem piller, şarj aletleri, güneş panelleri ve daha birçok elektrikli cihazın çıkışı DC'dir.

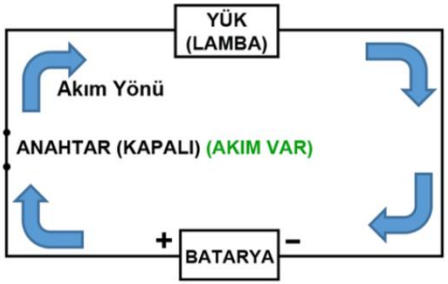
Güç Kaynakları

Bilgisayar, cep telefonu gibi çoğu elektronik cihaz, şebeke gerilimi olan 220V AC'den çok daha düşük gerilim seviyeleri ile çalıştığından adaptörler ile prizden alınan 220V AC gerilim, uygun DC gerilime çevrilir. Günümüzde elektrik enerjisini elde etmenin birçok farklı yöntemi mevcuttur. Hidroelektrik Santralleri (SU), Termik Santraller (YAKIT), Nükleer Santraller (ATOM) ve özellikle günümüzde çok daha önem kazanan yenilenebilir enerji üretim yöntemleri olan Güneş ve Rüzgar Santralleri başlıca elektrik enerjisi üretim yöntemlerindendir.

Açık/Kapalı Devre

Bir elektrik devresinde anahtar açık ise, devrede akım dolaşmaz. Bu tip devrelere açık devre adı verilir ve Yük çalışmaz, Lamba ışık vermez. Anahtarın kapalı olduğu devreye kapalı devre adı verilir. Bu devrede akım dolaşır, Yük çalışır ve Lamba ışık verir.

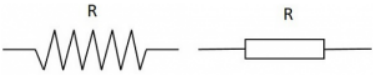




---

a. Direnç, Diyot, Transistör

DİRENÇ, elektrik devrelerinde, iletken üzerinden geçen akımının karşılaştığı zorlanmadır. Elektrik akımına zorluk göstererek, akımı sınırlayan elemana DİRENÇ denir. Mekanik sistemlerdeki sürtünmeye benzer özellikler gösterir. Direncin birimi Ohm (Ω)'dur. Denklemlerde R harfi ile gösterilir. Elektronik devrelerde direncin sembolü iki farklı şekilde gösterilebilir.



Dirençler, elektrik devrelerinde akımı sınırlayarak belli bir değerde tutmaya yararlar. Bunun haricinde hassas devre elemanlarının üzerlerinden yüksek akım geçmesini önlerler, besleme gerilimini ve akımı bölmek için kullanılırlar. Farklı tipteki bazı dirençler pasif algılayıcı görevi görerek ortamdaki fiziksel değişimleri algılayabilirler.Çeşitlerine göre dirençlerin farklı kullanım alanları vardır.

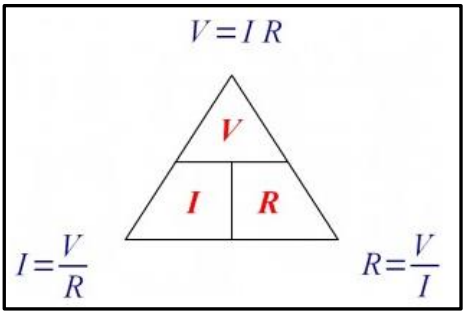
Direnç, yukarıda belirtilen görevleri sebebiyle neredeyse tüm elektronik devrelerde bulunmaktadır.

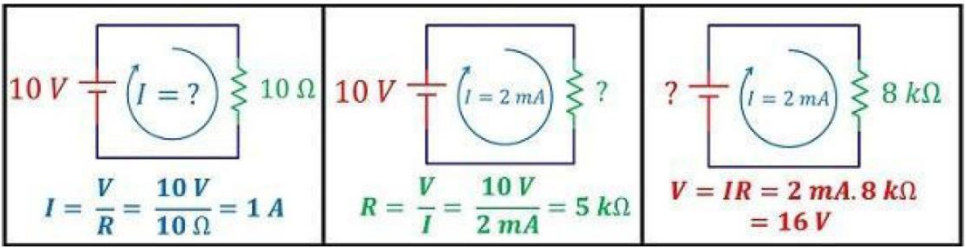
Kullanım alanlarına göre diğer tipteki dirençler farklı görevlerde de kullanılırlar. Potansiyometre gibi ayarlı dirençler, çıkış sinyalinin kontrol edilmesi istenilen devrelerde sıklıkla kullanılırlar. LDR tipi dirençler, üzerlerine düşen ışık şiddetine göre davranan bir algılayıcı görevi görürler; ışığa duyarlı devrelerde kullanılırlar. NTC-PTC termistörleri ise üzerlerine uygulanan ısıya göre davrandıklarından dolayı yine bir algılayıcı görevi görerek ısıya duyarlı devrelerde kullanılırlar.

Bir elektrik devresinde direnç değeri Ohm Yasası sayesinde hesaplanır.İki uçlu bir devre elemanının direnci, üzerindeki gerilimin (V), eleman üzerinden geçen akıma (I) bölümü ile hesaplanır.

Elektronik devrelerde akım, gerilim ve direnç arasındaki ilişki ise Ohm yasası ile hesaplanmaktadır.

Ohm yasası, Alman fizikçi George Simon Ohm tarafından 1827 yılında bulunmuştur. Bu yasa, elektriğin temel yasalarındandır. Bir elektronik devrede iki nokta arasında bulunan iletkenin üzerinden geçen akım, üzerindeki potansiyel fark ile doğru orantılı, sahip olduğu direnç ile ters orantılıdır.





Direnç Renk Kodları Hesaplama

Renk	1. Şerit (ilk basamak	2. Şerit (ikinci basamak	3. Şerit (üçüncü basamak	4. Şerit (katsayı)	5. Şerit (tolerans)	6. Şerit (sıcaklık katsayısı)
Siyah	0	0	0	x10^0		
Kahve	1	1	1	x10^1	± %1	100
Kırmızı	2	2	2	x10^2	± %2	50
Turuncu	3	3	3	x10^3	± %3	15
Sarı	4	4	4	x10^4	± %4	25
Yeşil	5	5	5	x10^5	± %0.5	
Mavi	6	6	6	x10^6	± %0.25	10
Mor	7	7	7	x10^7	± %0.1	5
Gri	8	8	8	x10^8	± %.05	
Beyaz	9	9	9	x10^9	± %1	
Altın					± %5	
Gümüş					± %10	

İlk şerit **KAHVERENGİ,**ilk basamak**1**. İkinci şerit **SİYAH,**ikinci basamak**0**. Elde edilen sayı **10**. Üçüncü basamak**KIRMIZI,**Çarpan Katsayısı **2**, $\mathbf{10^2}$ yani **100** değerini ifade ediyor. Yani $\mathbf{10 \times 10^2 = 1000}$ **Ohm** değerinde bir dirençtir. Son şerit **ALTIN,**olduğundan direncin **%5**'lik bir hata payı ile üretildiğini gösteriyor. Direnç**1000 Ohm** değerinin **%5**'i yani 50 Ohm daha yüksek veya düşük dirence sahip olabilir.

Direnç 5 şeritli olduğunda, katsayı değerinden önce bir hane daha vardır. Yine 1000 Ohm ± %5 değerine sahip 5 şeritli bir direncin renkleri yukarıdaki gibi olacaktır.

Direnç Çeşitleri

Üretildikleri malzeme çeşidi, kılıf tipleri, değişken-sabit gibi farklı parametrelere göre direnç çeşitleri aşağıdaki gibidir.

Karbon Dirençler

Karbon karışımı veya karbon direnç, toz halindeki karbon ve reçinenin ısıtılarak eritilmesi yolu ile elde edilir. Karışımdaki karbon oranı direncin değerini belirler. Büyüklüklerine göre ¼, ½, 1, 2, 3 watt / 1Ω dan 22 MΩ'a kadar değerlerde üretilirler. Üzerindeki çizgilerden değerleri belirtir. Hobi elektroniğinde en yaygın kullanılan direnç çeşididir.

Ayarlı Direnç (Potansiyometre)

Direnç değeri isteğe bağlı değiştirilebilen dirençlerdir. Kılıfının üstünde en yüksek değeri yazar ve sıfır ile maksimum değer arasında bir istenilen değere ayarlanabilir.

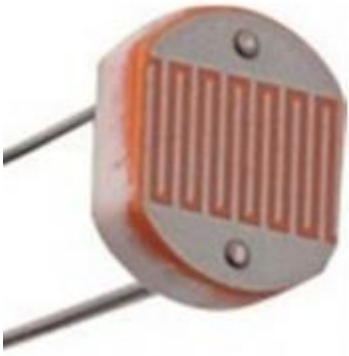
Termistörler (NTC–PTC)

Ortam sıcaklığına göre direnç değerleri değişir. Positive Temperature Coefficient (PTC) sıcaklık artınca, Negative Temperature Coefficient (NTC) sıcaklık azalınca dirençleri yükselen dirençlerdir.

Foto Direnç (LDR)

LDR, Light Dependent Resistor kelimelerinin baş harflerinden oluşan bir kısaltmadır. Bu elemanların yapısında yarı iletken madde olarak kadmiyum sülfat (CdSO4) kullanılmaktadır.

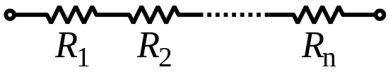
Aydınlık ortamlarda direnci 10Ω'a kadar düşer. Karanlık ortamda direnci 200MΩ'a kadar çıkar. Işığa bağlı olarak direncinin değişmesi özelliği sayesinde sensör olarak kullanılabilir.



Direnç Bağlantıları – Eşdeğer Direnç

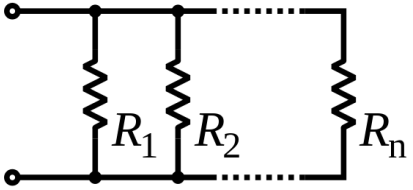
Dirençler, diğer iki uçlu devre elemanları gibi seri veya paralel şekilde birbirlerine bağlanabilirler. Böylelikle mevcut olmayan direnç değerleri elde edilebilir. Farklı bağlantı çeşitlerine göre toplam direnç -eşdeğer direnç- değeri elde edilir.

Seri Direnç Hesaplama



$$Reş = R1 + R2 + R3 + \cdots + Rn$$

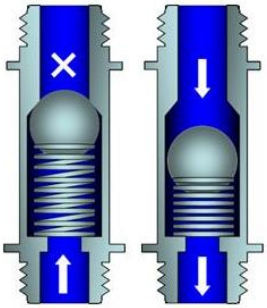
Paralel Direnç Hesaplama

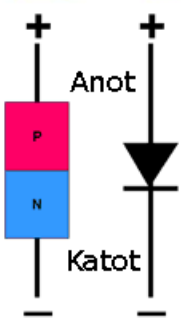


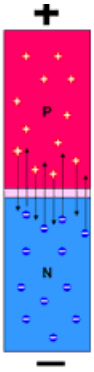
$$\frac{1}{Reş} = \frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} + \cdots + \frac{1}{Rn}$$

DİYOT, elektrik akımının yalnızca bir yönde geçişine izin veren, yarı iletken maddelerden yapılmış iki uçlu bir devre elemanıdır. Devrelerde aşağıdaki diyot sembolü ile gösterilir.

Diyotun **ANOT** ve **KATOT** olmak üzere iki bacağı bulunur. Diyotlar, akımı üzerlerinden yalnızca anottan katoda doğru iletirler. Su ile benzerliğinden örnek vermek gerekirse diyotlar, bir boru üzerindeki çek-valf gibi davranırlar.



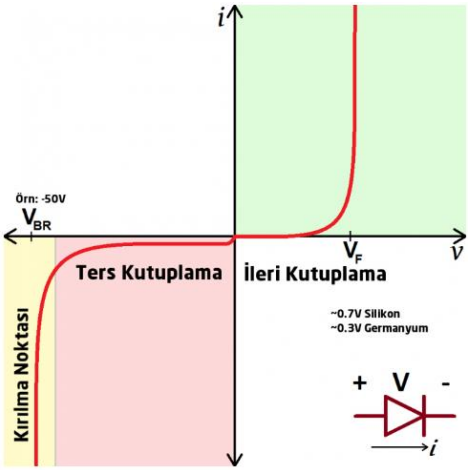




Diyot Çeşitleri

LED Diyot Zener Diyot Schottky Diyot Lazer Diyot Kristal Diyot Tünel Diyot Foto Diyot	Varaktör Diyot IMPATT Diyot Pin Diyot Köprü Diyot Silikon Diyot (1N4001-1N4007) Germanyum Diyot

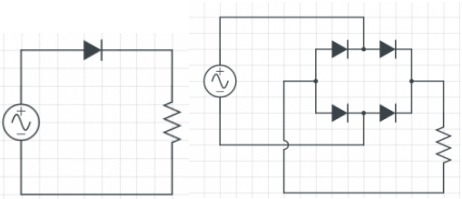
Bir elektronik devre elemanının davranışını (karakteristiğini) anlamak için **AKIM-GERİLİM** grafiğine bakılır.



Diyotun Kullanım Alanları

Diyotların kullanımına örnek olarak verebileceğimiz ilk devre tipi doğrultuculardır.

Doğrultucu, AC gerilimi DC gerilime dönüştürmede kullanılan devrenin ismidir.

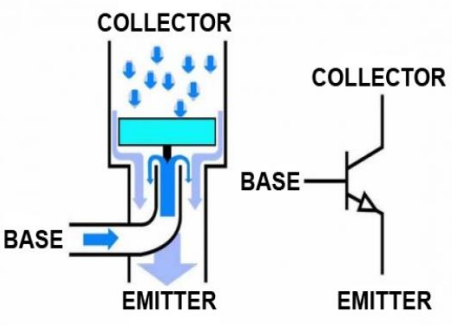


Yarım Dalga Doğrultmaç

Köprü Tipi Tam Dalga Doğrultmaç

Ayrıca Diyotlar, doğrultucu haricinde gerilim regülatörü, ters polarite koruması, lojik devre kapıları olarak, seri bağlı güneş panellerinde baypas amaçlı ve indüktif devrelerde gerilim sıçramalarına karşı koruma amaçlı olarak kullanılırlar.

TRANSİSTÖR, küçük elektrik sinyallerini yükseltmek veya anahtarlamak amacıyla kullanılabilecek bir yarıiletken devre elemanıdır. 3 veya daha fazla bacağı bulunan transistörün bacaklarından birisine uygulanan elektrik sinyali ile diğer bacakları arasındaki elektrik akımını kontrol edebiliriz.



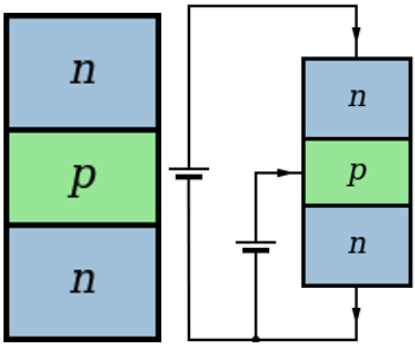
Yarı-İletken Maddeler

Bakır ve demir gibi maddeler iletken, plastik ve seramik gibi maddeler yalıtkandır. Yarı-iletken maddeler ise, normalde yalıtkan olmalarına karşın elektrik gerilimi gibi dış etkilere maruz kaldıklarında iletken olarak davranırlar.Elektronikte genellikle silisyum, germanyum, galyum, arsenit gibi yarı-iletken maddeler kullanılır. Bu maddeler doğadaki saf halleri yerine, işlemden geçirilerek **P-Tipi** ve **N-Tipi** olmak üzere iki farklı çeşide dönüştürülerek devre elemanlarının yapılarını oluşturur.

P-Tipi maddede, pozitif yüklere sahip "delik"ler, maddede yer alan negatif yüklerden daha fazla sayıdadır. N-Tipi yarı-iletkenlerde ise negatif yüklere sahip elektronlar, pozitif yük taşıyıcılardan daha fazla sayıdadır. Bu yük taşıyıcıları, madde üzerinden geçen elektrik akımının hareketini sağlar.

Transistör Nasıl Çalışır?

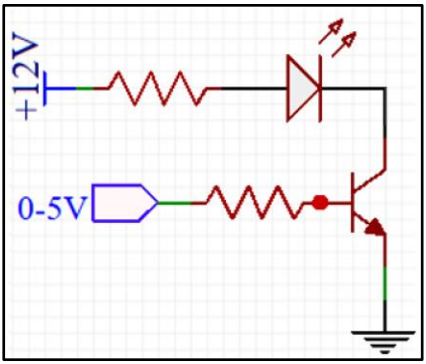
En basit yarı-iletken devre elemanı olan ve akımın yalnızca tek bir yönde akmasını sağlayan devre elemanı olan **DİYOT**,iki farklı şekilde kutuplanabilir.Diyotun anot ucuna pozitif, katot ucuna negatif bir gerilim uygulanması durumuna ileri kutuplama ismi verilir. İleri kutuplanmış bir diyotun anot ucundan katot ucuna akım geçişi olur. Gerilimlerin yerleri tam tersine çevrildiğinde ise diyot ters kutuplama durumundadır. Bu durumda diyot üzerinden akım geçmez. İki adet diyot anot uçlarından birleştirilirse, aşağıdakine benzer bir yapı elde edilir.



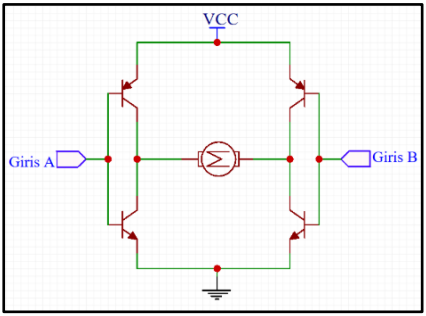
Bu yapının iki ucuna gerilim uygulandığında, güç kaynağı nasıl bağlanırsa bağlansın, **P-N**çiftlerinden bir tane mutlaka ters kutuplu olacaktır. İkinci bir güç kaynağını şekildeki gibi bağladığımızda, orta ve altta yer alan **P-N**çifti ileri kutuplanmış bir diyotu anımsatır. Alttaki **P-N**çiftinin oluşturduğu elektron hareketi, üstteki **P-N**çiftinin de elektronlarını harekete geçirecektir.Ve bir elektron akışı gerçekleşecektir.Bu şekildeki yarı-iletken maddelerden oluşturulan yapıya **Bipolar Junction Transistor (Çift Birleşim Yüzeyli Transistör-BJT)**adı verilir.

Transistörlerin Kullanım Alanları

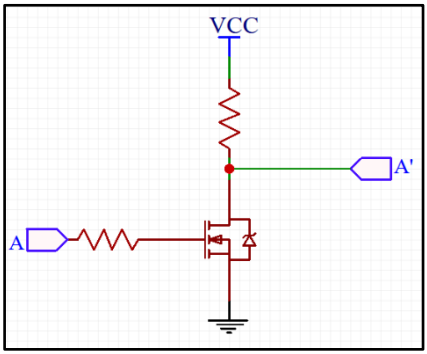
Transistörler sayesinde Arduino gibi mikrodenetleyiciler aracılığıyla LED şeritler, motorlar gibi yüksek güç tüketecek yükleri anahtarlamak mümkündür. Transistör ile LED anahtarlama devresi aşağıdaki gibidir.Devrede 0-5V ile gösterilen bağlantıya PWM sinyali uygulayarak, anahtarlama yüksek hızda yapılabilir, dolayısıyla bağlı olan LED'in parlaklığı veya bağlanan motorun hızı kontrol edilebilir.Motorun yalnızca hızını değil, yönünü de kontrol etmek gerekiyorsa, **H-KÖPRÜSÜ**adı verilen bir devre gereklidir.



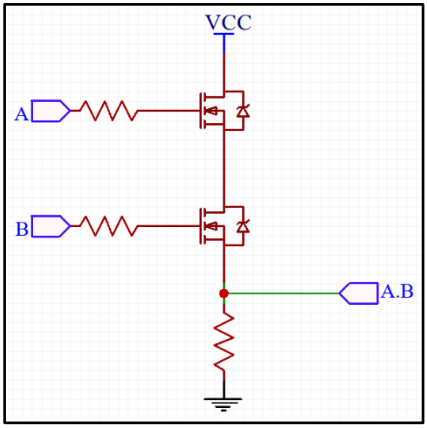
Transistörlü – Led Kontol Devresi



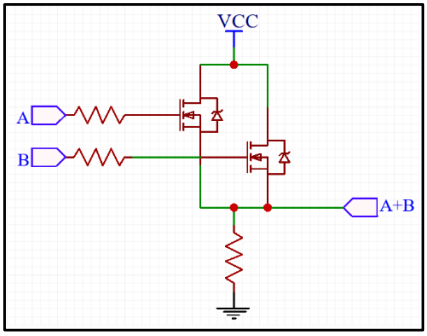
Transistörlü – H-KÖPRÜSÜ



NOT Kapısı (Inverter – Tersleyici)



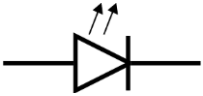
AND (VE) Kapısı



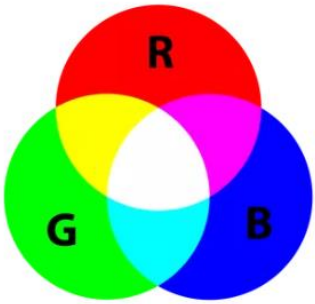
OR (VEYA) Kapısı

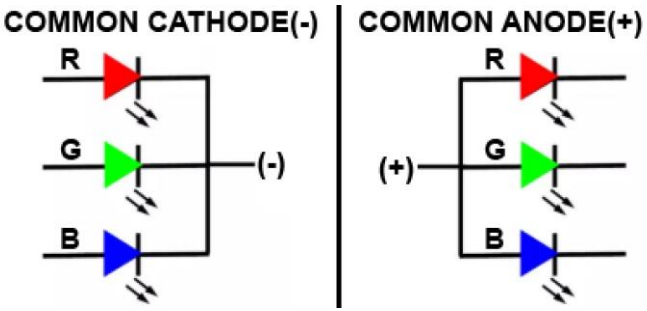
b. LED, RGB-LED, Buzzer, Button

LED-LED Diyot,(Light Emitting Diode – Işık Saçan Diyot) kelimelerinin baş harflerinden oluşan bir kısaltma olan LED, günlük hayatımızda en çok kullandığımız diyot tiplerinden birisidir. Elektronik cihazlarımızın güç ve durum göstergelerinde kullanıldığı gibi, verimli olmalarından dolayı günümüzde aydınlatma amaçlı olarak da sıkça kullanılırlar. Devre sembolüyandaki gibidir.



RGB-LED, normal LED'lerden farklı olarak tek paket içerisinde 3 farklı renk (Kırmızı-Yeşil-Mavi) LED'i bir arada bulundurur. Normal LED'lerin ANOT ve KATOT uçları olmasına karşın, RGB-LED'lerde, üretim şekline göre ANOT veya KATOT bağlantıları ortak olarak bulunmaktadır.



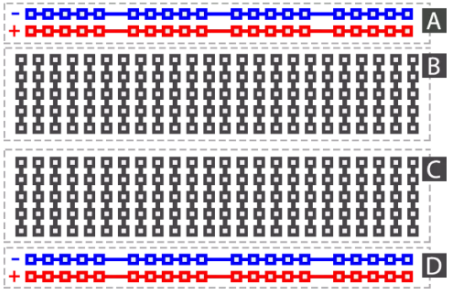


BUZZER,devre elemanıküçük bir hoparlör olarak düşünülebilir. Hoparlörler kadar yüksek ve detaylı ses üretemeseler de, "bip" leme seslerinde oldukça iyidirler.

BUTTON, devre elemanı, devre bağlantılarını açıp-kapatmak amacıyla kullanılmaktadır. İki veya dört bacaklı olmakla birlikte, bağlantı ve çalışma biçimleri aynıdır. Elektronik devrelere tuş sinyali üreterek, farklı görev ve altyordamlar çalıştırılmasını sağlamak amacıyla kullanılırlar.

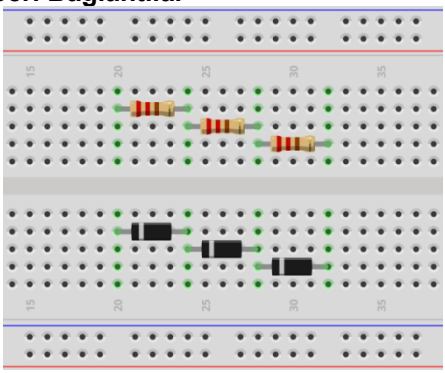
c. Breadboard, Bağlantı Kablosu

Breadboard, üzerinde devrelerin test edildiği bir araçtır. Kurulan devrelerin birbirlerine lehimlenmeden kolaylıkla test edilmesini sağlar. Tasarlanan devreleri baskı devre veya delikli plaketler üzerine aktarmadan önce denemeyeimkanı sunar. Bu sayede devre bağlantılarını kontrol ederek bir hata olup olmadığı gözlemlenmiş olur. Devreler tak-çıkar şeklinde kurulabildiği için, kullanılan elektronik bileşenleri başka projelerde tekrar kullanma imkanı verir.Breadboard iç yapısı dik ve yatay şekilde birbirlerine bağlı halde konumlanmış metal kıskaçlardan oluşur. Dışarıdan bakıldığında görünen kırmızı ve mavi kısımları breadboard-un satır kısımlarıdır. Bu kısımlar boydan boya bir satır şeklinde iletim halindedir. Bu kısımlar breadbord-un iki yanında görülebilir.Burada dikkat edilmesi gereken nokta, bu satır bazı breadboard-larda ortadan ikiye ayrılmış durumdadır. Yani baştan sona kısa devre değildir.Breadboard-un ortada kalan kısımları da sütun boyunca yerleştirilmiş iletkenlerden oluşur. Bu kısımlar da tıpkı satırlarda olduğu gibi breadboard-un iki tarafında bulunur. Tüm bu iletkenlerin üstü elektronik bileşenlerin ayaklarının yerleştirilmesi için açılmış deliklerden oluşan bir plastik ile kapalıdır.

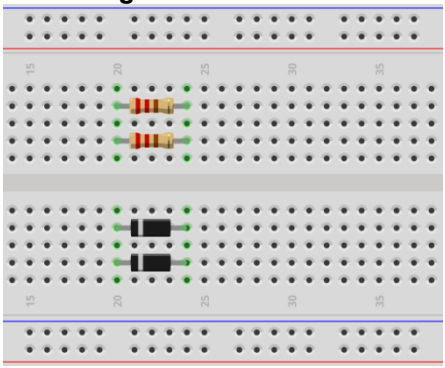


Breadboardlar ebatlarına göre birkaç çeşitten oluşurlar. Hangi breaboardun kullanılacağı, projedeki elektronik bileşenlerin miktarı ve bacak sayıları ile alakalıdır.Projenin büyüklüğüne göre istenilen breadboard tercih edilebilir.Karmaşık yapılı ve çok bileşenli elektronik projelerde büyük breadboard bile yeterli olmayabilir. Bu durumlarda birden fazla breadboardu yanlarındaki çentikler yardımıyla birleştirerek istenilen boyutta bir board elde edilebilir. Genel olarak breadboard-lar**Mini Boy**, **Orta Boy** ve **Büyük Boy** şeklinde sınıflandırılır.

Seri Bağlantılar

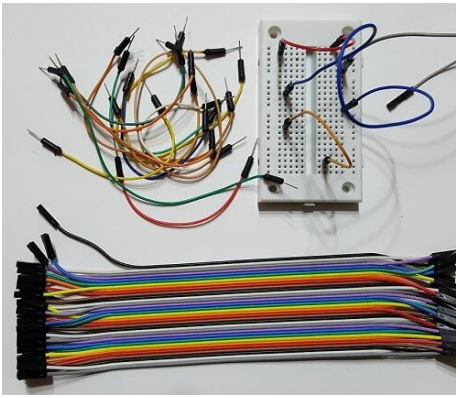


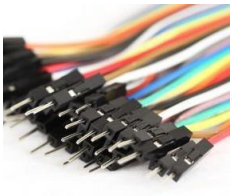
Paralel Bağlantılar



Bağlantı Kabloları (Jumper), özellikle breadboard ile Arduino gibi geliştirme kartlarının bir arada kullanıldığı devreler için oldukça uygundur. Uçlarında **Dişi** ve **Erkek** girişlerin olduğu üç çeşidi bulunmaktadır.

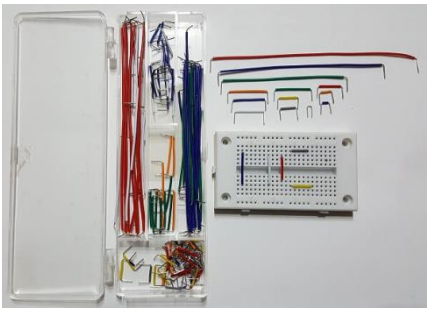
Erkek-Erkek / Erkek-Dişi / Dişi-Dişi



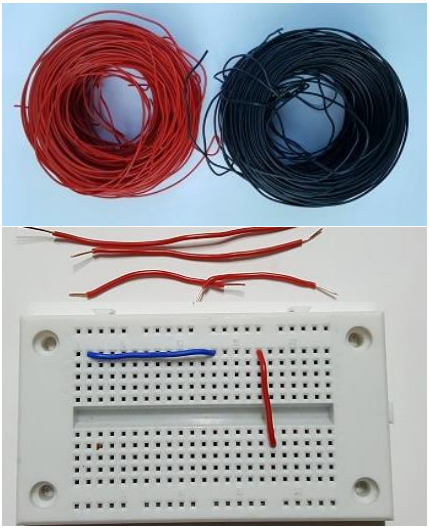


Bağlantı yapılacak girişlere göre, bu çeşitlerden uygun olanları seçilebilir. Ayrıca, farklı çeşitlerin bir arada kullanılması (bu kabloları birbirine bağlayarak), daha uzun iletim kabloları oluşturulmasını sağlar. Devre tahtaları için hazırlanmış, içerisinde değişik boylarda tek damarlı kabloların olduğu kutulu Jumper-lar bulunmaktadır.

Breadboard-a tam oturduğu için, bu seçenek bir öncekine göre kurulan devrenin daha düzenli görünmesine yol açar.



Eğer kutulu Jumper Kablolar pahalı geliyorsa, 100 metre rulolar halinde satılan tek damarlı bakır kablolardan alıp, istenilen boylarda kesip, uçlarını açmak yoluyla, Jumper kablolar daha ucuza hazırlanabilir.

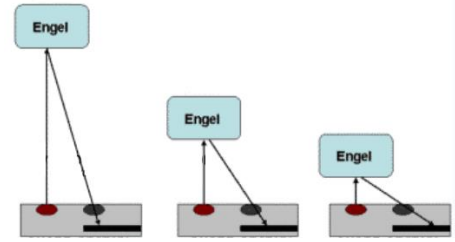


Tek damarlı bakır kablo gibi, ucuza mal edilebilecek bir diğer alternatif ise, CAT-5, CAT-6 gibi internet kablolarını Jumper kablo haline getirmektir. Yalnız, kablonun AWG değerine dikkat etmek gerekir. 23-24 AWG gibi daha kalın türleri breadboard (devre tahtası) için daha uygun olacaktır. 26 AWG fazla incedir.

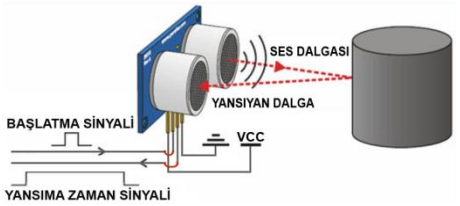
B.2. Algılayıcı ve Dönüştürücüler

a. Mesafe Algılayıcı Elemanlar

Robot ve otomasyon projelerinde bazen sadece engeli algılamak yeterli olmayabilir. Mesafe algılaması gerektiren durumlarda SHARP mesafe algılayıcıları (Sensör) kullanılabilir. SHARP algılayıcıların mesafeyi nasıl algıladıkları aşağıdaki şekilde gösterilmektedir. SHARP algılayıcının yaydığı IR ışık engele çarpıp geri yansır ve yansıyan IR ışık SHARP algılayıcının alıcısı tarafından algılanır. Engel ile SHARP algılayıcı arasındaki mesafeye bağlı olarak IR ışığın yansıma açısı ve ışığın dedektör üzerinde düştüğü nokta değişir. Dedektör bu veriyi okur ve mesafeyi hesaplar.



Kızılötesi (IR) dışında ultrasonik ses dalgaları yoluyla mesafe ölçebilen algılayıcılar da mevcuttur. Ultrasonik ses dalgaları 20kHz-500kHz arasında frekanslara sahip ses dalgalarıdır. İnsanların duyabildiği 300Hz-14000Hz bandının üzerindedir. Ultrasonik algılayıcılar, ultrasonik ses dalgaları yayan ve bunların engellere çarpıp geri dönmesine kadar geçen süreyi hesaplayarak aradaki uzaklığı belirleyebilen algılayıcılardır. Bu algılayıcılarda bu kadar yüksek frekanslarda ses dalgalarının yayılmasının nedeni; bu frekanslardaki dalgaların düzgün doğrusal şekilde ilerlemeleri, enerjilerinin yüksek olması ve sert yüzeylerden kolayca yansımasıdır.



b. LCD Ekranlar ve I2C Sürücüler

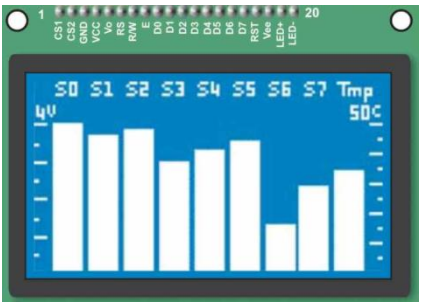
LCD (Liquid Crystal Display - Sıvı Kristal Görüntü), sıvı kristal yöntemi ile görüntü veren ekranlardır. Bu kristal görüntüleri oluşturan her bir noktaya piksel adı verilir. Bu piksellerin birleştirilmesi ile gerçek bir görüntü elde edilir.

Alfanümerik LCD Ekranlar

Bu ekranlar üzerinde sadece harf ve rakam yazdırılabilir. Genellikle basit elektronik devrelerde kullanılırlar.Ölçülen bir değeri veya herhangi bir hesaplamayı göstermek için kullanılırlar.Anlık değer alınan bazı uygulamalarda ve basit elektronik devrelerde kullanılırlar.Alfanümerik LCD modelleri karakter ve satır sayısına göre ifade edilirler. Bunlar **Karakter_Sayısı x Satır_Sayısı** olarak şöyledir. 16×2, 20×2, 20×4.

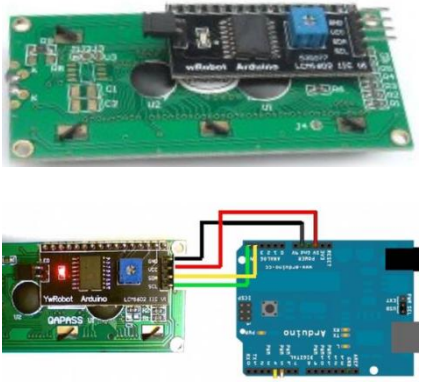
Grafik LCD Ekranlar

Bu tür ekranlar ile grafik görüntüler alınabilir. Yani ekran alanına uygun olarak istenilengörüntü bu ekrana alınabilir. TV ve telefon ekranları Grafik LCD ekranlara örnek verilebilir.



I2C Sürücüsü

Arduino ile LCD ekranı çalıştırmak için 9 adet bağlantı gereklidir. Bu durum bağlantılarda soruna yol açabileceği gibi,Arduino üzerinde çok fazla pin kullanır. Bu sorun I2C protokolü kullanılarak çözülebilir. Bazı LCD ekranlar I2C sürücü modülü dahili olarak bulunmaktadır.



c. DC Motor Çeşitleri ve Sürücüleri

DC motor, düz elektrik akımı enerjisini mekanik enerjiye dönüştüren makinedir. Motorun içinde yer alan sargılara elektrik akımı uygulandığında, yine motorun içerisinde bulunan sabit mıknatıslara zıt yönde oluşan manyetik kuvvetin etkisi ile hareket etme prensibiyle çalışır. Bu akımın yönünün, sürekli olarak sabit mıknatısa ters manyetik alan oluşturacak şekilde değiştirilmesi gereklidir. Bu değişim, fırçalı motorlarda motorun sarımlarına temas eden fırçalar ile, fırçasız motorlarda ise elektronik hız kontrol devresi tarafından yapılır.

Fırçalı DC Motorlar

En temel çeşit DC motor tipidir. Redüktör ile beraber veya redüktörsüz şekilde birçok projede kullanılırlar. Avantajları kolay bir şekilde sürülebilmeleri, dezavantajları ise fırça ya da kömür ismi verilen aşınan parçalarının periyodik olarak değiştirilmesi gerekliliğidir.

Fırçasız DC Motorlar

Fırçalı DC motorların yerini almaları için tasarlanmıştır. Çalışmaları için özel sürücü devreleri kullanılır. Sürtünmenin en az düzeyde olması sayesinde verimliliklerinin çok yüksek olması ve fırça gibi aşınan parça olmaması sebebiyle yüksek performans ihtiyaç duyulan uygulamalarda kullanılabilmektedirler.

Redüktörlü DC Motorlar

DC motorlar çoğunlukla yüksek devir özelliğine sahiptir. Yüksek devir yerine yüksek tork tercih edilen uygulamalarda, motorun miline bağlanan bir dişli seti sayesinde örnek olarak çıkış hızını 30'da 1'e düşürerek, torkun 30 katına çıkması sağlanabilir.

Step Motorlar

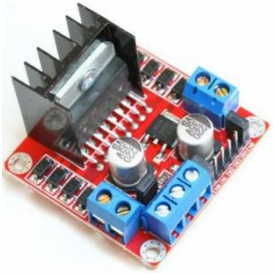
Konumlama hassasiyeti en yüksek motor çeşididir. 2 ve 3 boyutlu yazıcılarda kullanılırlar. Çok hassas konumlama yapabilmelerine karşın, fazla akım çekerler ve hareket hızları fırçalı ve fırçasız motorlara göre daha yavaştır.

Servo Motorlar

0 ile 180 derece açısal aralıkta, istenilen biçimde konumlama yapabilen motorlardır.Sürekli dönebilen tipte servo motorlar bulunsa da, asıl amaçları uçak, helikopter, araba gibi araçlarda kanat, direksiyon, kontrol yüzeyi gibi parçaların açısını kontrol etmektir. Harici bir sürücü devresine ihtiyaç duymazlar.PWM ile çalışırlar.

DC Motor Sürücü Kartları

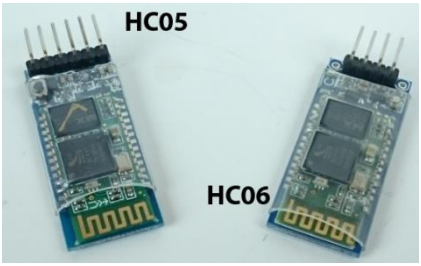
Redüktörlü veya redüktörsüz, fırçalı DC motorları sürmek için H-Köprüsü adı verilen devreler kullanılır. Hazır olarak H-Köprüsü devresi entegresini ve bu entegrelerin ihtiyaç duyacağı devre elemanlarını barındıran motor sürücü kartları tercih edilebilir.



d. Bluetooth ve Kızılötesi (IR) İletişim Elemanları

HC05-HC06 Bluetooth Modülleri

Mikrodenetleyici projelerinde en pratik biçimde kullanılabilen kablosuz bağlantı Bluetooth tekniğidir. Bluetooth haberleşmede, özellikle fiyatından dolayı en sık kullanılan modüller HC05 ve HC06 Bluetooth modülleridir. Her iki modülün fiziksel görünüşleri hemen hemen aynıdır.



HC06 AT Komut Yönergesi

HC06 modüllerini bilgisayar ile konfigüre etmek için şu şekilde bağlantı yapılmalıdır.

HC06 Modül	>>>	USB-Seri Kart
VCC	>>>	3.3V veya 5V
GND	>>>	GND
TXD	>>>	RXD
RXD	>>>	TXD

Arduino IDE üzerinde doğru COM portunun seçili olduğunun kontrolünden sonra, dikkat edilmesi gereken nokta, kartın baud rate'inin, AT komutlarınıgöndermek için varsayılan baud rate olan 9600'e ayarlanmasıdır.Bağlantıyı denemek için, "AT" komutu gönderilebilir. Eğer seri port ekranında "OK" cevabı görünüyorsa, bağlantı düzgün çalışıyor demektir.

Modülün ismini değiştirmek için;
AT+NAME YeniKartAdı

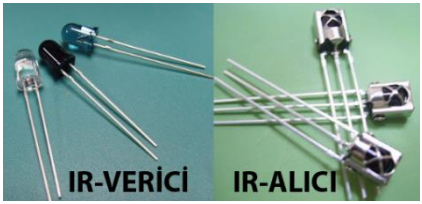
Modülün Şifresini değiştirmek için;
AT+PIN YeniŞifre

Modülün Baud rate'ini değiştirmek için;
AT+BAUD4

(1:1200, 2:2400, 3:4800, 4:9600, vs.)

IR (KızılÖtesi)

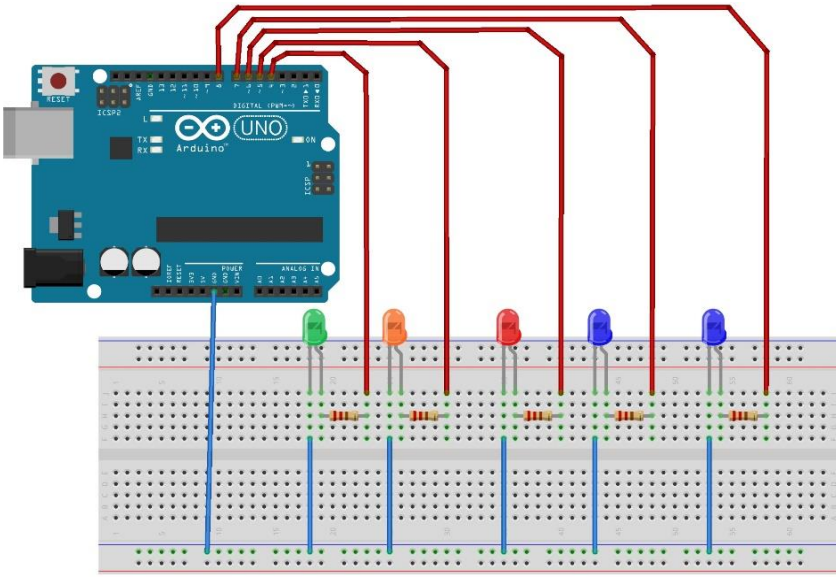
IR, kızılötesi dalga boyunda ışık yayarak veri transferi sağlayan bir sistemdir. Kızılötesi ışık Infrared Led'ten ilgili Infrared Alıcı'ya gönderilir. Bu sistem uzun zamandır birçok alanda mevcut ve bu donanımlara ulaşmak çok kolay ve ucuzdur. Arduino için bu uygulamalarda kullanılmak üzere yapılmış hazır modüller (Shield) olsa da bu malzemeleri elektronik devre elemanları satan mağazalarda bulmak mümkündür.



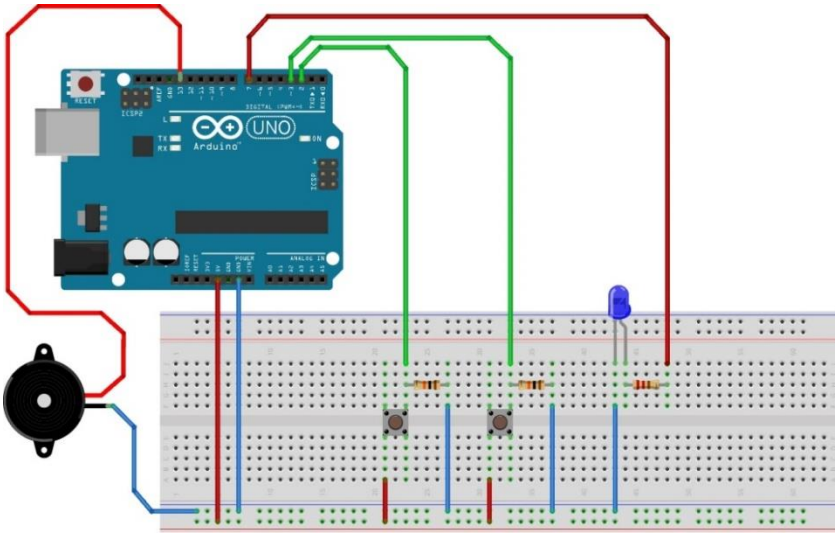
Birçok çeşit IR Alıcı-Verici bulunmaktadır. Burada önemli olan nokta, IR Alıcı pinlerinin çeşitlere göre farklılık göstermesidir. IR tekniği ile,birçok uygulama gerçekleştirilebilir. Yapılması gereken tek şey, kullanılacak olan kumandanın herhangi bir tuşuna basıldığında hangi kodu gönderdiğinin öğrenilmesidir.

B.3. Örnek Devre Uygulamaları

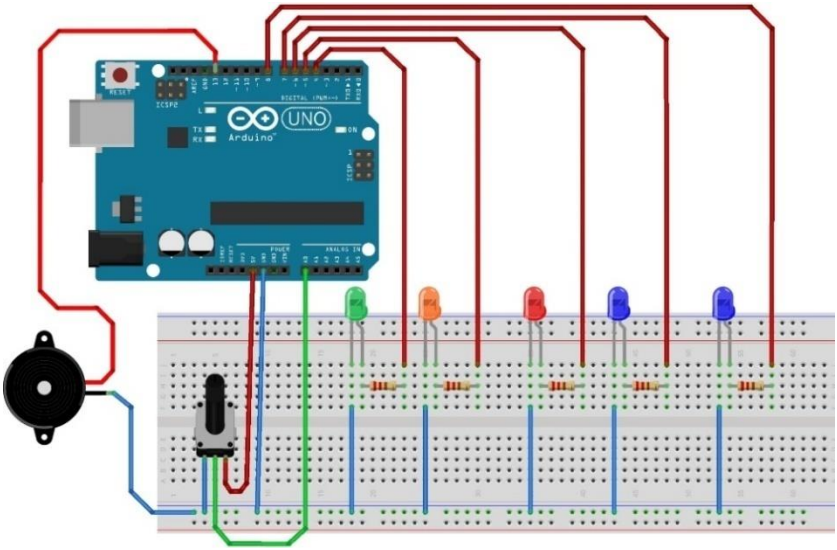
1. Arduino İle 4-5-6-7-8 Numaralı Sayısal Çıkış Pinleri Üzerinden Led Uygulaması



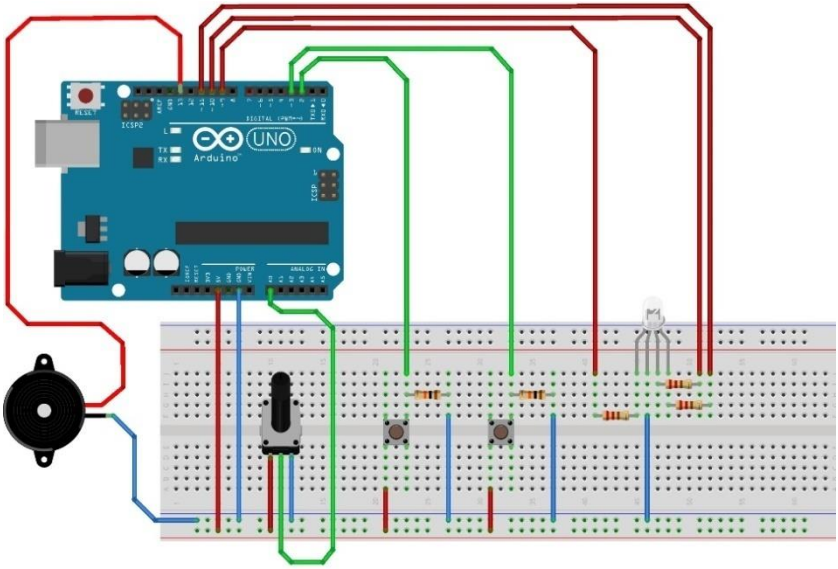
2. Arduino İle 2-3 Numaralı Sayısal Pinlere Button Ve 7-13 Numaralı Sayısal Çıkış Pinleri Üzerinden Led-Buzzer Uygulaması



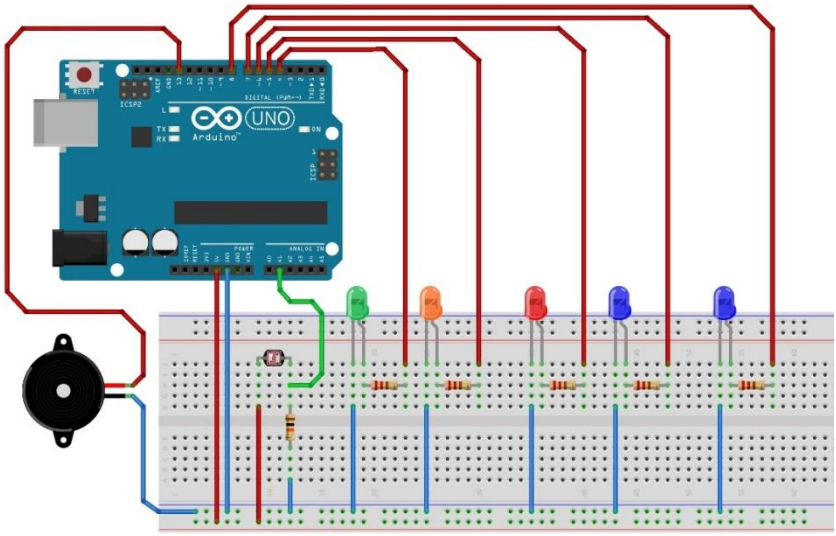
3. Arduino İle 4-5-6-7-8 Numaralı Sayısal Çıkış Pinleri Üzerinden Led Ve 13 Numaralı Sayısal Çıkış PiniÜzerinden Buzzer, A0 Analog Giriş Pini Üzerinden Potansiyometre Uygulaması



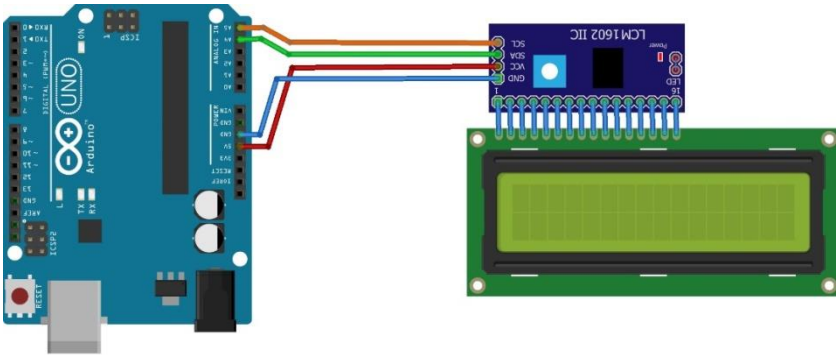
4. Arduino İle 2-3 Numaralı Sayısal Pinlere Button, 13 Numaralı Sayısal Çıkış PiniÜzerinden Buzzer, A0 Analog Giriş Pini Üzerinden Potansiyometre Ve 9-10-11 Sayısal Çıkış Pinleri Üzerinden RGB-LED Uygulaması



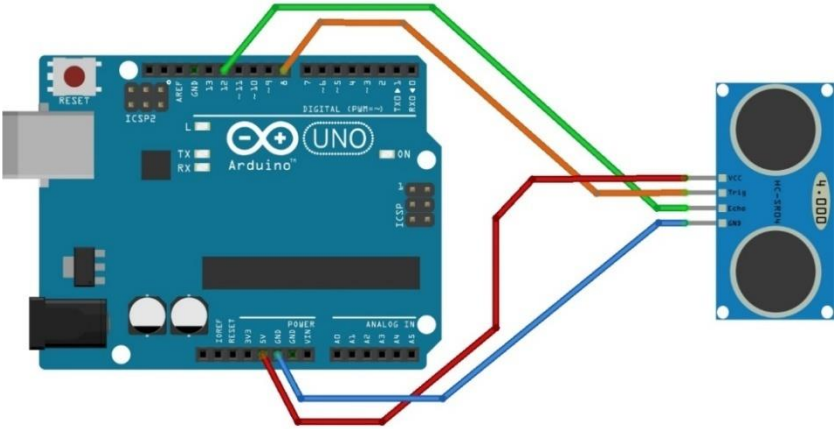
5. Arduino İle 4-5-6-7-8 Numaralı Sayısal Çıkış Pinleri Üzerinden Led Ve 13 Numaralı Sayısal Çıkış PiniÜzerinden Buzzer, A1 Analog Giriş Pini Üzerinden LDR Uygulaması



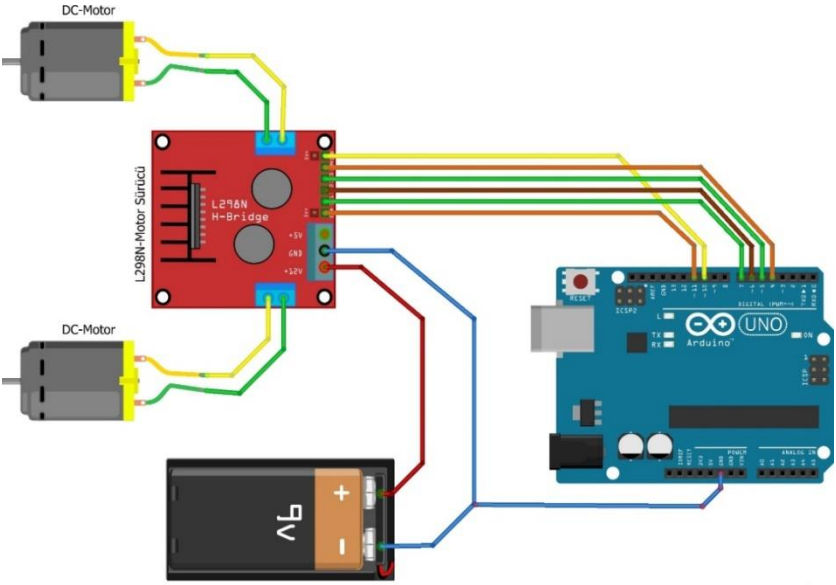
6. Arduino İle I2C – LCD Uygulaması



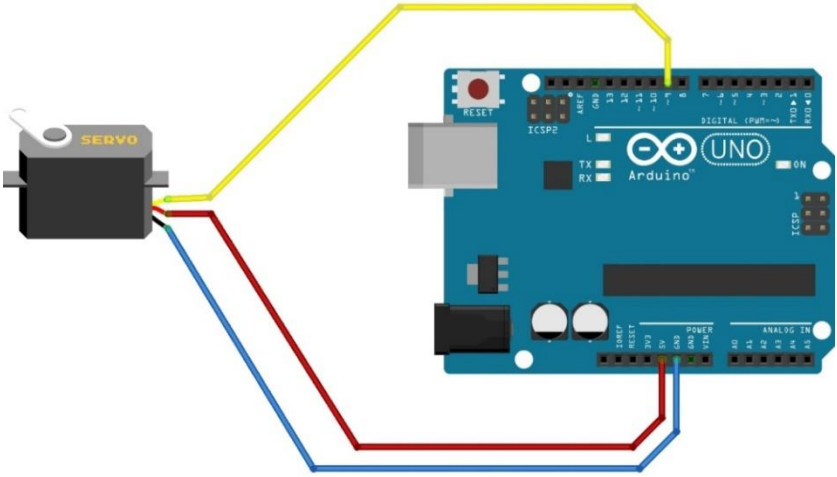
7. Arduino İle Mesafe Algılayıcı (HC-SR04) Uygulaması



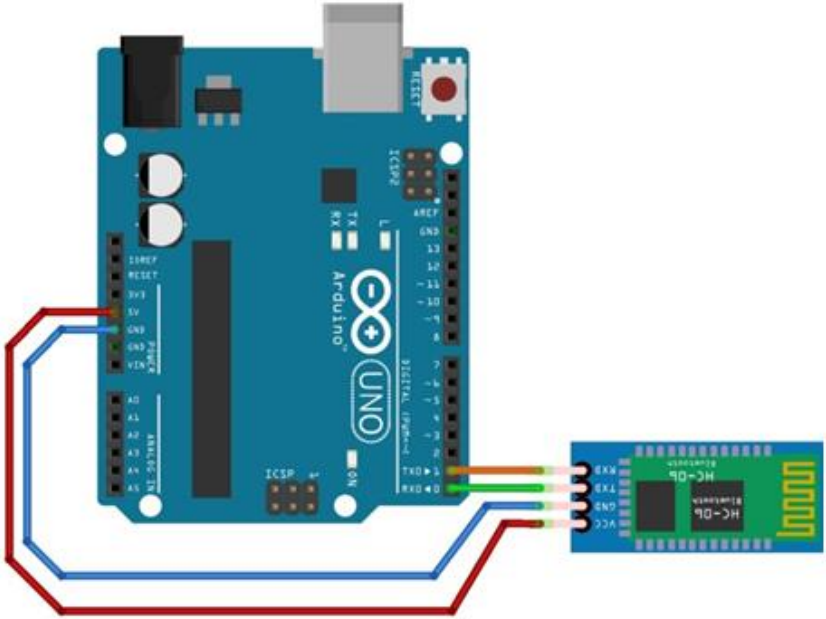
8. Arduino İle DC-Motor Uygulaması



9. Arduino İle Servo-Motor Uygulaması



10. Arduino İle Bluetooth (HC-06) Uygulaması



C. ARDUINO EĞİTİM SETİ TANITIMI

C.1. Genel Tanıtım ve Kullanım Amacı

Önceki bölümlerde belirtilen açıklamalar doğrultusunda, Trabzon İl Milli Eğitim Müdürlüğü AR-GE Birimi ile **T**rabzon **B**ilgisayar **B**ilimleri ve **K**odlama**M**erkezi koordinasyonu kapsamında geliştirilen, "ARDUINO TEMEL EĞİTİM SETİ", adı geçen tüm elektronik ve devre uygulama süreçlerini üzerinde bulunduran bir ürün olarak tasarlanmış ve içeriklendirilmiştir.

Sonraki aşamada ise, **T**rabzon **M**esleki ve **T**eknik **A**nadolu **L**isesi (TMTAL) Döner Sermaye İşletmesi bünyesinde, seri üretimine başlanmış ve halen devam edilmektedir.

Elektronik devre kurulum, bağlantı ve programlama aşamalarının tamamı göz önünde bulundurulduğunda, özellikle yeni başlayan kullanıcılar açısından, eş zamanlı bir şekilde, elektronik ve programlama uygulamalarının yürütülmesi büyük bir zorluk teşkil etmektedir. Bu amaçla, kullanıcıyı/programcıyı, elektronik iş ve işlem yükünden kurtararak, özellikle daha etkili kodlamaya yönlendirmek ve gelişim süreci içerisinde, kullanıcının ilgisini kaybetmesine imkan vermeden, gerekli durumlarda, gerekli miktarda elektronik altyapı ile karşı karşıya kalmasını sağlamak yoluyla, daha etkili bir "Robotik Kodlama" süreci meydana gelmesine sebep olmaktadır.

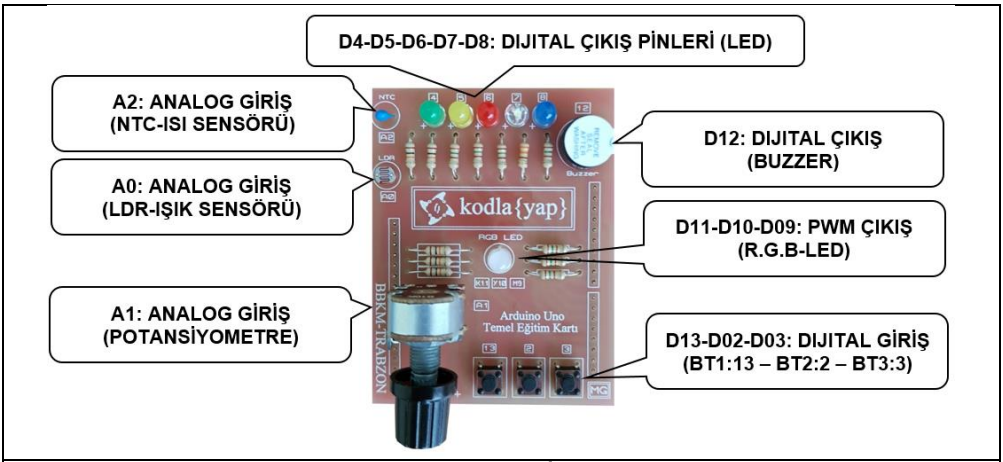
Bu amaçlar doğrultusunda, ilerleyen kısımlarda detaylı teknik tanıtımı yapılacak ve uygulama örnekleri sunulacak eğitim seti, özellikle Robotik Kodlama alanına yeni başlayanlar ve bu alanda eğitim süreçleri uygulayanlar için önemli bir eğitim materyali olabilecektir.

C.2. Çeşitleri ve Özellikleri

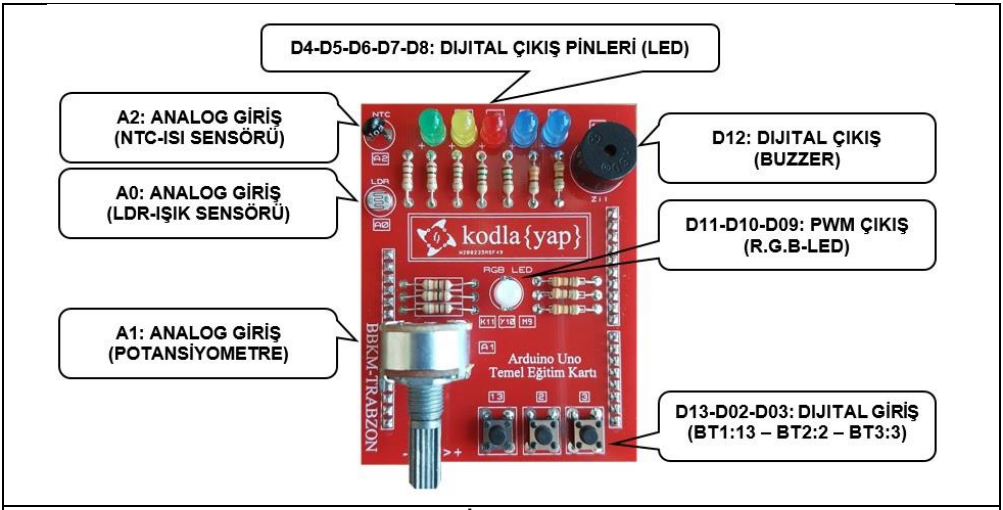
Özellikle kodlama/programlama süreçlerinde hızlı ilerleme sağlaması ve sonraki aşamalarda kullanıcının ilgisini kaybetmeden gerekli oranda elektronik süreçlerle karşı karşıya kalmasını sağlaması sebebiyle, eğitim süreçlerinde çok faydalı olduğu görülen "ARDUINO TEMEL EĞİTİM SETİ", üç aşamalı bir ürünleştirme altyapısına sahiptir.

Birinci aşamada Versiyon 1.0 eğitim kartını barındıran materyal, genişletme elemanlarına uyumluluk taşımaması sebebiyle, geliştirilmesine devam edilmiş ve ikinci aşamada Versiyon 2.0 ile dahili olarak birçok bileşeni üzerinde barındırmakla birlikte, tasarım, üretim, montaj, maliyet ve son kullanıcı taraflı arıza giderme noktasında zorluk çıkarmıştır. Son olarak üçüncü aşamada Versiyon 1.1 ile, tasarım, üretim, montaj ve son kullanıcı taraflı arıza giderme süreçleri optimize edilerek, özellikle V1.0 maliyetinde ve V2.0 yeteneklerinde, genişletme bağlantıları ile birçok ek algılayıcı ve anahtarlama elemanı bağlantısı yapılabilen bir ürün kimliğine kavuşmuştur.

İlerleyen kısımlarda bu materyallerin detaylı tanımlamaları yapılacak olup, gelinen son durumda V1.1 özelinde, TEMEL SEVİYE ve İLERİ SEVİYE olmak üzere iki aşamada, yaklaşık 100 farklı örnek BLOK ve KOD(C) tabanlı çözümleme ile uygulanacaktır.

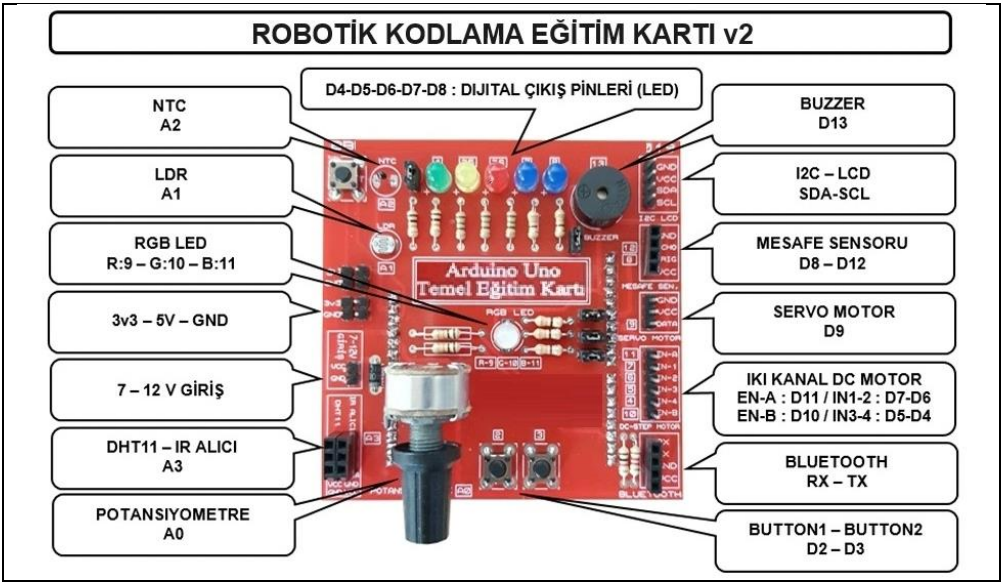


V1 - Birinci Nesil (İlk Üretim)

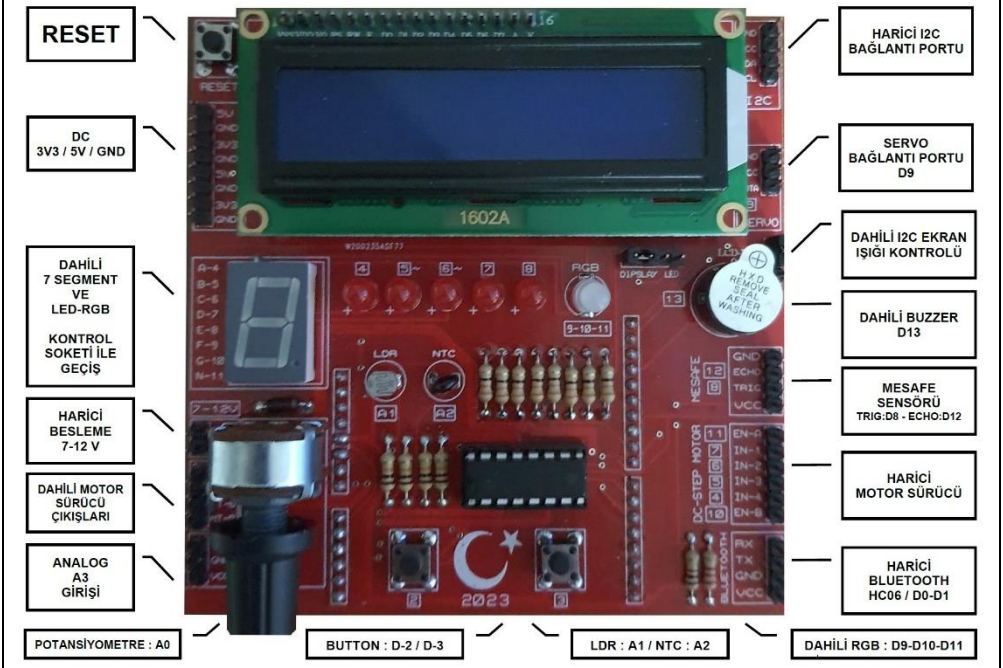


V1 - İkinci Nesil

Şekilde belirtildiği üzere, düşük maliyet, kolay montaj ve kullanıcı dostu hızlı programlama eğitimi uygulamalarına uygun fakat geliştirme bağlantıları olmayan V1.0 eğitim kartı, okul öncesi, ilk okul ve orta okul seviyeleri için, ilgi çekici ve kolay uygulama imkanları sebebiyle ideal bir eğitim materyalidir.



V2 - Birinci Nesil



V3 - Birinci Nesil

Daha sonra geliştirilen V2.0, V1.0'a ek olarak harici bağlantı genişletme portları ile genişletme çeşitliliğine sahip, tam olarak okul öncesi ile üniversite aralığında her tür robotik kodlama seviyesine uygun kullanımda bir üründür. Tüm harici algılayıcı ve anahtarlama elemanları bağımsız olarak her tür kaynaktan temin edilebilmekte ve gerekli uygulamalarda, standart bağlantı noktalarına takılarak, gerekli yazılım geliştirme sürecine hızlı bir şekilde devam edilebilmektedir

Birçok donanımsal programlama bileşenini, başta LCD ve 7SEG modüllerini dahili olarak sunan V3.0, bireysel kullanım, lise veya üniversite seviyesi robotik kodlama eğitimlerinde daha ideal bir çözüm sunmaktadır.

Bu kılavuz, V2-Birinci Nesil ve V3-Birinci Nesil model robotik kodlama eğitim kartı uygulamaları üzerine hazırlanmıştır.

C.3. İç Yapısı ve Bileşenleri

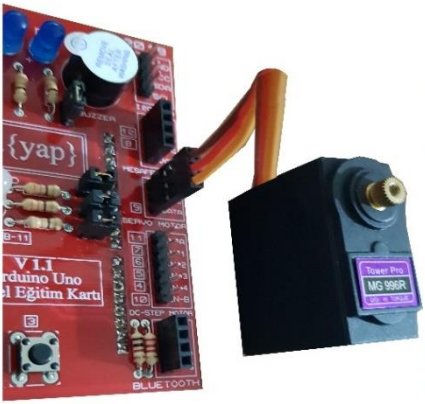
Digital "4-5-6-7-8" bağlantı noktasında, dahili 5 adet LED
Digital "9-10-11" bağlantı noktasında, dahili RGB-LED
Digital "8-12" bağlantı noktasında, genişletme soketi ile Ultrasonic Mesafe Algılayıcı (HC-SR04) bağlantısı
Digital "13" bağlantı noktasında, dahili BUZZER
Digital "9" bağlantı noktasında, genişletme soketi SERVO Motor bağlantısı
Digital "4-5-6-7-10-11" bağlantı noktasında, V2'de genişletme soketi ile, V3'te dahili ve genişletme soketi ile, iki kanal DC Motor Sürücü
Digital "4-5-6-7-8-9-10-11" bağlantı noktasında, V3'te dahili ortak katot 7 segment display
Digital "0-1" bağlantı noktasında, genişletme soketi ile Bluetooth (HC-06) bağlantısı
Digital "2-3" bağlantı noktasında, dahili BUTTON

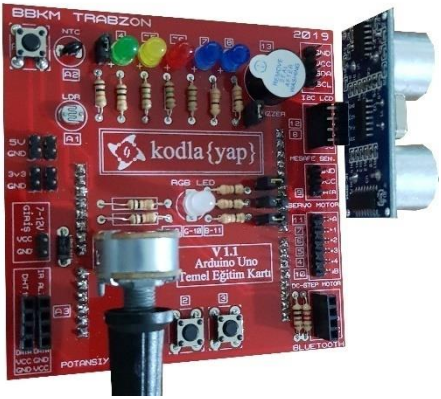
I2C bağlantı noktasında, V2'de genişletme soketi ile, V3'de dahili LCD

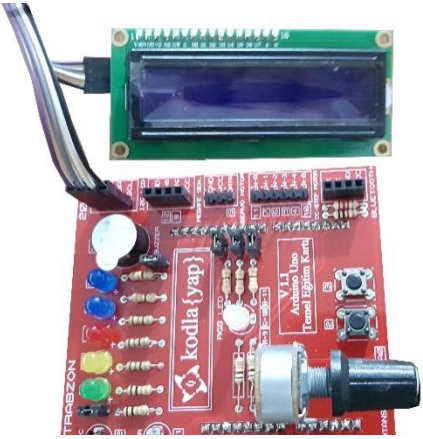
Analog "A0" bağlantı noktasında, dahili POTANSIYOMETRE
Analog "A1" bağlantı noktasında, dahili LDR (IŞIK Algılayıcı)
Analog "A2" bağlantı noktasında, dahili NTC (ISI Algılayıcı)
Analog "A3" bağlantı noktasında, genişletme soketi ile DHT11 (ISI-NEM Algılayıcı) veya IR (Kızıl Ötesi) bağlantısı

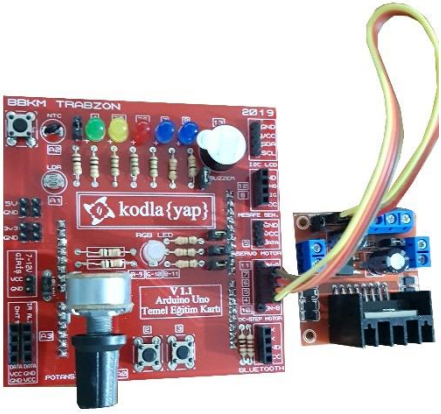
Harici 3v3–5V–GND bağlantı noktaları
Harici 7–12 V besleme girişi bağlantı noktası

C.4. Genişletme Elemanları









D. ARDUINO EĞİTİM SETİ UYGULAMALARI (TEMEL SEVİYE)

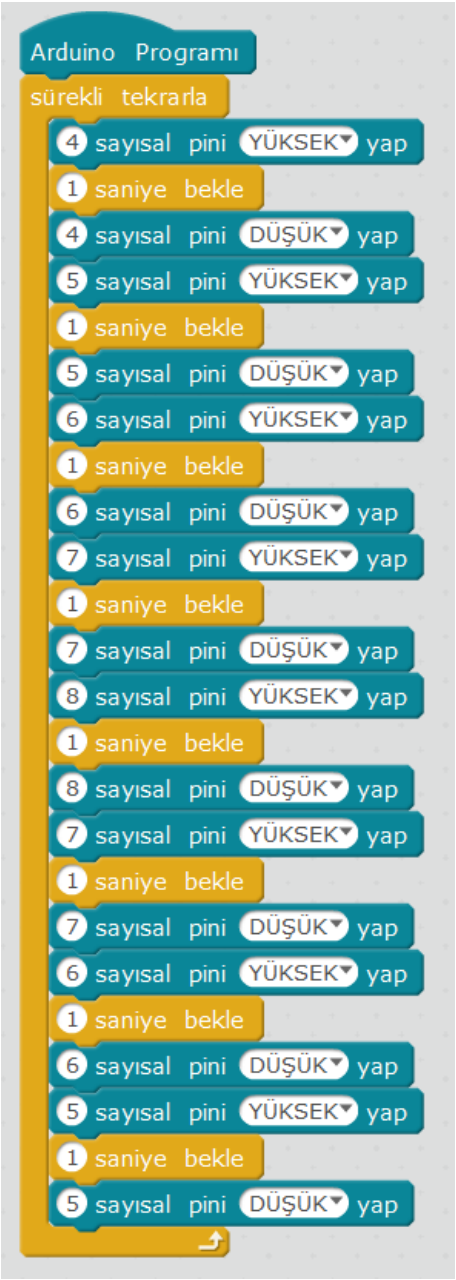
D.1. LED Uygulamaları

1. Bir LED Yak-Söndür (Blink)

2. İki LED Yak-Söndür (Flip-Flop)

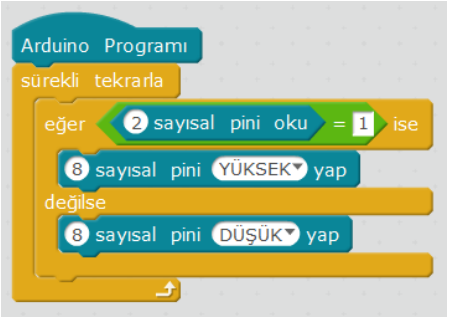
3. Yürüyen Işık Uygulaması (Eklenerek Yanan LED-ler)

4. Kara Şimşek Uygulaması (Tek Tek Yanarak Sağa-Sola İlerleme)

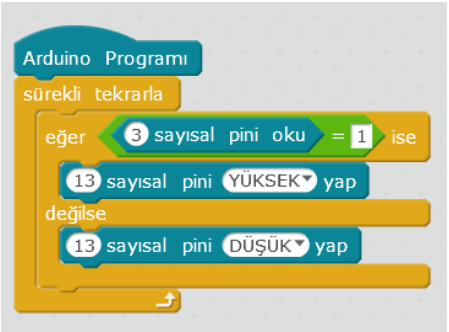


D.2. BUTTON Uygulamaları

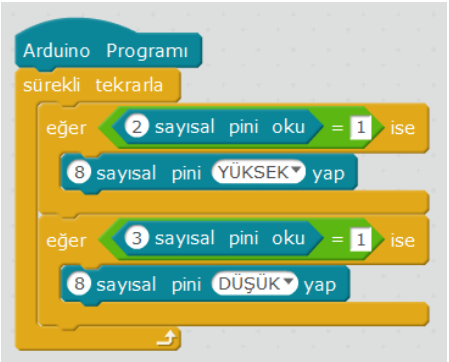
5. Bir BUTTONİle LED Yakma (Bırakınca Sönen)



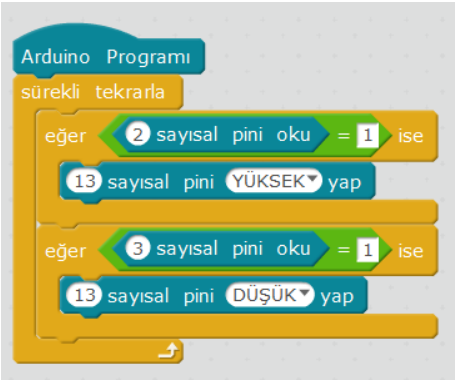
6. Bir BUTTONİle BUZZERÇalıştırma (Bırakınca Duran)



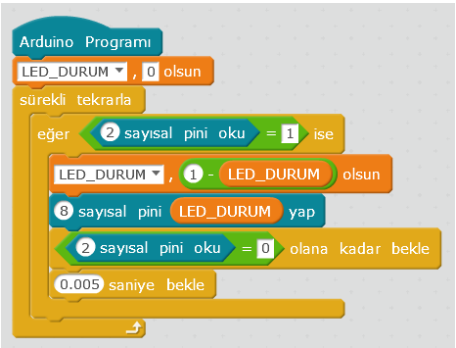
7. Bir BUTTONİle Yakılan LED-in İkinci BUTTONİle Söndürülmesi



8. Bir BUTTONİle BUZZERÇalıştırma, İkinci BUTTONİle Durdurulması



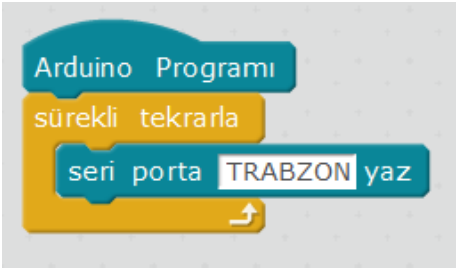
9. Bir BUTTONİle Yakılan LED-in, Aynı BUTTONİle Söndürülmesi



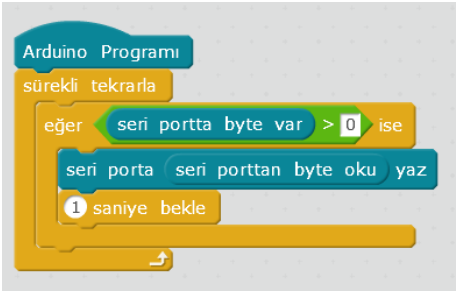
10. Bir BUTTONİle BUZZERÇalıştırma, Aynı BUTTONİle Durdurulması

D.3. SERİ İLETİŞİM Uygulamaları

11. Serial Monitor Giriş-Çıkış Uygulaması (PRINT)

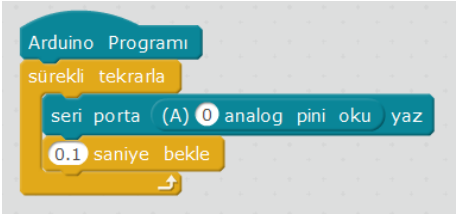


12. Serial Monitor Giriş-Çıkış Uygulaması (READ)



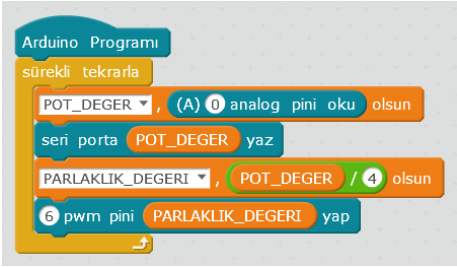
D.4. POTANSIYOMETRE Uygulamaları

13. POTANSIYOMETRE İle Okunan Analog Değerin Yazdırılması

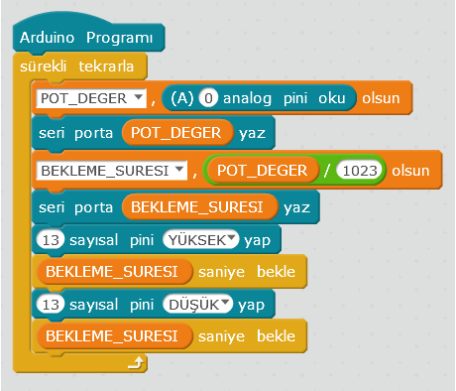


14. POTANSIYOMETRE İle LED Yanıp-Sönme Hızının Değiştirilmesi

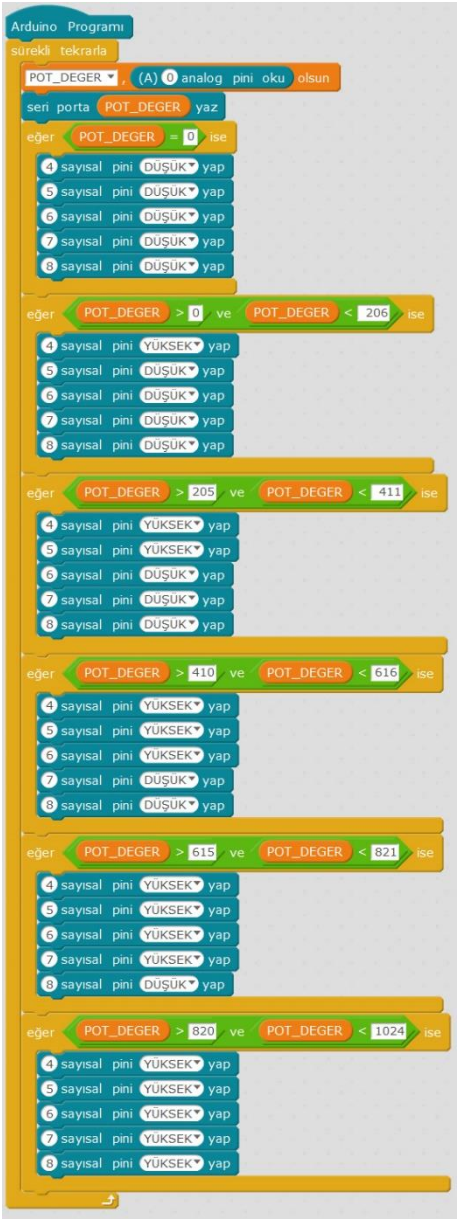
15. POTANSIYOMETRE İle Analog Çıkış LED Parlaklık Ayarı



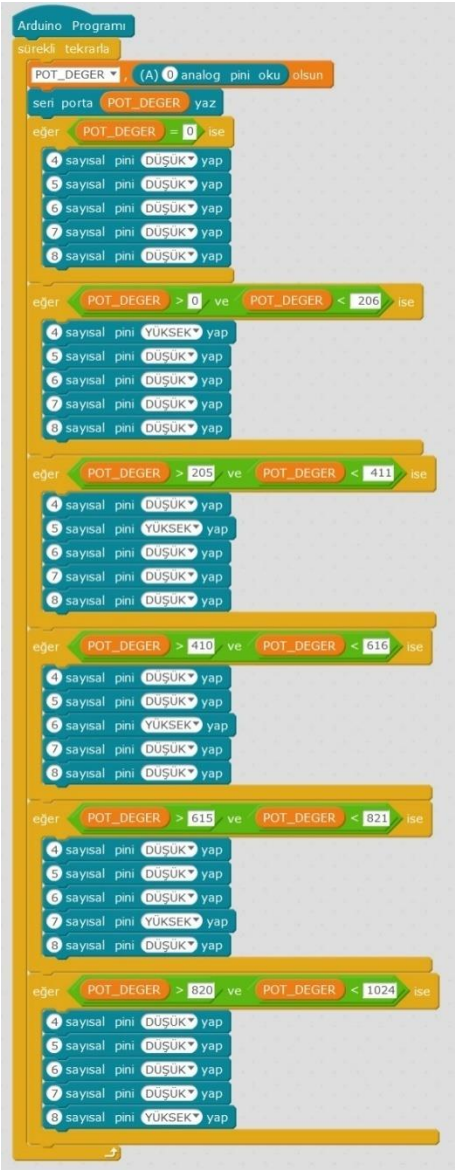
16. POTANSIYOMETRE İle BUZZERÇalışma Zaman Aralığının Değiştirilmesi



17. POTANSIYOMETRE İle LED Seviyesi (Eklenerek Yanan)

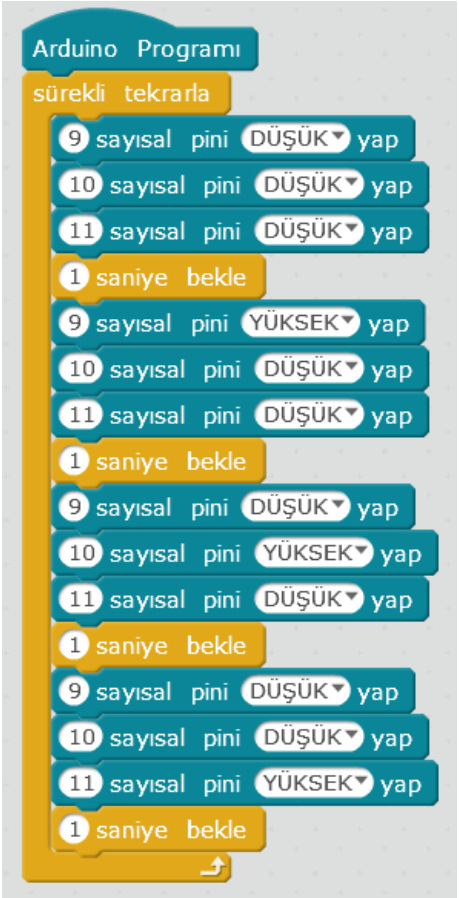


18. POTANSIYOMETRE İle Tek Tek Yanarak Sağa-Sola İlerleyen LED



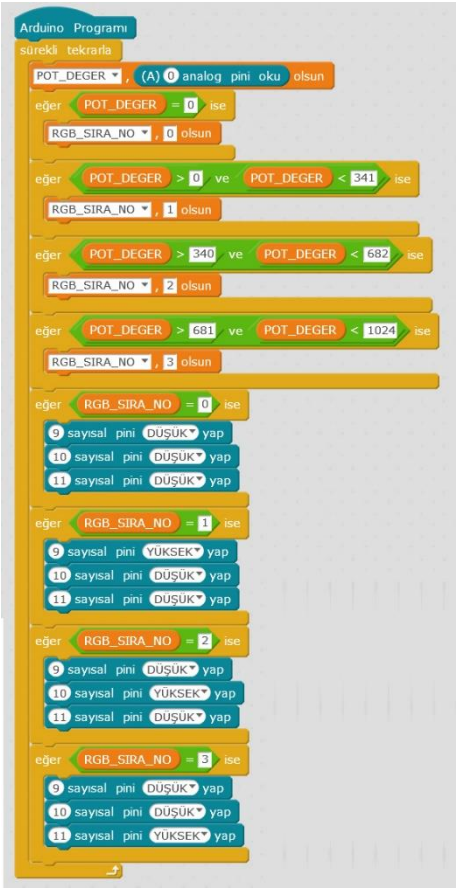
D.5. RGB-LED Uygulamaları

19. Sıra İle R-G-B 3 Ve 7 Renk Ayrı Ayrı Yakma

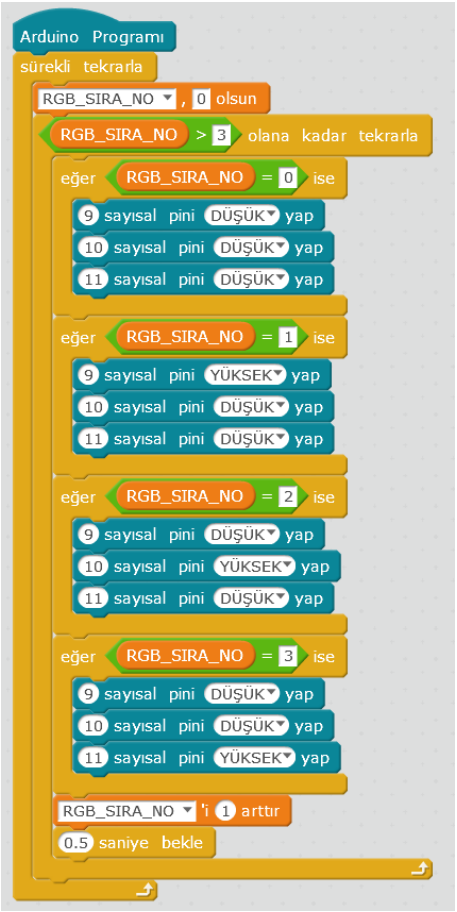


20. BUTTON Kontrollü R-G-B 3 Ve 7 Renk Ayrı Ayrı Yakma

21. POTANSIYOMETRE İle R-G-B 3 Ve 7 Renk Ayrı Ayrı Yakma



22. Döngü İle R-G-B 3 Ve 7 Renk Ayrı Ayrı Yakma

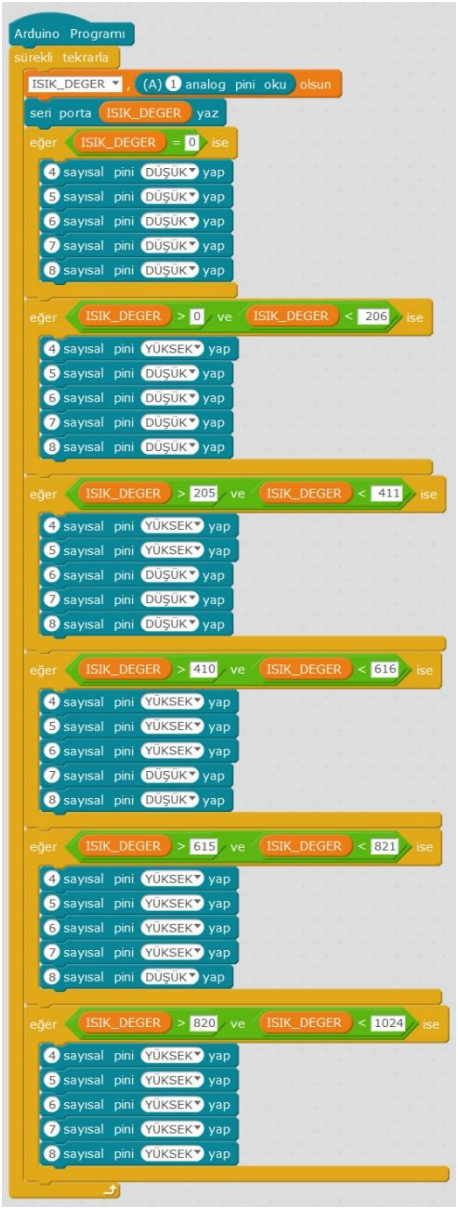


D.6. ISI ve IŞIK Uygulamaları

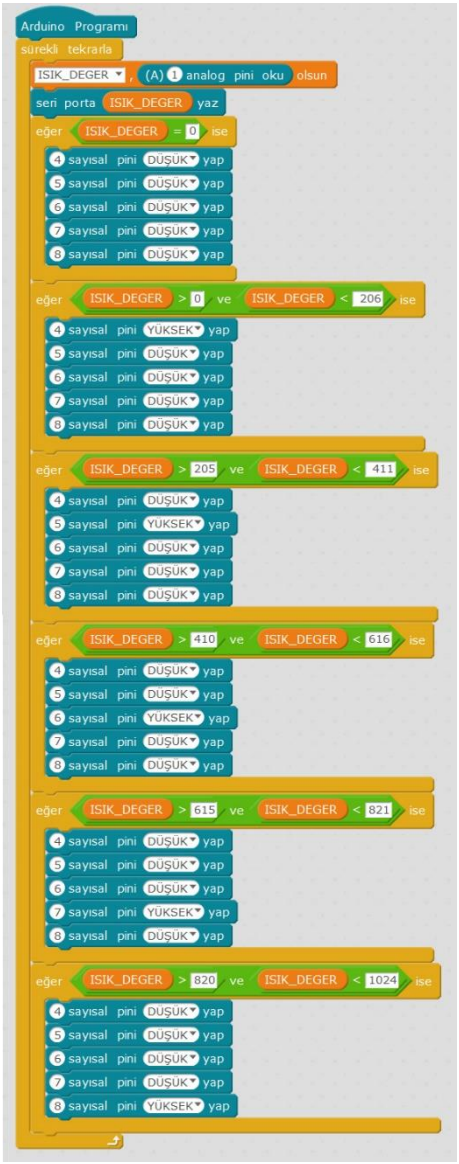
23. LDR İle Analog Işık Şiddeti Değerini Serial Monitorde Yazdırma

24. LDR İle Işık Şiddetine Bağlı LED Kontrolü (Yak-Söndür)

25. LDR İle Işık Şiddetine Bağlı LED Kontrolü (Eklenerek Yanan LED)

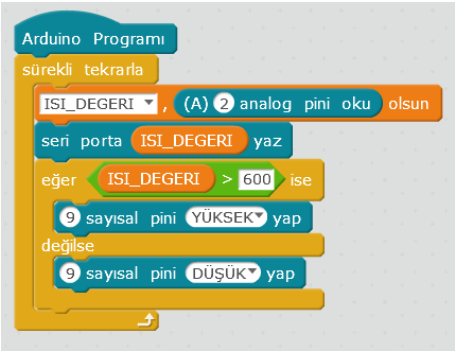


26. LDR İle Işık Şiddetine Bağlı LED Kontrolü (Tek Tek Sağa-Sola İlerleme)



27. NTC İle Analog Isı Değerini Serial Monitorde Yazdırma

28. NTC İle Sıcaklığa Bağlı LED Kontrolü (Yak-Söndür)



E. ARDUINO EĞİTİM SETİ UYGULAMALARI (İLERİ SEVİYE)

E.1. Arduino IDE Temel Uygulamaları

1. Led Yak-Södür Uygulaması

```
void setup() {

  pinMode(9,OUTPUT);
}

void loop() {

  digitalWrite(9, HIGH);          // digitalWrite(9, 1);
  delay(500);
  digitalWrite(9, LOW);           // digitalWrite(9, 0);
  delay(500);
}
```

2. Sıralı Led Yak-Södür Uygulaması

```
void setup() {

  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);

}

void loop() {

  digitalWrite(4, 1);
  delay(500);
  digitalWrite(4, 0);
  delay(500);

  digitalWrite(5, 1);
  delay(500);
  digitalWrite(5, 0);
  delay(500);

  digitalWrite(6, 1);
  delay(500);
  digitalWrite(6, 0);
  delay(500);

  digitalWrite(7, 1);
  delay(500);
  digitalWrite(7, 0);
  delay(500);

  digitalWrite(8, 1);
  delay(500);
  digitalWrite(8, 0);
  delay(500);
}
```

3. Döngü İle (FOR) Sıralı Led Yak-Södür Uygulaması

```
void setup() {

  for (int I=4; I<12; I++){
    pinMode(I,OUTPUT);
  }
}

void loop() {

  for (int I=4; I<12; I++){
    digitalWrite(I, 1);
    delay(500);
    digitalWrite(I, 0);
  }
}
```

4. Döngü İle (WHILE) Sıralı Led Yak-Södür Uygulaması

```
void setup() {

  int I = 4;
  while (I<12){
    pinMode(I,OUTPUT);
    I=I+1;              // I++;
  }
}

void loop() {

  int I = 4;
  while (I<12){
    digitalWrite(I, 1);
    delay(500);
    digitalWrite(I, 0);

    I++;
  }
}
```

5. Serial Monitör Uygulaması

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println("Bu bir deneme mesajidir...");
  delay(500);
}
```

```
void setup() {

  Serial.begin(9600);
}

void loop() {

  for (int I=0; I<10; I++)
    Serial.print(I);

  Serial.println();

  delay(500);
}
```

```
void setup() {

  Serial.begin(9600);
}

void loop() {

  if ( Serial.available() ){
    char K = Serial.read();
    Serial.println(K);
  }

  delay(500);
}
```

```
void setup() {

  Serial.begin(9600);
}

void loop() {

  if ( Serial.available() ){
    String S = Serial.readString();
    Serial.println(S);
  }

  delay(500);
}
```

```
void setup() {

  Serial.begin(9600);
  pinMode(9, OUTPUT);
}

void loop() {

  if ( Serial.available() ){
    char K = Serial.read();

    if ( K == 'A' )
      digitalWrite(9, 1);
    if ( K == 'B' )
      digitalWrite(9, 0);
  }

  delay(500);
}
```

6. Analog Veri Okuma (Potansiyometre-LDR-NTC) Uygulaması

```
void setup() {

  Serial.begin(9600);

  pinMode(A0,INPUT);     // POTANSIYOMETRE
  pinMode(A1,INPUT);     // LDR (ISIK)
  pinMode(A2,INPUT);     // NTC (SICAKLIK)
}

void loop() {

  Serial.print("POTANSIYOMETRE : ");
  Serial.println(analogRead(A0));
  Serial.print("LDR (ISIK)     : ");
  Serial.println(analogRead(A1));
  Serial.print("NTC (SICAKLIK) : ");
  Serial.println(analogRead(A2));

  delay(500);
  Serial.println();
}
```

7. Analog Çıkış (Potansiyometre ve MAP) LED Parlaklık Uygulaması

```
void setup() {

  Serial.begin(9600);

  pinMode(A0,INPUT);                    // POTANSIYOMETRE
  pinMode(6, OUTPUT);
}

void loop() {

  Serial.print("POTANSIYOMETRE : ");
  Serial.println(analogRead(A0));

  int M = map(analogRead(A0), 0,1023, 0, 255);

  Serial.println(M);

  analogWrite(6, M);
}
```

8. IF-ELSE (Potansiyometre-LDR-NTC) Kontrol Uygulaması

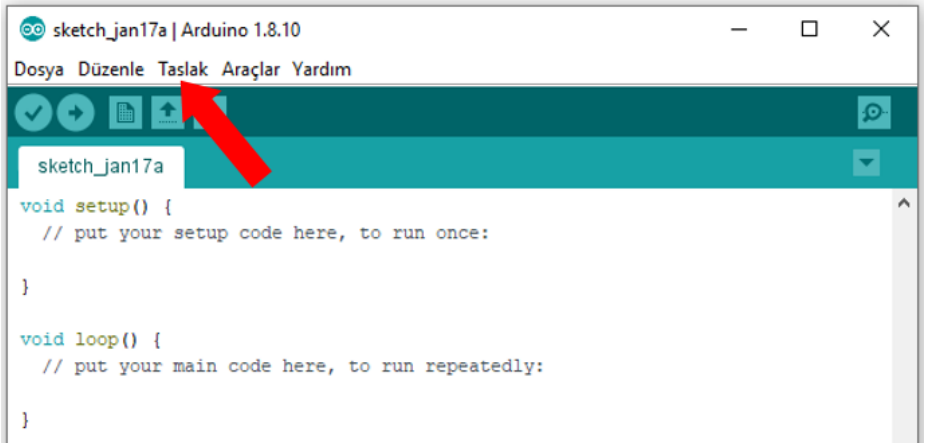
```
void setup() {

  Serial.begin(9600);

  pinMode(A0,INPUT);       // POTANSIYOMETRE
  pinMode(A1,INPUT);       // LDR (ISIK)
  pinMode(A2,INPUT);       // NTC (SICAKLIK)

  pinMode(6, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(13, OUTPUT);     // BUZZER
}

void loop() {

  Serial.print("POTANSIYOMETRE : ");
  Serial.println(analogRead(A0));

  Serial.print("LDR (ISIK)     : ");
  Serial.println(analogRead(A1));

  Serial.print("NTC (SICAKLIK) : ");
  Serial.println(analogRead(A2));

  Serial.println();

  if (analogRead(A0) > 800) digitalWrite(13, 1);
  else digitalWrite(13, 0);

  if (analogRead(A1) < 250) digitalWrite(8, 1);
  else digitalWrite(8, 0);

  if (analogRead(A2) > 550) digitalWrite(6, 1);
  else digitalWrite(6, 0);

  delay(500);
}
```

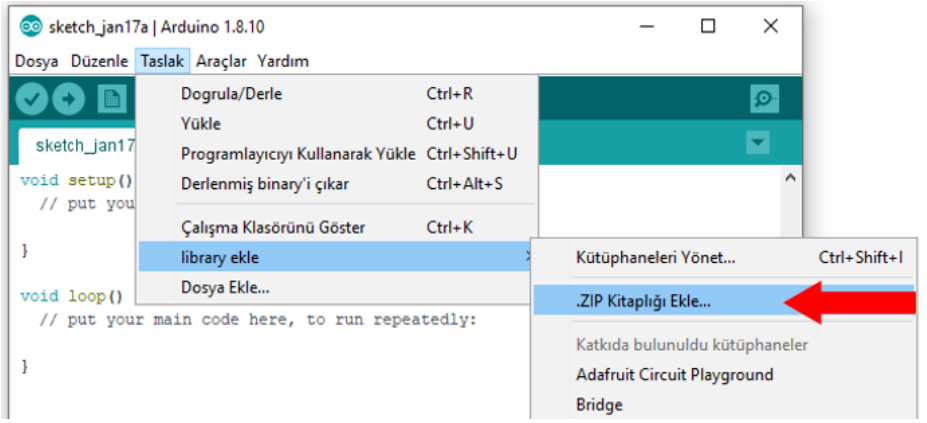
9. Potansiyometre ve MAP Tekniği ile Ayarlı LED Sıralama Uygulaması

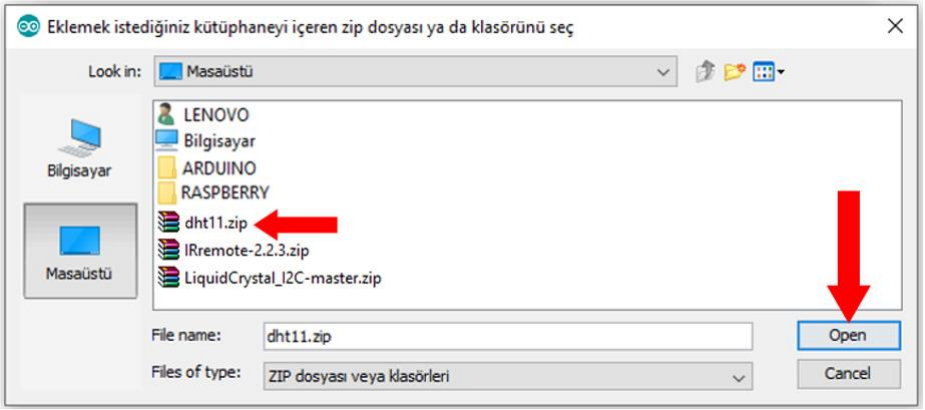
```
void setup() {
  Serial.begin(9600);

  pinMode(A0,INPUT);                    // POTANSIYOMETRE
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}
void loop() {
  Serial.print("POTANSIYOMETRE : ");
  Serial.println(analogRead(A0));

  int L = map(analogRead(A0), 0,1023, 4, 8);

  Serial.println(L);

  for (int I=4;I<9;I++)
    digitalWrite(I,0);

  digitalWrite(L,1);
}
```

```
void setup() {
  Serial.begin(9600);

  pinMode(A0,INPUT);                    // POTANSIYOMETRE
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}
void loop() {
  Serial.print("POTANSIYOMETRE : ");
  Serial.println(analogRead(A0));

  int L = map(analogRead(A0), 0, 1023, 4, 8);

  Serial.print(analogRead(A0));
  Serial.print(", ");
  Serial.println(L);

  for ( int I=4; I<9; I++ )
    if ( I <= L ) digitalWrite(I, 1);
    else digitalWrite(I, 0);
}
```

10. Kütüphane Yükleme ve DHT11 Sıcaklık-Nem Bilgisini Okuma Uygulaması

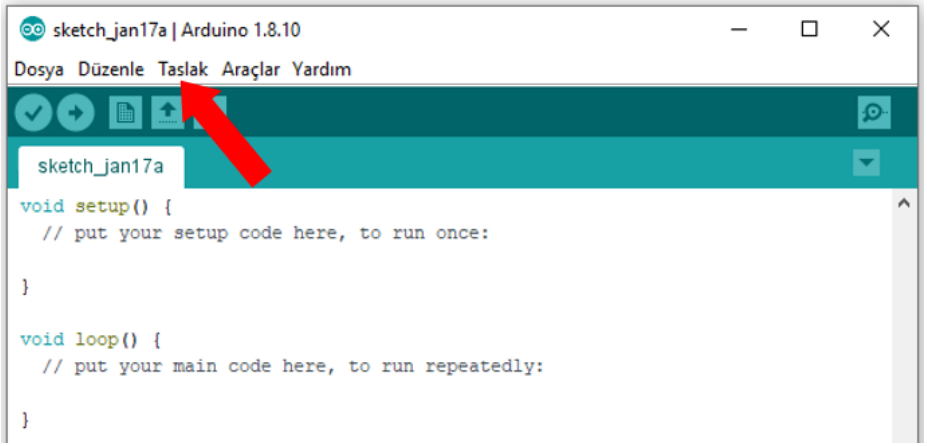


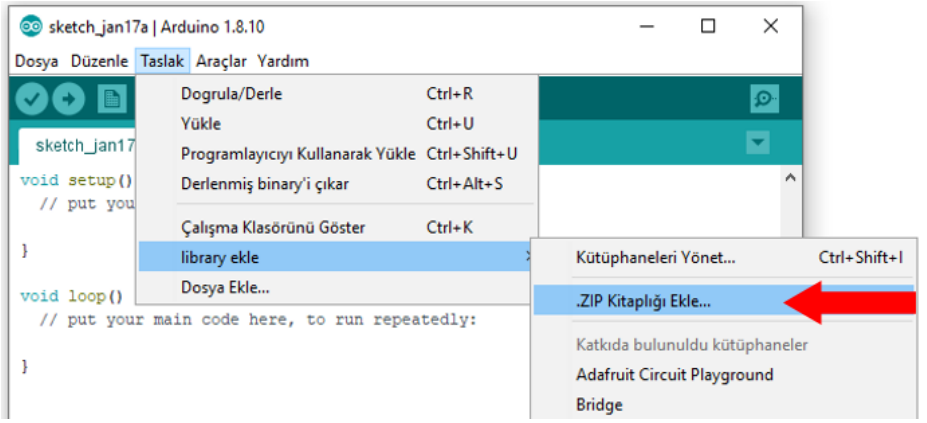


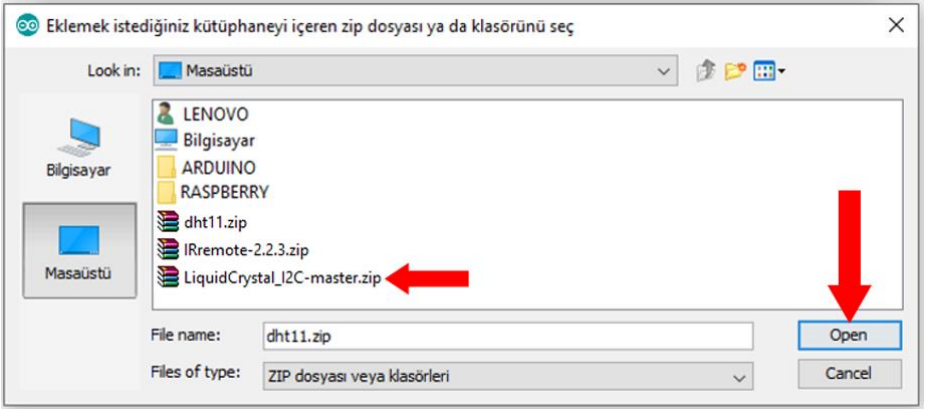


```
#include <dht11.h>
#define DHT11PIN A3

dht11 DHT11;

void setup(){
  Serial.begin(9600);
  Serial.println("DHT11 Test");
}

void loop(){
  Serial.println();

  int chk = DHT11.read(DHT11PIN);   // chk 0 ise sorun YOK...

  Serial.print("NEM      (%): ");
  Serial.println((float)DHT11.humidity, 2);

  Serial.print("SICAKLIK (C): ");
  Serial.println((float)DHT11.temperature, 2);

  Serial.print("SICAKLIK (F): ");
  Serial.println(DHT11.fahrenheit(), 2);

  Serial.print("SICAKLIK (K): ");
  Serial.println(DHT11.kelvin(), 2);

  delay(1000);
}
```

E.2. 16x2 I2C-LCD Ekran Uygulamaları







11. Tek Satır ve İki Satır Çıktı Uygulaması

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C EKRAN(0x27,16,2);           // 0x3F

void setup(){
  EKRAN.init();
  EKRAN.backlight();
  EKRAN.setCursor(0,0);                       // (SUTUN,SATIR)
  EKRAN.print("MERHABA DUNYA");
  EKRAN.setCursor(0,1);                       // (SUTUN,SATIR)
  EKRAN.print("BEN ARDUINO...");
}

void loop(){

}
```

12. Tek Satırlı Kayan Yazı Uygulaması

```
#include <LiquidCrystal_I2C.h>                                    -I-
LiquidCrystal_I2C EKRAN(0x27,16,2);           // 0x3F

int BEKLEME_SURESI = 500;
String METIN = "KAYAN YAZI UYGULAMASI... ";

void setup(){
  EKRAN.init();
  EKRAN.backlight();
}

void loop(){
  for (int SUTUN = 0; SUTUN<17; SUTUN++){
    EKRAN.clear();
    EKRAN.setCursor(SUTUN,0);                 // (SUTUN,SATIR)
    EKRAN.print(METIN);
    delay(BEKLEME_SURESI);
  }

  for (int SUTUN = 15; SUTUN>0; SUTUN--){
    EKRAN.clear();
    EKRAN.setCursor(SUTUN,0);                 // (SUTUN,SATIR)
    EKRAN.print(METIN);
    delay(BEKLEME_SURESI);
  }

  int KARAKTER_SAYISI = METIN.length();

  for (int K=0; K<KARAKTER_SAYISI; K++){
    String AltMETIN = "";
    for (int K2=K; K2<KARAKTER_SAYISI; K2++)
      AltMETIN += METIN[K2];
    EKRAN.clear();
    EKRAN.setCursor(0,0);                     // (SUTUN,SATIR)
    EKRAN.print(AltMETIN);
    delay(BEKLEME_SURESI);
  }

  for (int K=KARAKTER_SAYISI-1; K>0; K--){
    String AltMETIN = "";
    for (int K2=K; K2<KARAKTER_SAYISI; K2++)
      AltMETIN += METIN[K2];
    EKRAN.clear();
    EKRAN.setCursor(0,0);                     // (SUTUN,SATIR)
    EKRAN.print(AltMETIN);
    delay(BEKLEME_SURESI);
  }
}
```

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);           // 0x3F
```

-II-

```
String METIN = "KAYAN YAZI UYGULAMASI... ";

void setup(){
  Serial.begin(9600);
  EKRAN.init();
  EKRAN.backlight();
}

void loop(){
  int BEKLEME_SURESI = 500;
  int KARAKTER_SAYISI = METIN.length();
  int DIZI_BOYUTU = KARAKTER_SAYISI*2+16;
  char DIZI[DIZI_BOYUTU];
  for (int I=0; I<DIZI_BOYUTU; I++)
    DIZI[I] = '-';
  for (int I=0; I<KARAKTER_SAYISI; I++)
    DIZI[I] = METIN[I];
  for (int I=0; I<DIZI_BOYUTU; I++)
    Serial.print(DIZI[I]);
  Serial.println();
  for (int SAYAC=0; SAYAC<KARAKTER_SAYISI+16; SAYAC++) {
    for (int D=DIZI_BOYUTU-2; D>-1; D--)
      DIZI[D+1] = DIZI[D];
    DIZI[0]  = '-';
    for (int I=0; I<DIZI_BOYUTU; I++)
        Serial.print(DIZI[I]);
    EKRAN.setCursor(0,0);
    for (int S=20; S<36; S++)
      EKRAN.print(DIZI[S]);
    Serial.println();
    delay(500);
  }

  for (int SAYAC=0; SAYAC<KARAKTER_SAYISI+15; SAYAC++){
    for (int D=1; D<DIZI_BOYUTU; D++)
      DIZI[D-1] = DIZI[D];
    DIZI[DIZI_BOYUTU-1]  = '-';
    for (int I=0; I<DIZI_BOYUTU; I++)
        Serial.print(DIZI[I]);
    EKRAN.setCursor(0,0);
    for (int S=20; S<36; S++)
      EKRAN.print(DIZI[S]);
    Serial.println();
    delay(500);
  }
}
```

13. İki Satırlı Kayan Yazı Uygulaması

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);           // 0x3F
String METIN = "KAYAN YAZI UYGULAMASI...";
void setup(){
  EKRAN.init();
  EKRAN.backlight();
}
void loop(){
  int BEKLEME_SURESI = 500;
  int KARAKTER_SAYISI = METIN.length();
  String AltMETIN1;
  int K1;
  for (int I=16; I>0; I--){
    EKRAN.clear();
    EKRAN.setCursor(I,0);                     // (SUTUN,SATIR)
    EKRAN.print(METIN);
    int T = 16-I;
    AltMETIN1 = "";
    for (K1=KARAKTER_SAYISI-T; K1<KARAKTER_SAYISI; K1++)
      if (K1<0) AltMETIN1 += " ";
      else AltMETIN1 += METIN[K1];
    EKRAN.setCursor(0,1);                     // (SUTUN,SATIR)
    EKRAN.print(AltMETIN1);
    delay(BEKLEME_SURESI);
    K1=KARAKTER_SAYISI-T-1;
  }
  for (int K=0; K<KARAKTER_SAYISI; K++){
    String AltMETIN2 = "";
    for (int K2=K; K2<KARAKTER_SAYISI; K2++)
      AltMETIN2 += METIN[K2];
    EKRAN.clear();
    EKRAN.setCursor(0,0);                     // (SUTUN,SATIR)
    EKRAN.print(AltMETIN2);
    EKRAN.setCursor(0,1);                     // (SUTUN,SATIR)
    EKRAN.print(K1);
    int EK1 = K1;
    AltMETIN1 = "";
    while (K1<KARAKTER_SAYISI){
      if (K1<0) AltMETIN1 += " ";
      else AltMETIN1 += METIN[K1];
      K1++;
    }
    K1 = EK1 - 1;
    EKRAN.setCursor(0,1);                     // (SUTUN,SATIR)
    EKRAN.print(AltMETIN1);
    delay(BEKLEME_SURESI);
  }
}
```

14. Analog Veri (Potansiyometre-LDR-NTC) 16x2 I2C-LCD Ekranda Yazdırma

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);

int BEKLEME_SURESI = 500;

void setup(){
  EKRAN.init();
  EKRAN.backlight();
}

void loop(){
  int POT = analogRead(A0);
  int LDR = analogRead(A1);
  int NTC = analogRead(A2);

  EKRAN.clear();

  EKRAN.setCursor(0,0);

EKRAN.print("POTANSIYOMETRE");

  EKRAN.setCursor(0,1);
  EKRAN.print(POT);

  delay(BEKLEME_SURESI);

  EKRAN.clear();

  EKRAN.setCursor(0,0);
  EKRAN.print("LDR");

  EKRAN.setCursor(0,1);
  EKRAN.print(LDR);

  delay(BEKLEME_SURESI);

  EKRAN.clear();

  EKRAN.setCursor(0,0);
  EKRAN.print("NTC");

  EKRAN.setCursor(0,1);
  EKRAN.print(NTC);

  delay(BEKLEME_SURESI);
}
```

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);

int BEKLEME_SURESI = 500;

void setup() {
  EKRAN.init();
  EKRAN.backlight();
}

void loop() {
  int POT = analogRead(A0);
  int LDR = analogRead(A1);
  int NTC = analogRead(A2);

  EKRAN.clear();

  EKRAN.setCursor(0,0);
  EKRAN.print("POT__LDR__NTC");

  EKRAN.setCursor(0,1);
  EKRAN.print(POT);
  EKRAN.setCursor(6,1);
  EKRAN.print(LDR);
  EKRAN.setCursor(12,1);
  EKRAN.print(NTC);

  delay(BEKLEME_SURESI);
}
```

15. DHT11 ile Sıcaklık Ve Nem Bilgisini 16x2 I2C-LCD Ekranda Yazdırma

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);

#include <dht11.h>
#define DHT11PIN A3

dht11 DHT11;

void setup(){
  EKRAN.init();
  EKRAN.backlight();
}

void loop(){
  int chk = DHT11.read(DHT11PIN);

  EKRAN.clear();
  EKRAN.setCursor(0,0);
  EKRAN.print("NEM  (%): ");
  EKRAN.print((float)DHT11.humidity, 2);

  EKRAN.setCursor(0,1);
  EKRAN.print("SCKL (C): ");
  EKRAN.print((float)DHT11.temperature, 2);

  delay(1000);

  EKRAN.clear();
  EKRAN.setCursor(0,0);
  EKRAN.print("SCKL (F): ");
  EKRAN.print(DHT11.fahrenheit(), 2);

  EKRAN.setCursor(0,1);
  EKRAN.print("SCKL (K): ");
  EKRAN.print(DHT11.kelvin(), 2);

  delay(1000);
}
```

E.3. Ultrasonic Mesafe Algılayıcı Uygulamaları

16. Mesafe Okuma Ve Serial Monitörde Yazdırma Uygulaması

```
const int TRIG = 8;
const int ECHO = 12;

long GECIKME;
float MESAFE;

void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

  Serial.begin(9600);
}

void loop() {
  digitalWrite(TRIG, 0);
  delayMicroseconds(2);

  digitalWrite(TRIG, 1);
  delayMicroseconds(10);
  digitalWrite(TRIG, 0);

  GECIKME = pulseIn(ECHO, 1);

  MESAFE = GECIKME * 0.034 / 2;

  Serial.print("MESAFE :  ");
  Serial.print(MESAFE);
  Serial.println(" CM");
}
```

17. Mesafe Okuma Ve I2C-LCDEkranda Yazdırma Uygulaması

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);

const int TRIG = 8;
const int ECHO = 12;

long GECIKME;
float MESAFE;

void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

  EKRAN.init();
  EKRAN.backlight();
}

void loop() {
  digitalWrite(TRIG, 0);
  delayMicroseconds(2);
  digitalWrite(TRIG, 1);
  delayMicroseconds(10);
  digitalWrite(TRIG, 0);

  GECIKME = pulseIn(ECHO, 1);

  MESAFE = GECIKME * 0.034 / 2;

  EKRAN.clear();
  EKRAN.setCursor(0,0);
  EKRAN.print("MESAFE(CM):");
  EKRAN.print(MESAFE);

  EKRAN.setCursor(0,1);
  EKRAN.print("MESAFE(MT):");
  EKRAN.print(MESAFE/100);

  delay(100);
}
```

18. Mesafe Değerine Bağlı BUZZER Park Sensoru Uygulaması

-I-

```
const int TRIG = 8;
const int ECHO = 12;
const int BUZZER = 13;

long GECIKME;
float MESAFE;

void setup() {
  Serial.begin(9600);

  pinMode(BUZZER, OUTPUT);
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
}

void loop() {
  digitalWrite(TRIG, 0);
  delayMicroseconds(2);
  digitalWrite(TRIG, 1);
  delayMicroseconds(10);
  digitalWrite(TRIG, 0);

  GECIKME = pulseIn(ECHO, 1);

  MESAFE = GECIKME * 0.034 / 2;

  Serial.print("MESAFE-CM:");
  Serial.println(MESAFE);
  Serial.print("MESAFE-MT: ");
  Serial.println(MESAFE/100);
  Serial.println();

  if (MESAFE < 100) {
    digitalWrite(BUZZER, 1);
    delay(MESAFE*10);
    digitalWrite(BUZZER, 0);
  }
}
```

-II-

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);

const int TRIG = 8;
const int ECHO = 12;
const int BUZZER = 13;

long GECIKME;
float MESAFE;

void setup() {
  pinMode(BUZZER, OUTPUT);
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  EKRAN.init();
  EKRAN.backlight();
}

void loop() {
  digitalWrite(TRIG, 0);
  delayMicroseconds(2);
  digitalWrite(TRIG, 1);
  delayMicroseconds(10);
  digitalWrite(TRIG, 0);

  GECIKME = pulseIn(ECHO, 1);
  MESAFE = GECIKME * 0.034 / 2;

  EKRAN.clear();
  EKRAN.setCursor(0,0);
  EKRAN.print("MESAFE-CM: ");
  EKRAN.print(MESAFE);
  EKRAN.setCursor(0,1);
  EKRAN.print("MESAFE-MT: ");
  EKRAN.print(MESAFE/100);

  if (MESAFE < 100) {
    digitalWrite(BUZZER, 1);
    if (MESAFE > 10) {
      delay(MESAFE*10);
      digitalWrite(BUZZER, 0);
    }
  }
  if (MESAFE > 100)
    digitalWrite(BUZZER, 0);
}
```

19. Mesafe Okuma Ve I2C-LCDEkranda GöstergeYazdırma Uygulaması

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);

const int TRIG = 8;
const int ECHO = 12;

const int BUZZER = 13;

long GECIKME;
float MESAFE;

void setup() {
  pinMode(BUZZER, OUTPUT);

  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

  EKRAN.init();
  EKRAN.backlight();
}

void loop() {
  digitalWrite(TRIG, 0);
  delayMicroseconds(2);

  digitalWrite(TRIG, 1);
  delayMicroseconds(10);
  digitalWrite(TRIG, 0);

  GECIKME = pulseIn(ECHO, 1);

  MESAFE = GECIKME * 0.034/ 2;

  EKRAN.clear();
  EKRAN.setCursor(0,0);
  EKRAN.print(MESAFE);

  EKRAN.print(" / ");
  EKRAN.print(MESAFE/100);

  int GOSTERGE = 0;
  if ( MESAFE < 10 )
    GOSTERGE = 16;
  else if ( MESAFE < 20 )
    GOSTERGE = 15;
  else if ( MESAFE < 30 )
    GOSTERGE = 14;
  else if ( MESAFE < 40 )
    GOSTERGE = 13;
  else if ( MESAFE < 50 )
    GOSTERGE = 12;
  else if ( MESAFE < 60 )
    GOSTERGE = 11;
  else if ( MESAFE < 70 )
    GOSTERGE = 10;
  else if ( MESAFE < 80 )
    GOSTERGE = 9;
  else if ( MESAFE < 90 )
    GOSTERGE = 8;
  else if ( MESAFE < 100 )
    GOSTERGE = 7;
  else if ( MESAFE < 110 )
    GOSTERGE = 6;
  else if ( MESAFE < 120 )
    GOSTERGE = 5;
  else if ( MESAFE < 130 )
    GOSTERGE = 4;
  else if ( MESAFE < 140 )
    GOSTERGE = 3;
  else if ( MESAFE < 150 )
    GOSTERGE = 2;
  else if ( MESAFE < 160 )
    GOSTERGE = 1;

  EKRAN.setCursor(0,1);

  for (int I=0; I<16; I++)
    if (I<GOSTERGE)
      EKRAN.print("*");
    else
      EKRAN.print(" ");
}
```

-I-

-II-

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C EKRAN(0x27,16,2);

const int TRIG = 8;
const int ECHO = 12;

const int BUZZER = 13;

long GECIKME;
float MESAFE;

void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  pinMode(BUZZER, OUTPUT);

  EKRAN.init();
  EKRAN.backlight();
}
```

```
void loop() {
  digitalWrite(TRIG, 0);
  delayMicroseconds(2);

  digitalWrite(TRIG, 1);
  delayMicroseconds(10);
  digitalWrite(TRIG, 0);

  GECIKME = pulseIn(ECHO, 1);

  MESAFE = GECIKME * 0.034 / 2;

  EKRAN.clear();
  EKRAN.setCursor(0,0);
  EKRAN.print(MESAFE);

  EKRAN.print(" / ");
  EKRAN.print(MESAFE/100);

  int GOSTERGE = 0;

  for (int I=16; I>0; I--)
    if ( MESAFE < I*10 )
      GOSTERGE = 17 - I;
    else break;

  EKRAN.setCursor(0,1);

  for (int I=0; I<16; I++)
    if (I<GOSTERGE)
      EKRAN.print("*");
    else EKRAN.print(" ");

  if (MESAFE < 100){
    digitalWrite(BUZZER, 1);
    if (MESAFE > 10){
      delay(MESAFE*10);
      digitalWrite(BUZZER, 0);
    }
  }

  if (MESAFE > 100)
    digitalWrite(BUZZER, 0);
}
```

20. Mesafe Değerine Bağlı LED Park Sensoru Uygulaması

```
const int TRIG = 8;
const int ECHO = 12;

const int BUZZER = 13;

long GECIKME;
float MESAFE;

void setup(){
  Serial.begin(9600);

  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  pinMode(BUZZER, OUTPUT);

  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}
```

```
void loop(){
  digitalWrite(TRIG, 0);
  delayMicroseconds(2);

  digitalWrite(TRIG, 1);
  delayMicroseconds(10);
  digitalWrite(TRIG, 0);

  GECIKME = pulseIn(ECHO, 1);

  MESAFE = GECIKME * 0.034 / 2;

  Serial.print("MESAFE-CM: ");
  Serial.println(MESAFE);
  Serial.print("MESAFE-MT: ");
  Serial.println(MESAFE/100);
  Serial.println();

  digitalWrite(4, 0);
  digitalWrite(5, 0);
  digitalWrite(6, 0);
  digitalWrite(7, 0);
  digitalWrite(8, 0);

  if ( MESAFE < 100 )
    digitalWrite(4, 1);
  if ( MESAFE < 80 )
    digitalWrite(5, 1);
  if ( MESAFE < 60 )
    digitalWrite(6, 1);
  if ( MESAFE < 40 )
    digitalWrite(7, 1);
  if ( MESAFE < 20 )
    digitalWrite(8, 1);

  if (MESAFE < 100){
    digitalWrite(BUZZER, 1);
    delay(MESAFE*10);
    digitalWrite(BUZZER, 0);
  }
}
```

E.4. Servo Motor Ve Röle Uygulamaları

21. Servo Motor 0º→180º Zaman Gecikmeli Dönme Uygulaması

```
#include <Servo.h>
Servo SERVO_MOTOR;
void setup() {
  SERVO_MOTOR.attach(9);
}
void loop() {
  SERVO_MOTOR.write(0);
  delay(1000);
  SERVO_MOTOR.write(90);
  delay(1000);
  SERVO_MOTOR.write(180);
  delay(1000);
}
```

22. Potansiyometre ile Servo Motor 0º→180º Dönme Uygulaması

```
#include <Servo.h>
Servo SERVO_MOTOR;
int POT = A0;
int POT_DEGER, SERVO_DEGER;
void setup() {
  SERVO_MOTOR.attach(9);
}
void loop() {
  POT_DEGER = analogRead(A0);
  SERVO_DEGER = map(POT_DEGER, 0, 1023, 0, 180);
  SERVO_MOTOR.write(SERVO_DEGER);
  delay(15);
}
```

23. Işık Etkisiile Servo Motor 0º→180º Dönme Uygulaması

```
#include <Servo.h>
Servo SERVO_MOTOR;
int LDR = A1;
int LDR_DEGER, SERVO_DEGER;
void setup() {
  SERVO_MOTOR.attach(9);
}
void loop() {
  LDR_DEGER = analogRead(A1);
  SERVO_DEGER = map(LDR_DEGER, 0, 1023, 0, 180);
  SERVO_MOTOR.write(SERVO_DEGER);
  delay(15);
}
```

24. Isı Etkisiile Servo Motor 0º→180º Dönme Uygulaması

```
#include <Servo.h>
Servo SERVO_MOTOR;
int NTC = A2;
int NTC_DEGER, SERVO_DEGER;
void setup() {
  SERVO_MOTOR.attach(9);
}
void loop() {
  NTC_DEGER = analogRead(A2);
  SERVO_DEGER = map(NTC_DEGER, 0, 1023, 0, 180);
  SERVO_MOTOR.write(SERVO_DEGER);
  delay(15);
}
```

25. Button Kontrollü Röle Anahtarlama Uygulaması

```
void setup() {
  pinMode(9, OUTPUT);
  pinMode(2, INPUT);
  pinMode(3, INPUT);
}

void loop() {
  if ( digitalRead(2) )
    digitalWrite(9, 1);

  if ( digitalRead(3) )
    digitalWrite(9, 0);
}
```

E.5. DC Motor Ve Sürücü Uygulamaları

26. Tek DC Motor Yönlü Döndürme Uygulaması

```
const int EN_SAG = 10;
const int SAG_1 = 5;
const int SAG_2 = 4;

void setup() {
  Serial.begin(9600);
  pinMode(EN_SAG, OUTPUT);
  pinMode(SAG_1,  OUTPUT);
  pinMode(SAG_2,  OUTPUT);
}

void loop() {
  SAGA_DON();delay(1000);
  DUR();     delay(1000);

  SOLA_DON();delay(1000);
  DUR();     delay(1000);
}

void SAGA_DON() {
  analogWrite(EN_SAG, 255);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, HIGH);
}

void SOLA_DON() {
  analogWrite(EN_SAG, 255);
  digitalWrite(SAG_1, HIGH);
  digitalWrite(SAG_2, LOW);
}

void DUR() {
  analogWrite(EN_SAG, 0);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, LOW);
}
```

27. Tek DC Motor Yönlü Döndürme (PWM) Uygulaması

```
const int EN_SAG = 10;
const int SAG_1 = 5;
const int SAG_2 = 4;

int DONUS_HIZI;

void setup() {
  Serial.begin(9600);
  pinMode(EN_SAG, OUTPUT);
  pinMode(SAG_1,  OUTPUT);
  pinMode(SAG_2,  OUTPUT);
}

void loop() {
  DONUS_HIZI = analogRead(A0) / 4;

  SAGA_DON();delay(1000);
  DUR();delay(1000);

  SOLA_DON();delay(1000);
  DUR();delay(1000);
}

void SAGA_DON()
{
  analogWrite(EN_SAG, DONUS_HIZI);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, HIGH);
}

void SOLA_DON()
{
  analogWrite(EN_SAG, DONUS_HIZI);
  digitalWrite(SAG_1, HIGH);
  digitalWrite(SAG_2, LOW);
}

void DUR()
{
  analogWrite(EN_SAG, 0);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, LOW);
}
```

28. İki DC Motor Yönlü Döndürme Uygulaması

```
const int EN_SAG = 10;
const int SAG_1 = 5;
const int SAG_2 = 4;

const int EN_SOL = 11;
const int SOL_1 = 7;
const int SOL_2 = 6;

void setup() {
  Serial.begin(9600);

  pinMode(EN_SAG, OUTPUT);
  pinMode(SAG_1,  OUTPUT);
  pinMode(SAG_2,  OUTPUT);

  pinMode(EN_SOL, OUTPUT);
  pinMode(SOL_1,  OUTPUT);
  pinMode(SOL_2,  OUTPUT);
}

void loop() {
  SAGA_DON();
  delay(1000);

  DUR();
  delay(1000);

  SOLA_DON();
  delay(1000);

  DUR();
  delay(1000);
}

void SAGA_DON() {
  analogWrite(EN_SAG, 255);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, HIGH);

  analogWrite(EN_SOL, 255);
  digitalWrite(SOL_1, HIGH);
  digitalWrite(SOL_2, LOW);
}

void SOLA_DON() {
  analogWrite(EN_SAG, 255);
  digitalWrite(SAG_1, HIGH);
  digitalWrite(SAG_2, LOW);

  analogWrite(EN_SOL, 255);
  digitalWrite(SOL_1, LOW);
  digitalWrite(SOL_2, HIGH);
}

void DUR() {
  analogWrite( EN_SAG, 0);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, LOW);

  analogWrite( EN_SOL, 0);
  digitalWrite(SOL_1, LOW);
  digitalWrite(SOL_2, LOW);
}
```

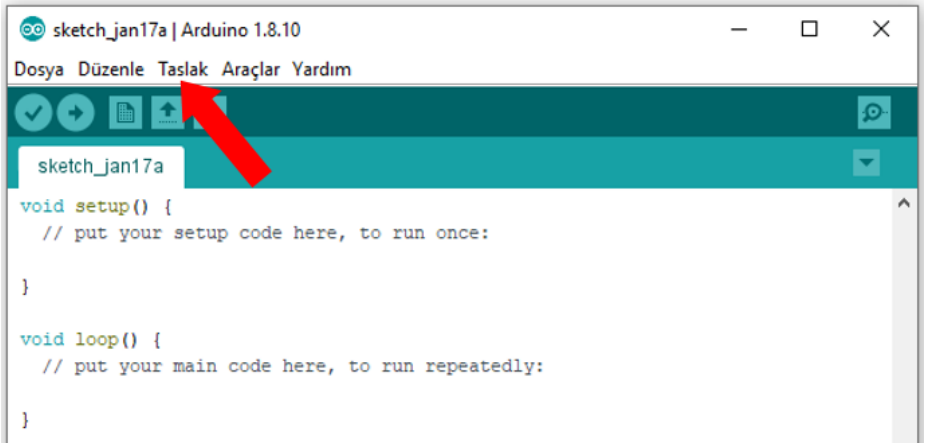
29. İki DC Motor Yönlü Döndürme (PWM) Uygulaması

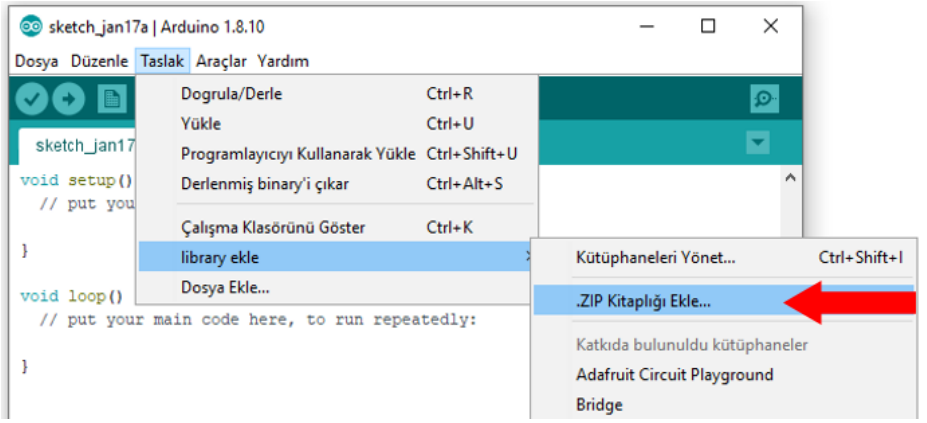
```
const int EN_SAG = 10;
const int SAG_1 = 5;
const int SAG_2 = 4;

const int EN_SOL = 11;
const int SOL_1 = 7;
const int SOL_2 = 6;

int DONUS_HIZI;

void setup() {
  Serial.begin(9600);

  pinMode(EN_SAG, OUTPUT);
  pinMode(SAG_1,  OUTPUT);
  pinMode(SAG_2,  OUTPUT);

  pinMode(EN_SOL, OUTPUT);
  pinMode(SOL_1,  OUTPUT);
  pinMode(SOL_2,  OUTPUT);
}

void loop() {
  DONUS_HIZI = analogRead(A0)/4;

  SAGA_DON();
  delay(1000);

  DUR();
  delay(1000);

  SOLA_DON();
  delay(1000);

  DUR();
  delay(1000);
}

void SAGA_DON() {
  analogWrite(EN_SAG, DONUS_HIZI);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, HIGH);

  analogWrite(EN_SOL, DONUS_HIZI);
  digitalWrite(SOL_1, HIGH);
  digitalWrite(SOL_2, LOW);
}

void SOLA_DON() {
  analogWrite(EN_SAG, DONUS_HIZI);
  digitalWrite(SAG_1, HIGH);
  digitalWrite(SAG_2, LOW);

  analogWrite(EN_SOL, DONUS_HIZI);
  digitalWrite(SOL_1, LOW);
  digitalWrite(SOL_2, HIGH);
}

void DUR() {
  analogWrite(EN_SAG, 0);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, LOW);

  analogWrite(EN_SOL, 0);
  digitalWrite(SOL_1, LOW);
  digitalWrite(SOL_2, LOW);
}
```

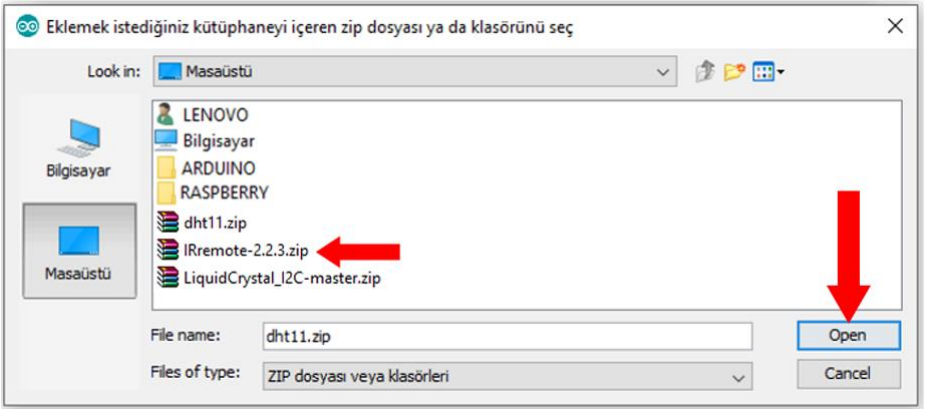
30. Ortak Motor Sürücü ile 4WD Araç Kiti Uygulaması

```
const int EN_SAG = 10;
const int SAG_1  = 5;
const int SAG_2  = 4;

const int EN_SOL = 11;
const int SOL_1  = 7;
const int SOL_2  = 6;

int   DONUS_HIZI;
char  KOMUT;

void setup(){
  Serial.begin(9600);

  pinMode(EN_SAG, OUTPUT);
  pinMode(SAG_1,  OUTPUT);
  pinMode(SAG_2,  OUTPUT);

  pinMode(EN_SOL, OUTPUT);
  pinMode(SOL_1,  OUTPUT);
  pinMode(SOL_2,  OUTPUT);
}

void loop(){
  DONUS_HIZI = analogRead(A0)/4;
  Serial.print("DONUS_HIZI : ");
  Serial.println(DONUS_HIZI);

  if ( Serial.available() ){
    KOMUT = Serial.read();

    if ( KOMUT == '8')
      ILERI_GIT();

    if ( KOMUT == '6')
      SAGA_DON();

    if ( KOMUT == '4')
      SOLA_DON();

    if ( KOMUT == '2')
      GERI_GIT();

    if ( KOMUT == '0')
      DUR();
  }
}

void ILERI_GIT() {
  analogWrite( EN_SAG, DONUS_HIZI);
  digitalWrite(SAG_1, HIGH);
  digitalWrite(SAG_2, LOW);

  analogWrite( EN_SOL, DONUS_HIZI);
  digitalWrite(SOL_1, LOW);
  digitalWrite(SOL_2, HIGH);
}

void GERI_GIT() {
  analogWrite( EN_SAG, DONUS_HIZI);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, HIGH);

  analogWrite( EN_SOL, DONUS_HIZI);
  digitalWrite(SOL_1, HIGH);
  digitalWrite(SOL_2, LOW);
}

void SAGA_DON() {
  analogWrite( EN_SAG, DONUS_HIZI);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, HIGH);

  analogWrite( EN_SOL, DONUS_HIZI);
  digitalWrite(SOL_1, LOW);
  digitalWrite(SOL_2, HIGH);
}

void SOLA_DON() {
  analogWrite( EN_SAG, DONUS_HIZI);
  digitalWrite(SAG_1, HIGH);
  digitalWrite(SAG_2, LOW);

  analogWrite( EN_SOL, DONUS_HIZI);
  digitalWrite(SOL_1, HIGH);
  digitalWrite(SOL_2, LOW);
}

void DUR() {
  analogWrite( EN_SAG, 0);
  digitalWrite(SAG_1, LOW);
  digitalWrite(SAG_2, LOW);

  analogWrite( EN_SOL, 0);
  digitalWrite(SOL_1, LOW);
  digitalWrite(SOL_2, LOW);
}
```

E.6. Kızılötesi (IR) Uygulamaları







31. Kızılötesi (IR) Kodunu Serial Monitorde Yazdırma Uygulaması

```
#include <IRremote.h>

const int IRPin = A3;

IRrecv IR_RECV(IRPin);

decode_results IR_RESULTS;

void setup() {

  Serial.begin(9600);

  IR_RECV.enableIRIn();

  IR_RECV.blink13(false);

  pinMode(IRPin, INPUT);
}

void loop() {

    if (IR_RECV.decode(&IR_RESULTS))
    {
        Serial.print("IR_CODE : ");
        Serial.println(IR_RESULTS.value, HEX);

        IR_RECV.resume();
    }
}
```

32. Kumanda ile TUŞ Koduna Bağlı LED Yak-Söndür Uygulaması

```
#include <IRremote.h>

const int IRPin = A3;

IRrecv IR_RECV(IRPin);

decode_results IR_RESULTS;

const int LED_1 = 4;
const int LED_2 = 6;

void setup() {

  Serial.begin(9600);

  IR_RECV.enableIRIn();
  IR_RECV.blink13(false);
pinMode(IRPin, INPUT);
  pinMode(LED_1, OUTPUT);
  pinMode(LED_2, OUTPUT);

}

void loop() {

    if (IR_RECV.decode(&IR_RESULTS))
    {
        Serial.print("IR_CODE : ");
        Serial.println(IR_RESULTS.value, HEX);

        if (IR_RESULTS.value == 0xFFA25D)         // "1"
        {
          digitalWrite(LED_1, HIGH);
          delay(1000);
          digitalWrite(LED_1, LOW);
        }

        if (IR_RESULTS.value == 0xFF629D)         // "2"
        {
          digitalWrite(LED_2, HIGH);
          delay(1000);
          digitalWrite(LED_2, LOW);
        }

        IR_RECV.resume();
    }
}
```

E.7. Bluetooth (HC-06) Uygulamaları

33. Bluetooth **KARAKTER** Komut Okuma Ve Serial Monitor Yazdırma Uygulaması

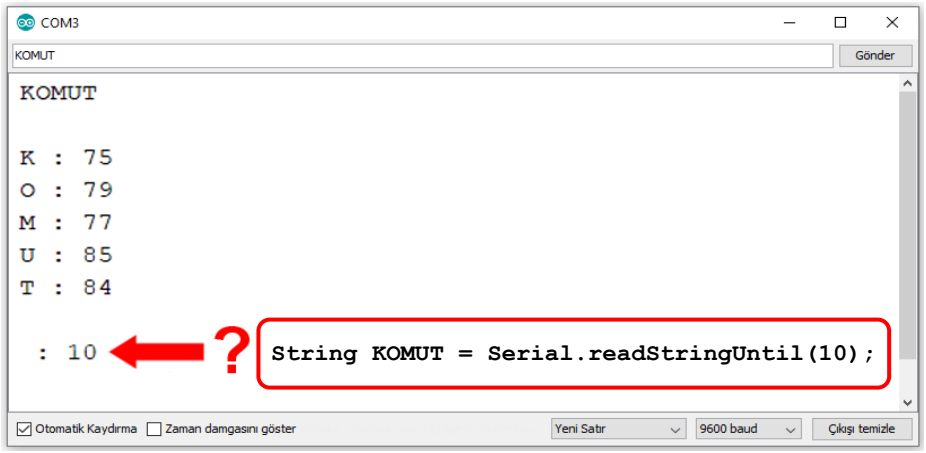
```
void setup() {
  Serial.begin(9600);
}

void loop() {
  if ( Serial.available() )
  {
char KOMUT = Serial.read();

    Serial.print( KOMUT );          // KARAKTER YAZAR
    Serial.print( " : " );
    Serial.println( int(KOMUT) );   // ASCII KOD YAZAR
  }
}
```

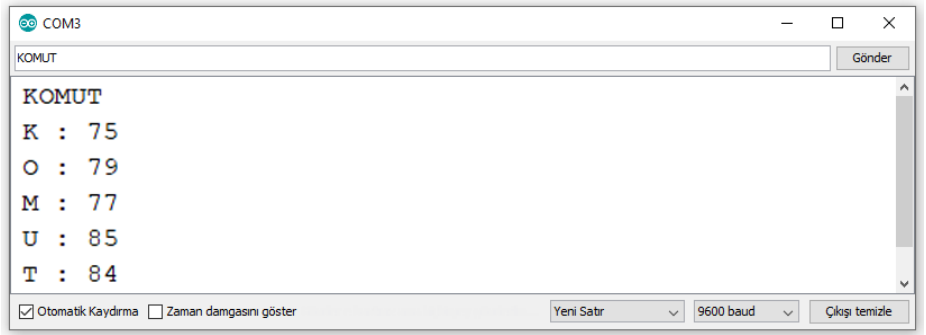
34. Bluetooth **KARAKTER** Komut Okuma Ve PORT Kontrol Uygulaması

```
int LED    = 6;
int BUZZER = 13;

void setup() {
  Serial.begin(9600);

  pinMode(LED,    OUTPUT);
  pinMode(BUZZER, OUTPUT);
}
void loop() {
  if ( Serial.available() )
  {
char KOMUT = Serial.read();

    if ( KOMUT == 'A' ) digitalWrite(LED, HIGH);
    if ( KOMUT == 'B' ) digitalWrite(LED, LOW);

    if ( KOMUT == 'C' ) digitalWrite(BUZZER, HIGH);
    if ( KOMUT == 'D' ) digitalWrite(BUZZER, LOW);
  }
}
```

35. Bluetooth **STRING** Komut Okuma Ve Serial Monitor Yazdırma Uygulaması

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  if ( Serial.available() )
  {
    String KOMUT = Serial.readString();

    Serial.println(KOMUT);

    for (int I=0; I<KOMUT.length(); I++) {
      Serial.print( KOMUT[I] );           // KARAKTER YAZAR
      Serial.print( " : " );
      Serial.println( int(KOMUT[I]) );    // ASCII KOD YAZAR
    }
  }
}
```





36. Bluetooth **STRING** Komut Okuma Ve PORT Kontrol Uygulaması

```
int LED    = 6;
int BUZZER = 13;

void setup() {
  Serial.begin(9600);

  pinMode(LED,    OUTPUT);
  pinMode(BUZZER, OUTPUT);
}

void loop() {
  if ( Serial.available() )
  {
    String KOMUT = Serial.readStringUntil(10);

    if ( KOMUT == "LED 1" ) digitalWrite(LED, HIGH);
    if ( KOMUT == "LED 0" ) digitalWrite(LED, LOW);

    if ( KOMUT == "BUZZER 1" ) digitalWrite(BUZZER, HIGH);
    if ( KOMUT == "BUZZER 0" ) digitalWrite(BUZZER, LOW);
  }
}
```

E.8. 7 Segment Display Modül Uygulamaları

37. A → G Segmentlerini Sırasıyla Kontrol Etme Uygulaması

```
const int PORTLAR[7] = {4, 5, 6, 7, 8, 9, 10};
// A, B, C, D, E, F, G

void setup() {
  Serial.begin(9600);

  pinMode(4,OUTPUT);       // A
  pinMode(5,OUTPUT);       // B
  pinMode(6,OUTPUT);       // C
  pinMode(7,OUTPUT);       // D
  pinMode(8,OUTPUT);       // E
  pinMode(9,OUTPUT);       // F
  pinMode(10,OUTPUT);      // G
}

void loop() {
  for (int j = 0; j < 7; j++)
  {
    digitalWrite (PORTLAR[j], 1);
    delay(250);
  }

  delay(500);

  for (int j = 0; j < 7; j++)
  {
    digitalWrite (PORTLAR[j], 0);
    delay(250);
  }
}
```

38. 0 → 9 Decimal **Yukarı** – Aşağı Sayıcı Uygulaması

```
const int SEG7K[16][7] = {
                           {1,1,1,1,1,1,0},     // 0
                           {0,1,1,0,0,0,0},     // 1
                           {1,1,0,1,1,0,1},     // 2
                           {1,1,1,1,0,0,1},     // 3
                           {0,1,1,0,0,1,1},     // 4
                           {1,0,1,1,0,1,1},     // 5
                           {1,0,1,1,1,1,1},     // 6
                           {1,1,1,0,0,0,0},     // 7
                           {1,1,1,1,1,1,1},     // 8
                           {1,1,1,1,0,1,1},     // 9
                           {1,1,1,0,1,1,1},     // A
                           {0,0,1,1,1,1,1},     // B
                           {1,0,0,1,1,1,0},     // C
                           {0,1,1,1,1,0,1},     // D
                           {1,0,0,1,1,1,1},     // E
                           {1,0,0,0,1,1,1},     // F
                         };

// A,B,C,D,E,F,G

const int PORTLAR[7] = {4, 5, 6, 7, 8, 9, 10};
// A, B, C, D, E, F, G

void setup() {
  Serial.begin(9600);

  pinMode(4,OUTPUT);       // A
  pinMode(5,OUTPUT);       // B
  pinMode(6,OUTPUT);       // C
  pinMode(7,OUTPUT);       // D
  pinMode(8,OUTPUT);       // E
  pinMode(9,OUTPUT);       // F
  pinMode(10,OUTPUT);      // G
}

void YAZ (int SAYI) {
  for (int j = 0; j < 7; j++)
    digitalWrite (PORTLAR[j], SEG7K[SAYI][j] );

  delay(500);
}

void loop() {
  for (int i = 0; i < 10; i++) //for (int i=9; i>=0; i--)
    YAZ(i);
}
```

39. 0 → F HexDecimal **Yukarı** – Aşağı Sayıcı Uygulaması

```
const int SEG7K[16][7] = {
                           {1,1,1,1,1,1,0},     // 0
                           {0,1,1,0,0,0,0},     // 1
                           {1,1,0,1,1,0,1},     // 2
                           {1,1,1,1,0,0,1},     // 3
                           {0,1,1,0,0,1,1},     // 4
                           {1,0,1,1,0,1,1},     // 5
                           {1,0,1,1,1,1,1},     // 6
                           {1,1,1,0,0,0,0},     // 7
                           {1,1,1,1,1,1,1},     // 8
                           {1,1,1,1,0,1,1},     // 9
                           {1,1,1,0,1,1,1},     // A
                           {0,0,1,1,1,1,1},     // B
                           {1,0,0,1,1,1,0},     // C
                           {0,1,1,1,1,0,1},     // D
                           {1,0,0,1,1,1,1},     // E
                           {1,0,0,0,1,1,1},     // F
                         };
// A,B,C,D,E,F,G

const int PORTLAR[7] = {4, 5, 6, 7, 8, 9, 10};
// A, B, C, D, E, F, G

void setup() {
  Serial.begin(9600);

  pinMode(4,OUTPUT);       // A
  pinMode(5,OUTPUT);       // B
  pinMode(6,OUTPUT);       // C
  pinMode(7,OUTPUT);       // D
  pinMode(8,OUTPUT);       // E
  pinMode(9,OUTPUT);       // F
  pinMode(10,OUTPUT);      // G
}

void YAZ (int SAYI) {
  for (int j = 0; j < 7; j++)
    digitalWrite (PORTLAR[j], SEG7K[SAYI][j] );

  delay(500);
}

void loop() {

  for (int i = 0; i < 16; i++)//for (int i=15; i>=0; i--)
    YAZ(i);
}
```

40. Button Kontrollü Yukarı – Aşağı Sayıcı Uygulaması

```
const int SEG7K[16][7] = {
                           {1,1,1,1,1,1,0}, {0,1,1,0,0,0,0},
                           {1,1,0,1,1,0,1}, {1,1,1,1,0,0,1},
                           {0,1,1,0,0,1,1}, {1,0,1,1,0,1,1},
                           {1,0,1,1,1,1,1}, {1,1,1,0,0,0,0},
                           {1,1,1,1,1,1,1}, {1,1,1,1,0,1,1},
                           {1,1,1,0,1,1,1}, {0,0,1,1,1,1,1},
                           {1,0,0,1,1,1,0}, {0,1,1,1,1,0,1},
                           {1,0,0,1,1,1,1}, {1,0,0,0,1,1,1},
                           {0,0,0,0,0,0,1}, {0,0,0,0,0,0,0},
                         };

const int PORTLAR[7] = {4, 5, 6, 7, 8, 9, 10};
// A, B, C, D, E, F, G

String MOD    = "YUKARI";
int    SAYAC  = 0;

void setup() {
  pinMode(4,OUTPUT);       // A
  pinMode(5,OUTPUT);       // B
  pinMode(6,OUTPUT);       // C
  pinMode(7,OUTPUT);       // D
  pinMode(8,OUTPUT);       // E
  pinMode(9,OUTPUT);       // F
  pinMode(10,OUTPUT);      // G
  pinMode(2,INPUT);
  pinMode(3,INPUT);
}

void YAZ (int SAYI) {
  for (int j = 0; j < 7; j++)
    digitalWrite (PORTLAR[j], SEG7K[SAYI][j] );

  delay(500);
}

void loop() {
  YAZ(SAYAC);
  if ( digitalRead(2) ) MOD = "ASAGI";
  if ( digitalRead(3) ) MOD = "YUKARI";
  if ( MOD == "ASAGI"  ) SAYAC--;
  if ( MOD == "YUKARI" ) SAYAC++;
  if ( SAYAC < 0 ) SAYAC = 15;
  if ( SAYAC > 15 ) SAYAC = 0;
}
```

41. Tek 7 Segment Potansiyometre Kalibrasyon Bilgisi Gösterge Uygulaması

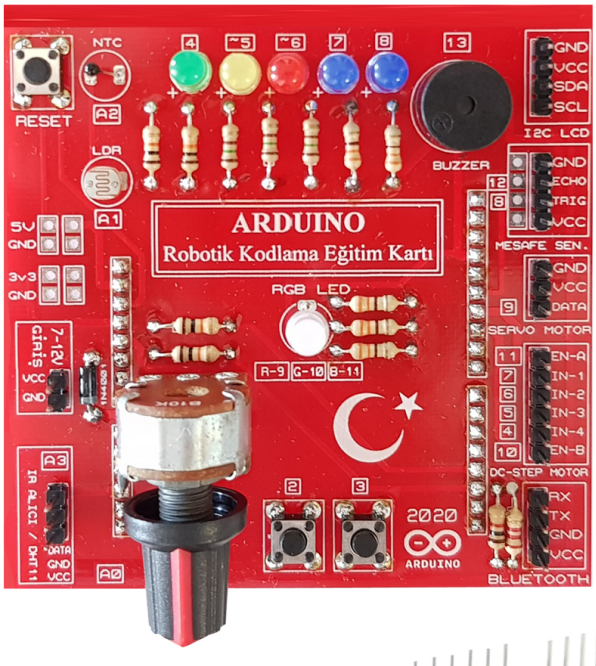
```
const int SEG7K[16][7] = {
                           {1,1,1,1,1,1,0},     // 0
                           {0,1,1,0,0,0,0},     // 1
                           {1,1,0,1,1,0,1},     // 2
                           {1,1,1,1,0,0,1},     // 3
                           {0,1,1,0,0,1,1},     // 4
                           {1,0,1,1,0,1,1},     // 5
                           {1,0,1,1,1,1,1},     // 6
                           {1,1,1,0,0,0,0},     // 7
                           {1,1,1,1,1,1,1},     // 8
                           {1,1,1,1,0,1,1},     // 9
                           {1,1,1,0,1,1,1},     // A
                           {0,0,1,1,1,1,1},     // B
                           {1,0,0,1,1,1,0},     // C
                           {0,1,1,1,1,0,1},     // D
                           {1,0,0,1,1,1,1},     // E
                           {1,0,0,0,1,1,1},     // F
                         };
// A,B,C,D,E,F,G

const int PORTLAR[7] = {4, 5, 6, 7, 8, 9, 10};
// A, B, C, D, E, F, G

void setup() {
  Serial.begin(9600);

  pinMode(4,OUTPUT);       // A
  pinMode(5,OUTPUT);       // B
  pinMode(6,OUTPUT);       // C
  pinMode(7,OUTPUT);       // D
  pinMode(8,OUTPUT);       // E
  pinMode(9,OUTPUT);       // F
  pinMode(10,OUTPUT);      // G
}

void YAZ (int SAYI) {
  for (int j = 0; j < 7; j++)
    digitalWrite (PORTLAR[j], SEG7K[SAYI][j] );
}

void loop() {
  int POT_K = map( analogRead(A0), 0, 1023, 0, 15 );

  YAZ(POT_K);
}
```

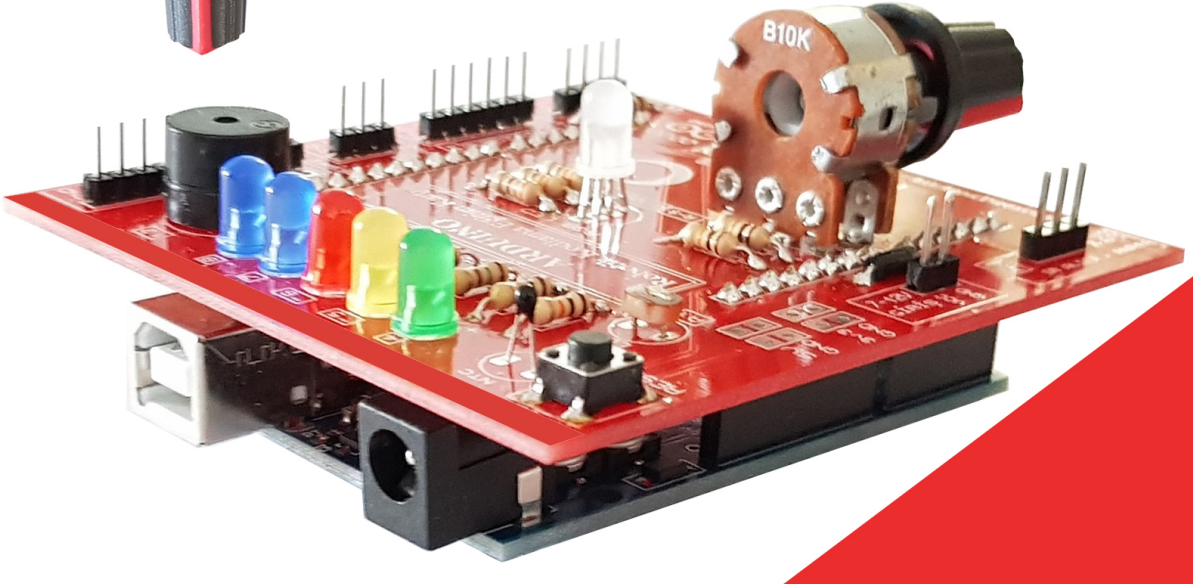
42. Tek 7 Segment Dijital Saat Gösterge Uygulaması

```
const int SEG7K[18][7] = {
                          {1,1,1,1,1,1,0}, {0,1,1,0,0,0,0},
                          {1,1,0,1,1,0,1}, {1,1,1,1,0,0,1},
                          {0,1,1,0,0,1,1}, {1,0,1,1,0,1,1},
                          {1,0,1,1,1,1,1}, {1,1,1,0,0,0,0},
                          {1,1,1,1,1,1,1}, {1,1,1,1,0,1,1},
                          {1,1,1,0,1,1,1}, {0,0,1,1,1,1,1},
                          {1,0,0,1,1,1,0}, {0,1,1,1,1,0,1},
                          {1,0,0,1,1,1,1}, {1,0,0,0,1,1,1},
                          {0,0,0,0,0,0,1}, {0,0,0,0,0,0,0},
                        };

const int PORTLAR[7] = {4, 5, 6, 7, 8, 9, 10};
// A, B, C, D, E, F, G

String SAAT = "12:34";

void setup() {
  Serial.begin(9600);

  pinMode(4,OUTPUT);       // A
  pinMode(5,OUTPUT);       // B
  pinMode(6,OUTPUT);       // C
  pinMode(7,OUTPUT);       // D
  pinMode(8,OUTPUT);       // E
  pinMode(9,OUTPUT);       // F
  pinMode(10,OUTPUT);      // G

  pinMode(2,INPUT);
  pinMode(3,INPUT);
}

void YAZ (int SAYI) {
  for (int j = 0; j < 7; j++)
    digitalWrite (PORTLAR[j], SEG7K[SAYI][j] );

  delay(500);
}

void loop() {
  for ( int I = 0; I < 5; I++ )
    if ( SAAT[I] != ':' )  YAZ( int(SAAT[I])-48 );
    else YAZ( 16 );

  YAZ( 17 );
  delay(1000);
}
```

TEMEL SEVİYE (BLOK TABANLI)
İLERİ SEVİYE (METİN TABANLI)
UYGULAMA GELİŞTİRME İMKANI

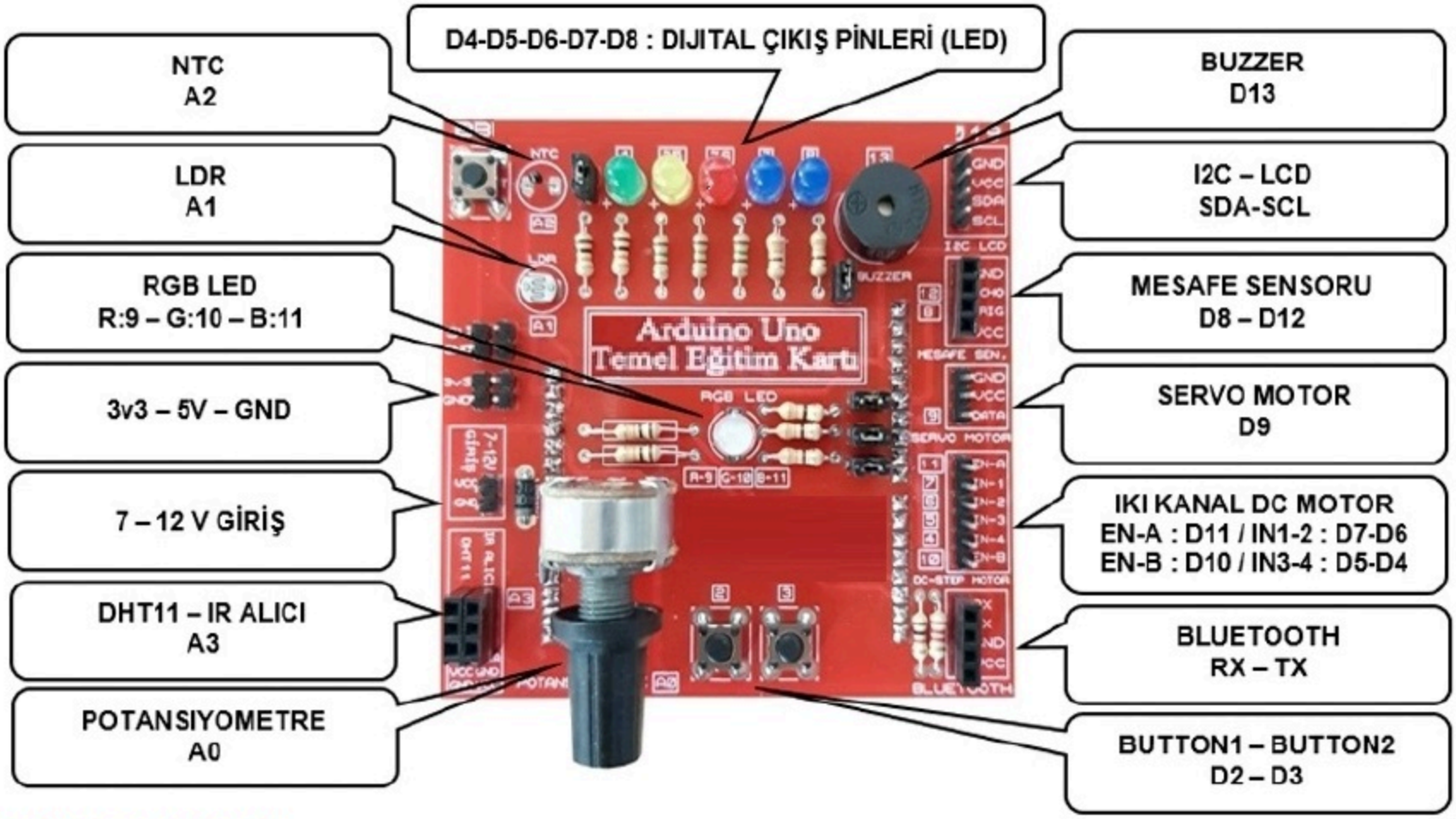
BLOK VE METİN TABANLI
100 ÖRNEK UYGULAMA
KLAVUZU





ARDUINO UNO
ROBOTİK KODLAMA
EĞİTİM-UYGULAMA KARTI [v2]

ROBOTİK KODLAMA EĞİTİM KARTI v2



EĞİTİM SETİ İÇERİĞİ

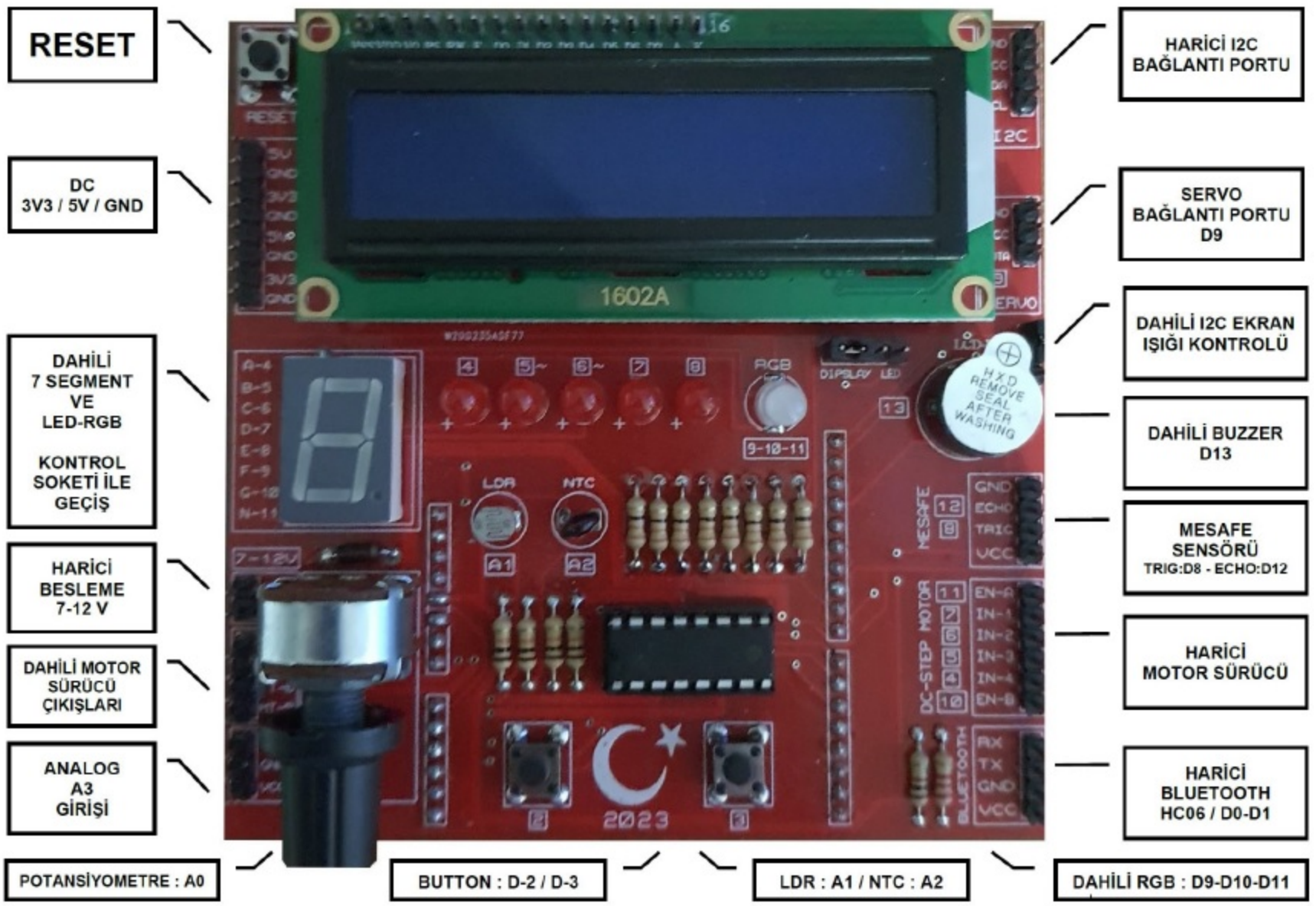
ARDUINO UNO R3 KLON + KABLO
DHT11 ISI VE NEM SENSÖR KARTI
DİŞİ-DİŞİ JUMPER KABLO - 20cm
HAREKET SENSÖRÜ MODÜLÜ
LAZER MODÜLÜ - 5V - 650nm - 5mW
MESAFE SENSÖRÜ - HC-SR04
ROLE MODÜLÜ - 5V TEK KANAL
SERVO MOTOR - METAL DİŞLİ - MG90S
SES SENSÖRÜ KARTI
TOPRAK NEM SENSÖR KARTI
PLASTİK MALZEME KUTUSU

TEMEL SEVİYE (BLOK TABANLI)
İLERİ SEVİYE (METİN TABANLI)
UYGULAMA GELİŞTİRME İMKANI

BLOK VE METİN TABANLI
100 ÖRNEK UYGULAMA
KLAVUZU

ARDUINO UNO
ROBOTİK KODLAMA
EĞİTİM-UYGULAMA KARTI [v3]

ROBOTİK KODLAMA EĞİTİM KARTI v3



EĞİTİM SETİ İÇERİĞİ

ARDUINO UNO R3 KLON + KABLO
BLUETOOTH - HC06 MODÜLÜ
ÇOK AMAÇLI MOBİL ROBOT PLATFORMU 4WD
DHT11 ISI VE NEM SENSÖR KARTI
DİŞİ-DİŞİ JUMPER KABLO - 20cm
HAREKET SENSÖRÜ MODÜLÜ
JOYSTICK MODÜLÜ - 2 EKSENLİ
KIZILÖTESİ KUMANDA KİTİ
L298N MOTOR SÜRÜCÜ KARTI
LAZER MODÜLÜ - 5V - 650nm - 5mW
MESAFE SENSÖRÜ - HC-SR04
MESAFE SENSÖRÜ - HC-SR04 - TUTUCU
ROLE MODÜLÜ - 5V TEK KANAL
SERVO MOTOR - METAL DİŞLİ - MG90S
SES SENSÖRÜ KARTI
STEP MOTOR + SÜRÜCÜ MODÜLÜ
TOPRAK NEM SENSÖR KARTI
PLASTİK MALZEME KUTUSU