# 2. Bölüm – Temel Devre Elemanları ile Deneyap Kart ve Kodlama Aracı Arduino IDE



**Ön bilgi:**

* Öğrenciler “gerilim”, “akım”, “direnç” tanımlarını bilir.
* Öğrenciler direncin çalışma prensibini bilir ve akım ile arasındaki ilişkiyi açıklayabilir.
* Öğrenciler “Deneyap Kart” bileşenlerini bilir ve “Arduino IDE” arayüzünü kullanabilir.

**Bölüm Kazanımları:**

* Öğrenciler “breadboard” çalışma prensiplerini bilir.
* Öğrenciler “Seri/Paralel Bağlantı”, “Analog/Dijital veri” ve “pinler” hakkında bilgi sahibidir.
* Öğrenciler dirençler üzerindeki farklı renklerden dirençlerin ohm değerlerini hesaplayabilir.
* Öğrenciler “buzzer” devre elemanının çalışma prensiplerini bilir.
* Öğrenciler “potansiyometre” devre elemanının çalışma prensiplerini bilir.
* Öğrenciler “buton” devre elemanının çalışma prensiplerini bilir.
* Öğrenciler “potansiyometre” ile LED parlaklığını ayarlama ve seri haberleşme uygulamalarını gerçekleştirebilir.
* Öğrenciler “Buton” kontrollü LED yakma ve “buton” ile hareketli LED Uygulamalarını gerçekleştirebilir.
* Öğrenciler LED’ler, “buton”lar ve “buzzer” ile farklı uygulamaların devrelerini kurabilir ve “Arduino IDE” program kodlarını yazabilirler.
* “If/else” karar yapısı, “for” döngüsü, dizi değişkenini kullanarak programlama yapabilir.

**Haftanın Amacı:**

Bu haftanın amacı, öğrencilerin öncelikle “breadboard”, “seri/paralel bağlantı”, “Analog/Dijital veri” ve “Deneyap Kart” üzerindeki pinler hakkında bilgi sahibi olmalarını hedeflemektedir. Ayrıca direnç renk kodları, “potansiyometre”, “buton”, ve “buzzer” devre elemanlarının özelliklerini öğrenme ve bu devre elemanları ile farklı uygulamalar gerçekleştirebilmeleri amaçlanmaktadır. Bu kapsamda programlamada farklı kontrol yapıları, sabitler ve giriş-çıkışlar gibi konuların uygulamalar içerisindeki kullanımlarını da deneyimlemeleri hedefler içerisindedir.

**Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:**

| Malzeme Listesi | |
| --- | --- |
| Deneyap Kart | **Erkek-Dişi bağlantı kablosu** |
| Breadboard | **Buzzer** |
| 220 ohm direnç | **Hoparlör** |
| 560 ohm direnç | **Kırmızı LED** |
| 100 ohm direnç |  |
| 10 Kohm Potansiyometre |  |

**Haftanın İşlenişi:**

***Gözle:*** Breadboard, Seri/Paralel Bağlantı, Analog/Dijital Veri ve Pinler, direnç renk okuma, Buzzer, potansiyometre ve buton tanımlarının yapılması ve özelliklerinin verilmesi.

***Uygula:*** Buzzer, potansiyometre ve buton devre elemanlarının kullanıldığı örnek devrelerin oluşturulması ve programlanması.

***Tasarla:*** Bir buton ile dört hareketli LED uygulaması ve butonları kullanarak piyano tasarımının gerçekleştirilmesi.

***Üret:*** Bir buton ile dört hareketli LED uygulaması ve butonları kullanarak piyano tasarımını gerçekleştirme; devrelerini oluşturma ve Arduino IDE içerisinde program kodlarının yazılması.

***Değerlendir:*** Yansıtma Etkinliği.

## 1. ADIM: GÖZLE ve UYGULA

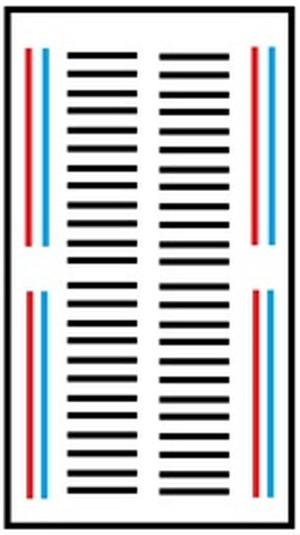
### 1.1 Gözle ve Uygula - Breadboard, Seri/Paralel Bağlantı, Analog/Dijital Veri ve Pinler

| Malzeme Listesi |
| --- |
| Deneyap Kart |
| Breadboard |
| Bağlantı kabloları |
| 220 ohm direnç |
| Kırmızı LED |

**Breadboard**

Elektronik devre elamanlarını bir arada tutan ve gerekli ara bağlantıları daha sağlıklı şekilde gerçekleştirmemizi sağlayan araçtır. Breadboard üzerinde aşağıdaki resimde de görüldüğü gibi yatay ve dikeyde ilerleyen çizgiler aslında kısa devredir, yani bu çizgiler bir kablo üzerinden bağlantılıdır diye düşünülebilir.

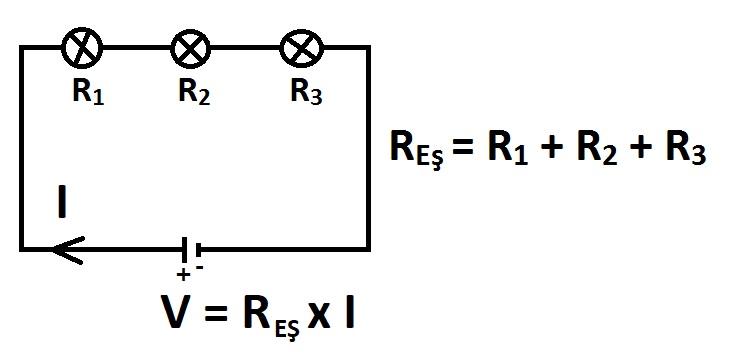
Breadboard’un yanlardan dikey şekilde uzanan hatlar genellikle güç bağlantılarını yapmak için kullanılır.



*Resim 2.1: Breadboard Bağlantıları*

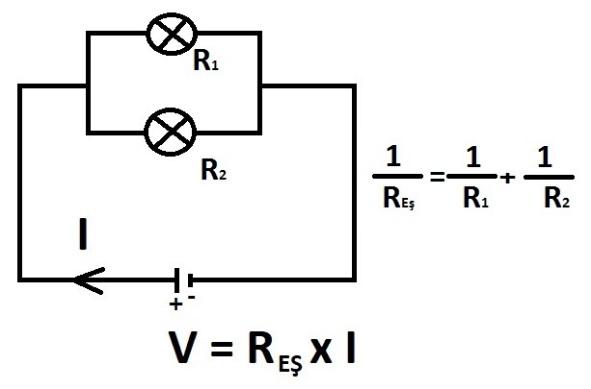
**Seri-Paralel Bağlantı**

Elektrik devreleri bağlantı şekillerine göre paralel ve seri olmak üzere ikiye ayrılırlar. Seri devrelerde akım devredeki elemanlardan tek bir hat üzerinden geçerler. Akım her bir eleman üzerinden geçer ve bu elemanlar akıma karşı direnç gösterirler. Bu nedenle devrede oluşan direnç artar. “Ohm kanunu” üzerinden düşünecek olursak “V=IxR”de voltaj sabitken, direnç artacağı için devre üzerinden geçecek akım azalacaktır. Aşağıdaki resimde dirençlerin seri bağlanması görülmektedir.



*Resim 2.2: Dirençlerin Seri Bağlanması*

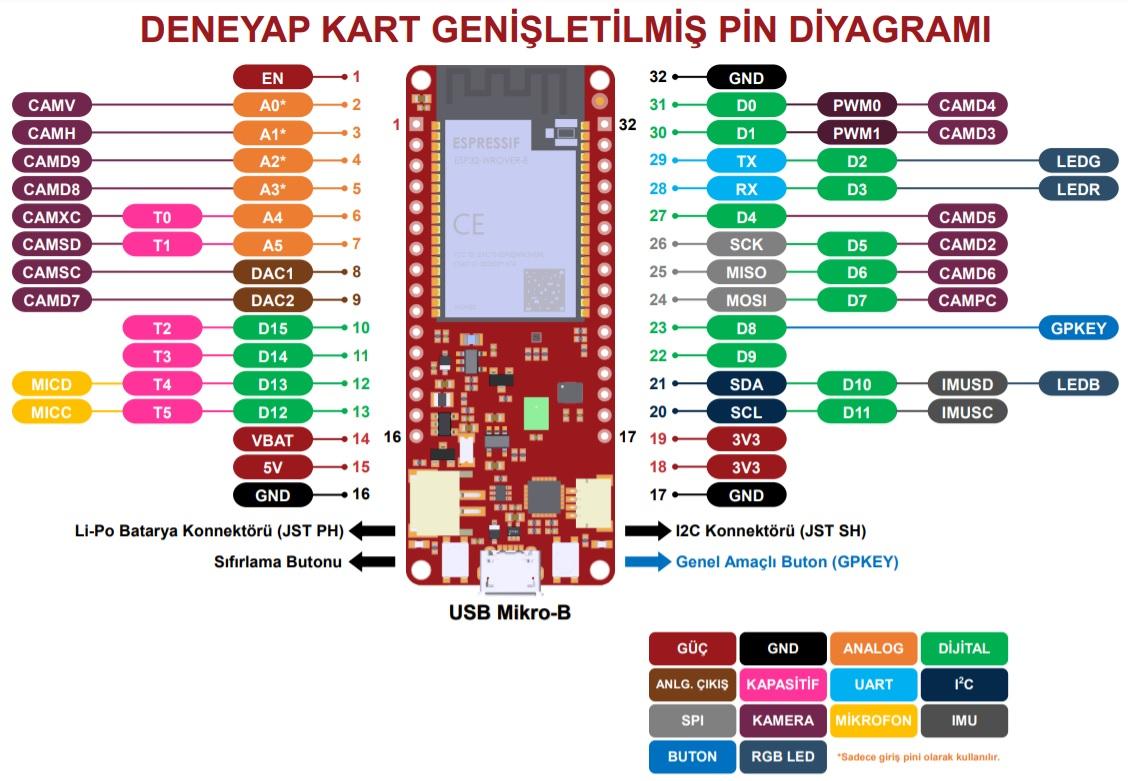
Paralel devrelerde ise devre elemanları birbirine paralel olarak bağlanır ve buna bağlı olarak direnç azalır. Gerilim sabitken direncin azalması sonucunda devre üzerinden geçecek akım artacaktır. Aşağıdaki resimde dirençlerin paralel bağlanması görülmektedir.



*Resim 2.3: Dirençlerin Paralel Bağlanması*

**Deneyap Kart ve Giriş – Çıkış Pinleri**

Deneyap Kart ve diğer mikrodenetleyici kart versiyonlarında farklı sayıda ve türde giriş – çıkış pinleri bulunmaktadır. Bu ders kapsamında Deneyap Kart üzerindeki giriş-çıkış pinlerinden bahsedilecektir. Bu pinler aslında Deneyap Kart’a bağladığımız sensör ve devre elemanlarıyla veri alışverişi için kullanılır. Aşağıdaki şekilde (*Resim 2.4*) Deneyap Kart üzerinde bulunan pinler detaylı olarak gösterilmiştir.



*Resim 2.4: Deneyap Kart Genişletilmiş Pin Diyagramı*

Deneyap Kart üzerinde bulunan pinlerden bazıları aşağıda açıklanmıştır. İlerleyen bölümlerde diğer pinlerin kullanılm durumlarına geçildiğinde daha detaylı bilgilendirme de yapılacaktır.

GND: Sensörlerin ve devre elemanlarının GND veya “-“ uçları bu pine bağlanır ve topraklama sağlar.

3,3V: Bu pinler 3,3V’luk besleme gerektiren cihazlara bağlanır ve enerji çıkış pinidir. USB ve Li-Po batarya güç girişi bağlantıları ile kart beslendiğinde DC-VC voltaj dönüştürücü aracılığıyla 3.3V gerilim ve 1A akım çıkış kapasitesine sahiptir.

5V: Bu pin 5V gerilim ve 0.5A akım çıkışı vermektedir. 5V’luk besleme gerektiren cihazlara bağlanır ve enerji çıkış pinidir.

VBAT: Harici olarak bağlanan Li-Po bataryanın 3.3V-4.2V arasındaki gerilimini çıkış olarak vermektedir. Harici bağlanacak üniteleri beslemek için kullanılabilir.

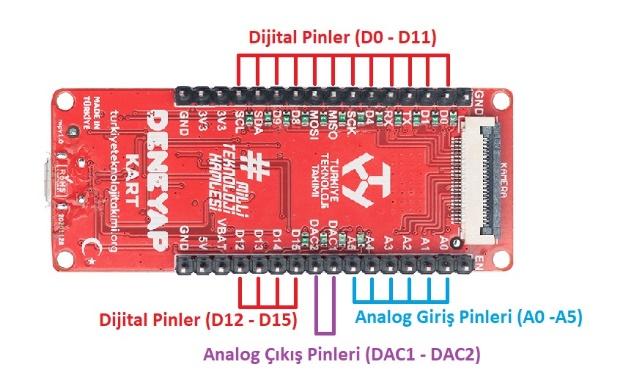
**Analog- Dijital Veri ve Pinler**

Analog ve dijital veri kavramlarını kısaca açıklamak gerekirse analog veri zamana göre sonsuz bir değer alan veri türü iken dijital veri genel olarak “var” (1) veya “yok” (0) bilgisinden oluşur. Örneğin bir arabanın hareketli ya da hareketsiz olması dijital bir veri gibi düşünülebilecekken arabanın seyir halindeki hızı ise analog bir veridir.

Deneyap Kart’ta aşağıdaki şekilde görüldüğü üzere D0 ile D15 arasında 16 adet dijital giriş-çıkış pini bulunmaktadır. Bu pinler “0” iken 0V, “1” iken ise 3.3V gerilim değerine sahip olurlar. Deneyap Kart’taki tüm dijital pinler PTC sigortalar ile kısa devre koruması altındadır. Burada dikkat edilmesi gereken nokta; Deneyap Kart’taki dijital pinler diğer bazı mikrodenetleyici kartlardaki gibi 5V gerilim toleranslı değildir.

Deneyap Kart üzerindeki “A0, A1, A2, A3, A4 ve A5” analog giriş pinleridir. 0V ile 3.3V aralığındaki analog işaretler için giriş pini olarak kullanılmaktadır. Bu pinler, bazı sensörler ve diğer devre elemanlarından gelen analog veriyi Deneyap Kart’a 0 ile 4095 arasında bir değer alarak aktarırlar. Deneyap Kart üzerindeki analog pinlerden A0 ile A3 arasındaki pinler sadece giriş pini olarak kullanılırken, A4 ve A5 pinleri dijital giriş/çıkış pini olarak da kullanılabilir.

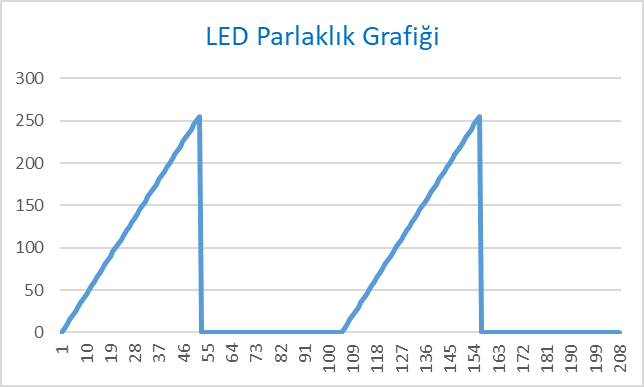
Analog çıkış sağlamak için ise Deneyap Kart üzerinde “DAC1 ve DAC2” pinleri bulunmaktadır. Bu pinler aracılığı ile 0V ile 3.3V aralığında analog sinyal çıkışı alınabilir. Bu pinler aynı zamanda dijital giriş/çıkış pini olarak da kullanılabilir.



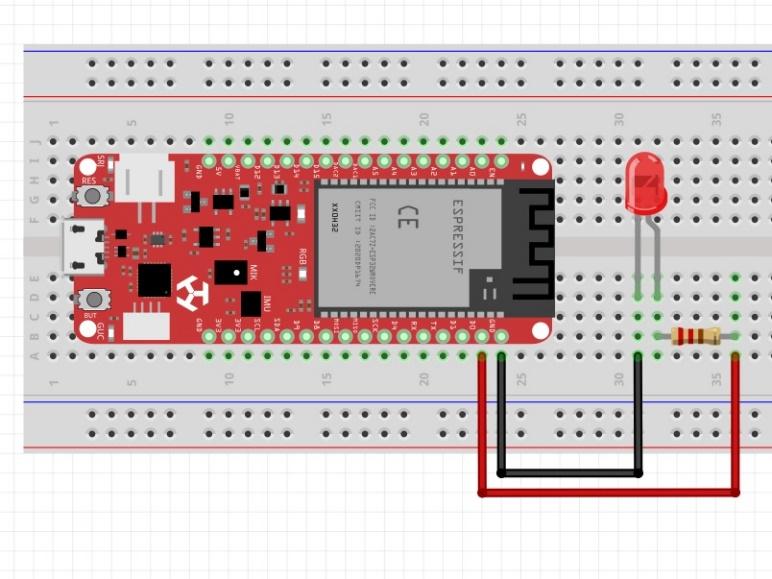
*Resim 2.5: Arduino UNO Analog Giriş-Çıkış Pinleri*

PWM: Deneyap Kart üzerinde analog çıkış verebilmek için tüm dijital pinler (D0 - D15) “0 ile 255” arasında değerler alarak ve dijital çıkışın frekansını arttırarak analog bir pin gibi davranırlar. Bu çıkışlara “PWM (Darbe Genişlik Modülasyonu)” çıkışları denir. Deneyap Kart üzerindeki dijital pinlerin analog çıkışı için PWM pini olarak kullanılabilmeleri için kodlama esnasında kanal ataması gibi ekstra ayarlamalar yapılması gerekmektedir. Aşağıdaki gözle ve uygula etkinliklerinde bu ayarlamalar ile ilgili detaylı bilgiler sunulacaktır. Bunlara ek olarak Deneyap Kart üzerindeki D0 ve D1 pinleri aynı zamanda PWM0 ve PWM1 pinleri olarak ek bir kodlama ayarı gerektirmeksizin PWM pini olarak analog çıkış için kullanılabilirler.

PWM pinlerinin kullanımına ilişkin bir örnek verilecek olursa normalde bu pinler dijital çıkış (0-5V, HIGH-LOW vb.) üretmesine karşın bu pinlere bağlı olan LED’lerin parlaklığı analog çıkış verecek şekilde ayarlanabilir. Aşağıda öğrencilerle LED parlaklığını 5 sn içerisinde maksimum parlaklığa ulaştırıp sönen ve “5 sn” sonra tekrar yanan bir uygulama gerçekleştirilecektir. LED parlaklığının 5 sn içerisinde maksimum parlaklığa ulaşması “0” değerinden başlayarak “255” değerine kadar ulaşmasını gerektirmektedir. “5 sn” içinde sönüp tekrar yanması da bu uygulamanın bir döngü içerisinde devam edeceğini göstermektedir. Aşağıdaki resimlerde led parlaklık grafiği ve etkinlik örnek devre şeması görülmektedir.



*Resim 2.6: LED Parlaklık Grafiği*



*Resim 2.7: Gözle-Uygula Etkinliği Örnek Devre Şeması*

| **Dikkat** |
| --- |
| Eğitmen devrenin fiziksel kurulumunu ve programın yazımından önce kontrol yapılarından döngüler hakkında öğrencileri kısaca bilgilendirmelidir. Özellikle **“for”** döngüsü hakkında “for” deyiminin küme parantezi içine alınmış bir deyim bloğunu tekrarlamak için kullanıldığı; döngüyü artırmak ve sonlandırmak için içerisinde genellikle bir *artış sayacı (değişken)* kullanıldığını özellikle belirtip aşağıdaki söz dizimini öğrencilere göstermelidir. “for” ifadesinin tekrar eden işlemler için yararlı olduğu ve genellikle veri / pin topluluğu üzerinde çalışmak için dizilerle birlikte kullanıldığı öğrencilere iletilmelidir.  Arduino IDE içerisinde yazılacak program kodlarında “i”, “j”, “k”, “l”, “m” ve “x” gibi değişken isimlerinin sıklıkla kullanılacağı konusunda öğrenciler bilgilendirilmelidir. Özellikle aşağıdaki söz diziminde parantez içerisindeki “i” değişkeninin Void fonksiyonlarından önceki bölümde tanımlanmadığı için parantez içerisinde “int” olarak yazılıp başlangıç değerinin “0” olarak tanımlandığının dikkati çekilmelidir. |

Uygulamanın devamında devre şeması ve daha sonra da devreye ilişkin “Arduino IDE” programı eğitmen tarafından tüm öğrencilere gösterilecektir. Bu aşamada LED parlaklığının 5 sn içerisinde hangi aralıkta değişim gösterdiğini öğrencilere gösterebilmek için Arduino IDE’nin sağ üst köşesindeki simgeye tıklanarak seri port ekranı açılmalıdır. LED parlaklık değer değişiminin gözlemleneceği bu ekranın sağ alt köşesinde yer alan değer (baud değeri) Arduino IDE programında kullanılan değer ile aynı olmalıdır (örnek: 115200 bound). Aşağıdaki resimde etkinlik için örnek kod görülmektedir.

| **Dikkat** |
| --- |
| Arduino IDE aracılığıyla çeşitli devre elemanları ile kodlama yapabilmek için bu devre elemanlarına ait kütüphanelerin kod bloğuna eklenmesi gerekiyor. Aşağıdaki kod incelendiğinde ilk olarak Denayap Kart üzerindeki analog sinyal ile ilgili kodları kullanabilmek için gerekli olan <analogWrite.h> kütüphanesinin eklendiği öğrencilere açıklanır. Kütüphane ekleme ile ilgili daha detaylı bilgi ilerleyen bölümlerde açıklanacağı da eğitmen tarafından öğrencilere açıklanır. |

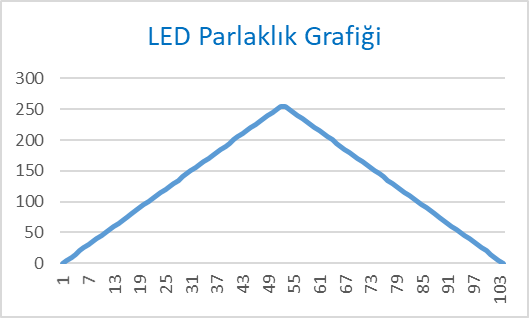


*Resim 2.8: Gözle-Uygula Etkinliği Örnek Kodu*

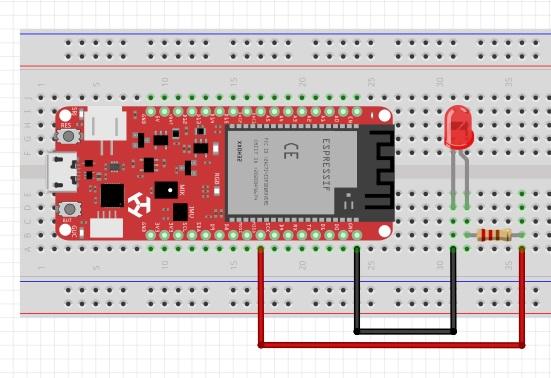
### 1.2 Uygula- LED Parlaklığı (Öğrenci 1)

| Malzeme Listesi |
| --- |
| Deneyap Kart |
| Breadboard |
| Bağlantı kabloları |
| 330 ohm direnç |
| Kırmızı LED |

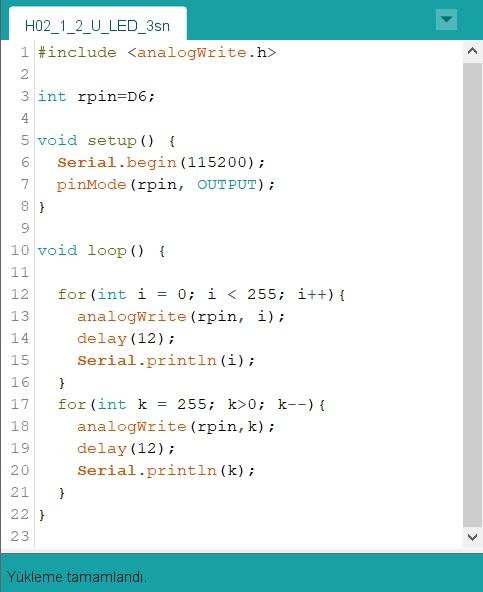
Bu etkinlikteki amaç öğrencilerin Deneyap Kart’ın D6 nolu pinini (PWM olarak) kullanarak bu pine bağlı olan LED’in parlaklığını 3 saniye içerisinde maksimuma ulaştırıp; daha sonraki 3 saniye içerisinde de söndüren uygulamayı gerçekleştirmektir. Eğitmen bu uygulamaya ilişkin öğrencilere aşağıdaki voltaj seviyesini gösteren grafiğe uygun şekilde devreyi kurmalarını ve Arduino IDE içerisinde programı yazmalarını isteyecektir. Aşağıdaki resimlerde led parlaklık grafiği, Deneyap Kart devresi ve istenen görevi yerine getiren programın örnek bir kodu verilmiştir.



*Resim 2.9: LED Parlaklık Grafiği*



*Resim 2.10: Uygula Etkinliği Örnek Devre Şeması*



*Resim 2.11: Uygula Etkinliği Örnek Kodu*

### 1.3 Gözle - Direnç, Direnç Okuma

**Direnç**

Daha önceki derste Ohm kanunundan (**V = I x R)** bahsedilmişti. Bu konuyu tekrar hatırlatmak gerekirse; sabit bir gerilim altında akımı azaltmak için devrede yer alan direnci değiştirebiliriz. Yani devrede akımı ayarlamak için direnç değerinde değişiklik yapılabilir. Bu durumu gerçek hayattan bir örnek ile aktarmak gerekirse altı şeritli bir otoyolda gerçekleşen bir kaza sonucu 2 şeridin iptal olması durumunda (direnç artımı) trafik akışı yavaşlayacaktır (akım azalacaktır).

**Direnç renk okuma:**

Dirençlerin üzerinde “ohm” değerleri direk olarak yazılmaz. Bunun yerine direnç üzerindeki renk kodları ile ohm cinsinden değerleri gösterilir. Dirençlerin üzerinde hassasiyetlerine bağlı olarak 4, 5 ya da 6 farklı renk bulunur. Bu renklerden sonuncusu direncin tolerans değerini belirtir. Direnci toleransı direncin üretimden kaynaklanan hata payını gösterir. Örnek vermek gerekirse 100 ohmluk direnç eğer %5’lik bir toleransa sahipse bu dirençten 95-105 ohm arasında direnç değeri elde edilebilir. Dolayısıyla direncin toleransı düştükçe daha hassas devreler kurulabilir. Diğer renkler ise aşağıda verilen formül kullanılarak direncin ohm değerinin hesaplanmasında kullanılır. Formülde yer alan şeritlerin renk değerleri ise şu şekildedir:

Renklerin değerleri: Siyah = 0, Kahverengi = 1, Kırmızı = 2, Turuncu = 3, Sarı = 4, Yeşil = 5, Mavi = 6, Mor = 7, Gri = 8, Beyaz = 9.

Direncin Ohm Değeri = (10X(ilk şeridin renk değeri)+1X(ikinci şeridin renk değeri))X10üçüncü şeridin renk değeri

| **Dikkat** |
| --- |
| Direnç renk kodları  Dirençlerin renk sayı değerlerini kolayca akılda tutmak için “ Sokakta Sayamam Gibi ” ifadesi ezberlenebilir. |

### 1.4 Gözle ve Uygula - Buzzer

| Malzeme Listesi |
| --- |
| Deneyap Kart |
| Breadboard |
| Bağlantı kabloları |
| 100 ohm direnç |
| Buzzer |

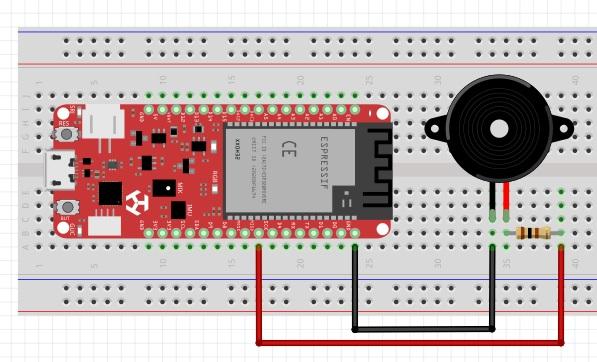
Deneyap Kart ile kullanılabilecek elektronik devre elemanlarından ışıklı devre elemanları daha önce işlenmişti. Işıklı devre elemanları gibi Deneyap Kart ile kullanılabilcek elektronik devre elemanları arasında sesli devre elemanları da bulunur. Bunlar elektrik sinyallerini sese çevirerek ses üretebilirler. Bu uygulamada kullanılacak elektronik devre elemanı olan Buzzer da ses üretebilen bir devre elemanıdır. Günlük yaşamda şarkı ve ses kaydı gibi ses unsurlarını oynatmak için hoparlörler sıklıkla kullanılır. Buzzerlar mini hoparlörlere benzetilebilir. Buzzerlar farklı ses sinyalleri üretebilir. Fakat hoparlör kadar net ses üretemez. Sadece belirli notalarda sesler üretebilir. Örneğin farklı notalarda bip sesi üretebilir. Alarm ve fırın gibi günlük hayatta kullanılan birçok cihaz içerisinde kullanılabilir. Piyasada aktif ve pasif olmak üzere iki çeşit buzzer bulunmaktadır. Bunlar birbirine çok benzemektedir. Her iki buzzer çeşidinin de iki bacağı bulunur. Bu bacaklardan büyük olanı “**+”** (artı) küçük olanı ise “**–“** (eksi) bacaktır. Aşağıda resimlerde iki adet buzzer görülmektedir. Bu resimlerde + bacaklar gösterilmiştir.

| buzzerUstten | buzzerYandan |
| --- | --- |

*Resim 2.12: Buzzer Bacak Bağlantıları*

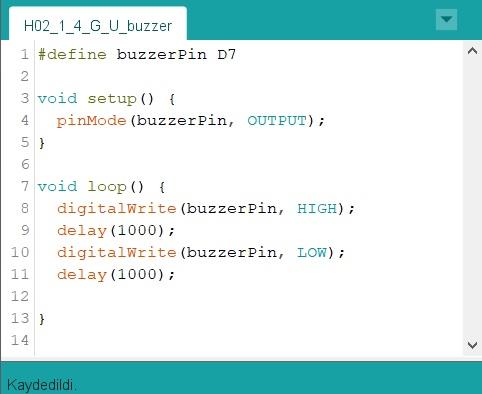
Aktif buzzer’a çalışma aralığı içerisinde bir doğru akım uygulandığında bip sesi üretir. Pasif buzzera doğru akım uygulandığında istenilen ses elde edilemez. Pasif buzzer’dan istenilen sesi elde etmek için Arduino IDE içerisinde gerekli komutları kullanmak gerekir. Bu komutlar bazı aktif buzzer’lar ile de kullanılabilir fakat her zaman istenilen sesi üretmeyebilir. Bu yüzden aktif ve pasif buzzer’lar farklı şekilde kodlanır.

Aktif ve pasif buzzer’ın devre şeması veya bağlantıları değişmez. İkisi de aynı şekilde yapılır. Bundan dolayı aşağıda bir devre şeması verilecektir, fakat bu her iki buzzer çeşidi için de uygulanabilir. Aşağıdaki resimde görüldüğü gibi buzzer’ın artı bacağı Deneyap Kart’ın D7 dijital pinine bağlanmıştır. Buzzer’ın diğer bacağı ise 100 ohm bir direnç üzerinden Deneyap Kart’ının GND pinine bağlanmıştır.



*Resim 2.13: Gözle-Uygula Etkinliği Örnek Devre Şeması*

Aşağıda resimde aktif buzzer için Arduino IDE’de yazılmış bir kod bulunmaktadır. Deneyap Kart’ın D7 dijital pinine bağlanan buzzer bu kod sayesinde bir saniye bip sesi çıkarır, bir saniye bekler ve bu işlemi sürekli yapar.



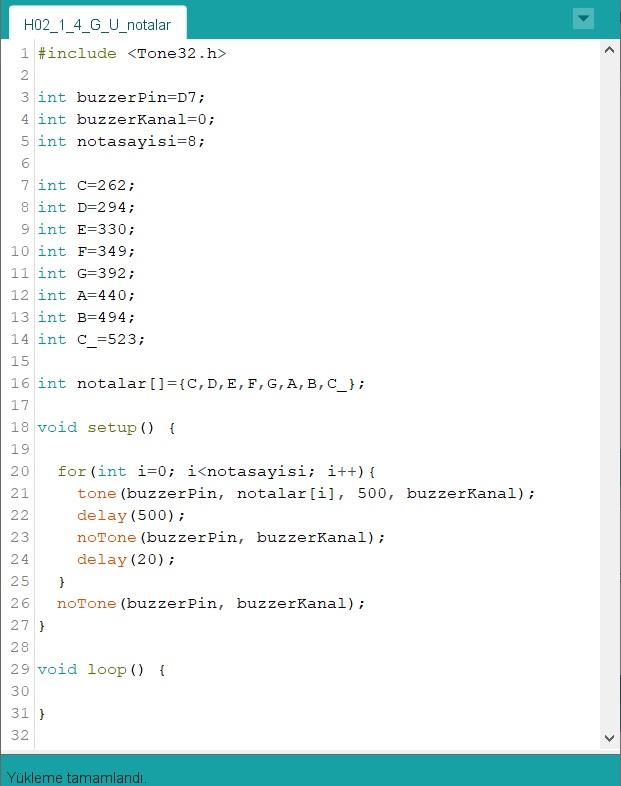
*Resim 2.14: Gözle-Uygula Etkinliği Örnek Kodu*

Bildiğimiz gibi hemen hemen her sesin kendine ait bir notası vardır. Pasif buzzer ile (buzzer’ın desteklediği ölçüde) istenilen nota veya istenilen frekanstaki ses çalınabilir. Notaların her biri için belirli olan bir Hertz cinsinden frekans değeri vardır. Aşağıdaki tabloda notaların Hertz cinsinden frekans karşılıkları verilmiştir.

| **Notaların Frekans (Hz) Karşılıkları** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DO | RE | Mİ | FA | SOL | LA | Sİ | DO (ince) |
| 262 | 294 | 330 | 349 | 392 | 440 | 494 | 523 |

Arduino IDE içerisinde her bir notanın frekans olarak karşılığı girildiğinde **tone** komutu ve gerekli kontrol yapıları kullanılarak C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La), B (Si), C\_ (ince Do) notaları çalınabilir. Frekans değerleri bilindiğinde ara notalar olan diyezler de çalınabilir. Aşağıda devre şeması ve Arduino IDE ile yazılmış program C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La), B (Si), C\_ (İnce Do) notalarını sırayla çalıp duran bir buzzer uygulamasını içermektedir. Uygulamada buzzer’ın artı ucu D7 dijital pinine bağlanmış ve bir kez çalışması amacıyla program setup fonksiyonu altına yazılmıştır. Buzzer ile çalınacak notalar aşağıdaki resimde sıra ile bir dizi içine alınmış; nota sayısını bir arttırarak for döngüsü içerisinde tanımlanmıştır.

Deneyap Kart ile buzzer’dan notaları çaldırabilmek için öncelikle Arduino IDE üzerinden gerekli kütüphane (<Tone32.h>) eklenir. Sonrasında kod bloğunda ‘tone’ kodunu kullanırken sırası ile öncelikle atanan pin, çalınan frekans, çalma süresi ve atanan kanal yazılır. Eğitmen kodu öğrencilere açıklarken bu noktaya değinmeli ve aşağıdaki kodda kanal için ‘0’ kanalını bir değişkene atadığını (int buzzerKanal=0) belirtmelidir.



*Resim 2.15: Gözle-Uygula Etkinliği Örnek Kodu*

| **Dikkat** |
| --- |
| Eğitmen devrenin fiziksel kurulumunu ve programın yazımından önce **diziler** hakkında öğrencileri kısaca bilgilendirmelidir. Özellikle dizilerin aynı türden bilgileri saklamak için kullanılan veri yapıları olduğu ve belirli sayıda aynı türden veriyi bellekte saklamak için kullanılan değişkenler listesinden oluştuğu vurgulanmalıdır.  Ayrıca dizilerin tanımlanmasında köşeli parantezlerin ( [ ] ) kullanıldığı; dizi içerisindeki elemanlara yine köşeli parantezler ile erişim sağlandığı ve dizilerde ilk elemanın sayımının **sıfırdan (0)** başlandığı öğrencilere iletilmelidir. Dizilere başlangıç değer atamasının küme parantezleri ile yapılabileceğine dikkat çekilmelidir. |

### 1.5 Uygula – Şarkı Çalan Buzzer (Öğrenci 1)

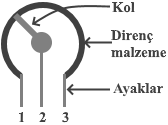
| Malzeme Listesi |
| --- |
| Deneyap Kart |
| Breadboard |
| Bağlantı kabloları |
| 100 ohm direnç |
| Buzzer |

Bu etkinlikteki amaç öğrencilerin basit bir şarkıyı buzzer’dan çaldırmasıdır. Öğrenciler buldukları/bildikleri basit bir şarkının notalarını kullanarak gerekli kodu yazar. Eğitmen öğrencilerin İnternet’ten nota aramasına izin verir. Fakat eğitmen nota okumasını gerektiren veya nota uzunluğu gibi etkinliği yavaşlatıcı durumlarda öğrencilere yardımcı olmalıdır.

### 1.6 Gözle ve Uygula - Potansiyometre ile LED parlaklığını ayarlama ve Seri Haberleşme

| Malzeme Listesi |
| --- |
| Deneyap Kart |
| Breadboard |
| Bağlantı kabloları |
| Buzzer |
| 10K Potansiyometre |

Yukarıda bahsedilen analog pinlerden gelen değerlerin okunmasında birçok devre elemanından yararlanılır. 0V ile 5V arasındaki ara değerleri de üreten bu devre elemanlarından biri de potansiyometredir. **Potansiyometre**, dışarıdan gelen fiziksel müdahaleler ile değeri değiştirilebilen dirençlerdir. Potansiyometrenin iki görevi bulunur. Bu görevler direnç ayarlama (ayarlı direnç) ve voltaj bölmedir. Aşağıdaki resimde potansiyometrenin içyapısı gösterilmiştir. Potansiyometrenin “10 K Ohm” olduğunu düşünelim. Potansiyometre ayarlı direnç olarak kullanmak istenirse 1 ve 2 numaralı bacakları kullanılarak devreye bağlanır. Kol 1 numaralı bacağa yaklaştırıldıkça direnç azalır ve “0 Ohm” değerine kadar iner. Kol 3 numaralı bacağa yaklaştırıldıkça direnç artar ve “10K Ohm” değerine kadar çıkar. Aynı işlem 2 ve 3 numaralı bacaklarla da yapılabilir. Fakat bu durumda kol 3 numaralı bacağa yaklaştıkça direnç azalacaktır ve 1 numaralı bacağa yaklaştıkça direnç artacaktır. Bu durumun nedeni bağlı iki bacak arasındaki direnç oluşturan malzemenin miktarıdır. Miktar arttıkça direnç artar ve tersinde ise azalır. Aşağıdaki resimde potansiyometrenin ayak yapısı görülmektedir.



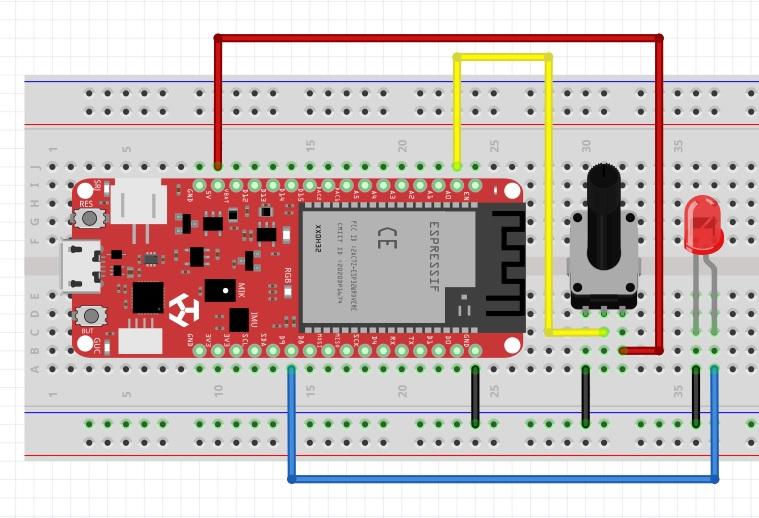
*Resim 2.16: Potansiyometre Ayak Yapısı*

Potansiyometre voltaj bölücü olarak kullanılmak istenirse bir güç kaynağının (5V olduğu varsayılsın) “+” ucu 1 numaralı bacağa, “-“ (GND) ucu ise 3 numaralı bacağa bağlanır. Çıktı değeri 2 numaralı bacaktan alınır. Potansiyometrenin kolu çevrilerek çıkış gerilimi değiştirilebilir. Böylece potansiyometre kullanılarak 5V’luk bir güç kaynağı kullanıldığında “0V-5V” arası voltaj değerleri elde edilir. Aşağıdaki resimde “10K Ohm” bir potansiyometre görülmektedir. Potansiyometrenin “10K” olduğu üst kısmında bulunan yazı ile belirtilmiştir. Aşağıdaki resimde ptansiyometre görülmektedir.

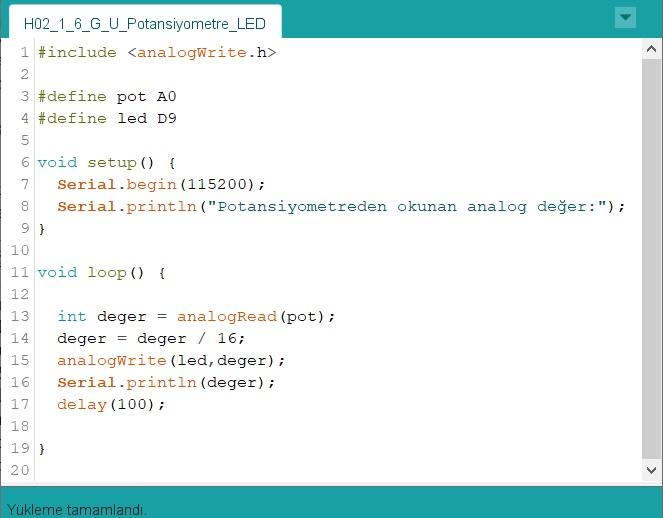


*Resim 2.17: Potansiyometre*

Aşağıdaki resimlerde örnek olarak verilen devre şeması ve Arduino IDE ile yazılmış program potansiyometre kullanılarak bir LED’in parlaklığını değiştirme uygulamasını içermektedir. Potansiyometrenin sol bacağı GND, sağ bacağı 5V ve orta bacağı A0 analog pinine bağlandığı devrede LED’in bir bacağı Deneyap Kart üzerrindeki D3 pinine, diğer bacağı da GND pinine bağlanmalıdır. Analog girdi 0-4095 arasında değer alırken Deneyap Kart üzerinde pwm pini olarak kullanılacak dijital pin (D3 pini) 0-255 arasında bir değer çıktı verebilmektedir. 0-4095 arasındaki değerin 0-255 arasındaki değere çevirmek için programda analog pinden alınan değer “deger” isimli bir değişkene atılır ve bu değer 16’ya bölünerek LED’e aktarılmak suretiyle parlaklık kontrolü yapılır. Eğitmen devre şemasını kurulumunu öğrencilere gösterir ve kurar; Arduino IDE ile yazılmış programın çalışmasını öğrencilere gösterir. Uygulama öğrencilere gösterilirken seri port ekranı açılıp öğrencilerin potansiyometre ile değerin değiştirildiğinde LED’in parlaklığının değişimini gözlemlemeleri sağlanır. Ardından öğrencilerin aynı devreyi kurmalarını ve kodu yazarak çalıştırmalarını sağlar.



*Resim 2.18: Gözle-Uygula Etkinliği Örnek Devre Şeması*

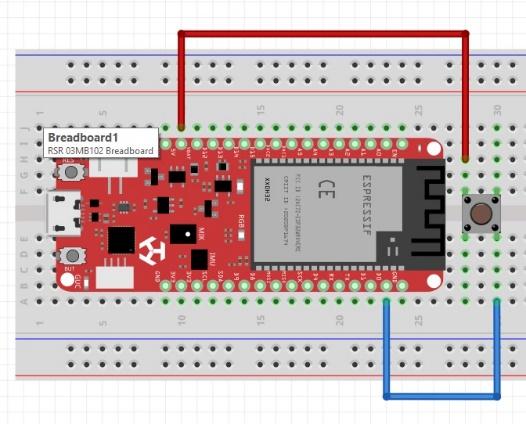


*Resim 2.19: Gözle-Uygula Etkinliği Örnek Kodu*

### 1.7 Gözle ve Uygula – Buton Kontrollü LED Yakma

| Malzeme Listesi |
| --- |
| Deneyap Kart |
| Breadboard |
| Bağlantı kabloları |
| Kırmızı LED |
| Buton |
| 220 Ohm direnç |
| 10 Kohm direnç |

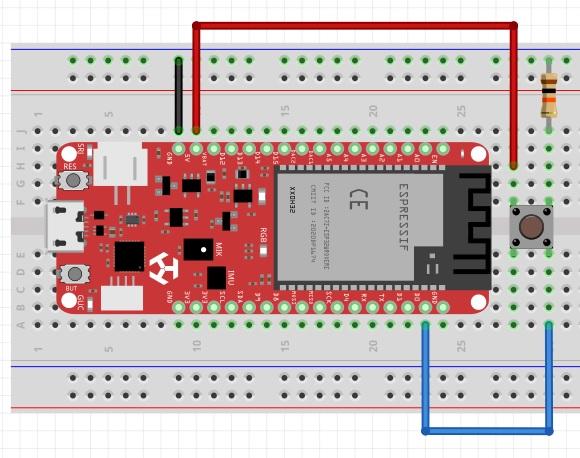
Deneyap Kart’ın pinlerinden “0” ve “1” olmak üzere dijital çıktı verilebildiği daha önce anlatılmıştı. Dijital pinler bunun yanında dijital girdi için de kullanılabilir. Yani dijital pinlere “0” (0V) veya “1” (5V) değerleri gönderip işlemler yapılabilir. Bu etkinlikteki amaç bir push butona basıldığında bir LED yakan devreyi tasarlayıp programı yazmaktır. Bu iş için bir push buton kullanılacaktır. Push butonun aşağıdaki şekilde gösterildiği gibi dört bacağı bulunur. Bunlardan karşılıklı ikisi (mavi çizgi ile bağlı gösterilenler) birbirlerine bağlıdır. Aynı şekilde karşıdaki iki bacak da birbirine bağlıdır. Fakat bu gruplar birbirine bağlı değildir. Butona basıldığında bu iki grup birbirine bağlanmış olur.

*Resim 2.20: Buton Bağlantı Örneği*

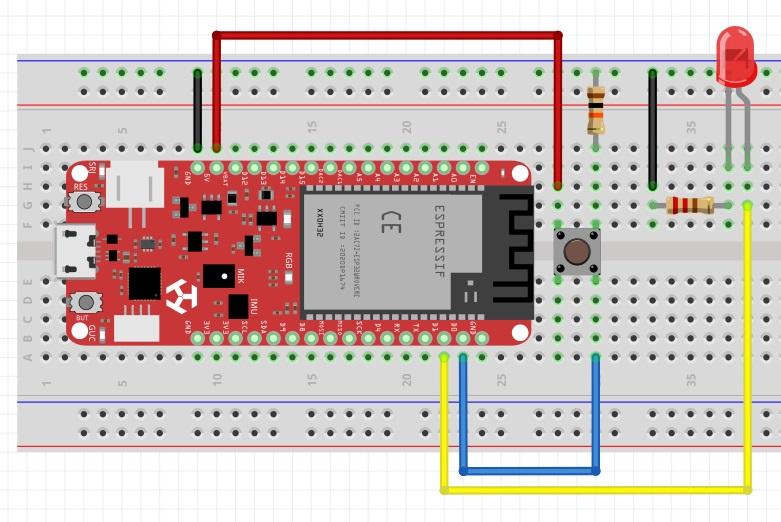
Yukarıdaki devre resminde botunun bir bacağına Deneyap Kart üzerinden “5 V” pinine verilmiştir, diğer bacağı ise D0 pinine bağlanmıştır. Butona basıldığında D0 pinine “1” bilgisi iletilecektir. Bu pinden gelen “1” değeri okutulabilir. Fakat bu bağlantının çeşitli sakıncaları bulunmaktadır. Deneyap Kart bu bağlantı ile butondan gelen değerleri yanlış okuyabilir. Yanlış okumaları engellemek için pull down direnç bağlantısı yapılmalıdır.

**Pull down** direnç bağlantısı aşağıdaki resimde gösterilmiştir. Butonun bacağına “10 Kohm” bir direnç ile GND bağlantısı yapılmıştır. Bu bağlantı sayesinde butona basılmadığında “0” değeri D0 pinine gönderilecektir. Böylece butona basıldığında D0 pinine “1”; butona basılmadığında D0 pinine “0” bilgisi gönderilmiş olur.



*Resim 2.21: Buton Bağlantı Örneği*

Aşağıdaki resimde görülen devre butona basıldığında D1 pinine bağlı LED’in yanması için tasarlanmıştır.



*Resim 2.22: Gözle-Uygula Etkinliği Örnek Devre Şeması*

Butona basıldığında D0 pininde “1” değeri okunur. D0 pininde “1” değeri okunduğunda D1 pininden “1” (5V) bilgisi gönderilip LED yakılmalıdır. Yukarıda devre şeması verilen uygulamanın Arduino IDE’de yazılmış program örneği aşağıdaki resimde verilmiştir. Program kodunda buton değerinin “1” olması şartıyla (if kontrol yapısı) LED’in yanma eylemi gerçekleştirmesi; diğer durumlarda ise (else komutu) LED’in sönme ifadesi eklenmiştir. Eğitmen devre şemasını kurulumunu öğrencilere gösterir ve kurar; Arduino IDE ile yazılmış programın çalışmasını öğrencilere gösterir. Ardından öğrencilerin aynı devreyi kurmalarını ve kodu yazarak çalıştırmalarını sağlar.



*Resim 2.23: Gözle-Uygula Etkinliği Örnek Kodu*

| **Dikkat** |
| --- |
| D2, D3 ve D8 pini özel kullanımı olan pinler olup, kod atma sırasında hata alımına sebep verebilecek pinlerdir. Bu pinlerin detaylı kullanımı https://docs.deneyapkart.org/ sayfasından inceleyebilirsiniz. |

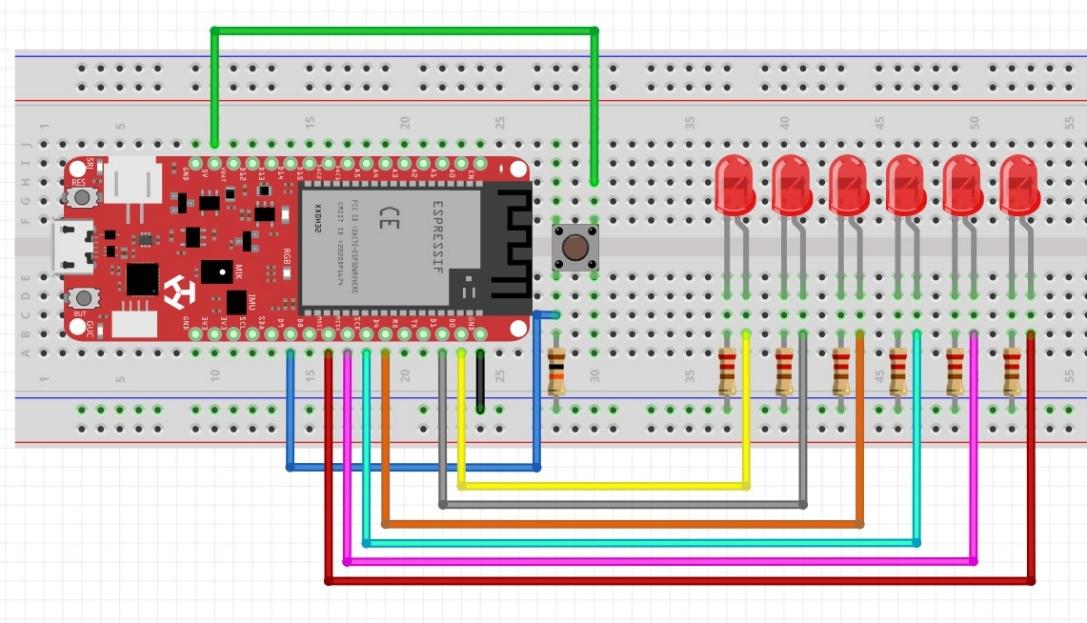
| **Dikkat** |
| --- |
| Eğitmen Arduino IDE içerisinde programın kodlarını öğrencilere gösterirken “**#if/else”** deyiminin koşullu ifadeleri yürütmek için kullanıldığını ve temel kod akışı üzerinde daha fazla denetim sağlamak için tercih edildiğini öğrencilere aktarmalıdır. “#if/else” deyiminde “if”in eğer, “else”in ise değil anlamına geldiği belirtilmelidir. Ayrıca “if” ve “else” in program içerisinde birlikte kullanıldığı; “else”in tek başına kullanılamadığı uyarısı yapılmalıdır. |

### 1.8 Uygula – Buton ile Soldan-Sağa ve Sağdan-Sola Hareket Eden LED Uygulaması

| Malzeme Listesi |
| --- |
| Deneyap Kart |
| Breadboard |
| Bağlantı kabloları |
| Kırmızı LED |
| Buton |
| 220 Ohm direnç |
| 10 Kohm direnç |

Bu uygulamada amaç LED’leri kullanarak ışığın soldan sağa ve sağdan sola kayan bir şekilde hareket ettiği hissini uyandırmaktır. Bunun için fiziksel olarak yan yana bulunan LED’ler buton yardımıyla sırayla açılıp kapatılmalıdır. Ardışık LED’lerdeki açılıp kapanma göz tarafından ışığın hareketi olarak algılanır. Butona bastıktan sonra ışıktaki hareket birinci LED’den başlayarak son LED’e kadar gitmeli, oradan da geriye ilk LED’e dönmelidir. Bu şekilde soldan sağa ve sağdan sola hareket işlemi sürekli devam etmelidir. Bu uygulamada öğrencilerin bilgisayar ve set başında çalışarak istenilen görevi yerine getirmeleri beklenmektedir. Öğrencilere eğitmen tarafından uygulamayı tamamlamak için yeterli süre verilir. Gerekli durumlarda eğitmen öğrencilere rehberlik etmelidir. Öğrencilerin aşağıdaki resimlerde görülen devreye ve programa benzer bir program hazırlamaları beklenir.

| **Dikkat** |
| --- |
| Eğitmen Arduino IDE içerisinde programın kodlarını öğrencilere gösterirken **#return** deyiminin bir fonksiyonun işlemi bittikten sonra çağrıldığı yere değer döndürmesi anlamı taşıdığı ve o fonksiyonun çalışmasını durduğunu hatırlatılması yerinde olacaktır. Uygulamada butona basılmadığı durumda “else” ifadesi ile LED’lerin başlangıç durumuna geleceği belirtilmelidir. |



*Resim 2.24: Uygula Etkinliği Örnek Devre Şeması*



*Resim 2.25: Uygula Etkinliği Örnek Kodu*

## 2. ADIM: TASARLA ve ÜRET

### 2.1 Tasarla- Bir Buton Dört Hareketli LED Uygulaması

Bu etkinlikte 4 adet LED bir buton tarafından çalıştırılacaktır. Devrede butonun basılma şekline göre LED’ler ile animasyon yapılacaktır. Devrenin çalışma şekli aşağıdaki gibi olmalıdır:

(i) Butona birinci defa basıldığında 1, 2, 3 ve 4 numaralı LED’ler sırasıyla yanıp söner. Devamında sadece 1 numaralı LED yanar ve yanık kalır.

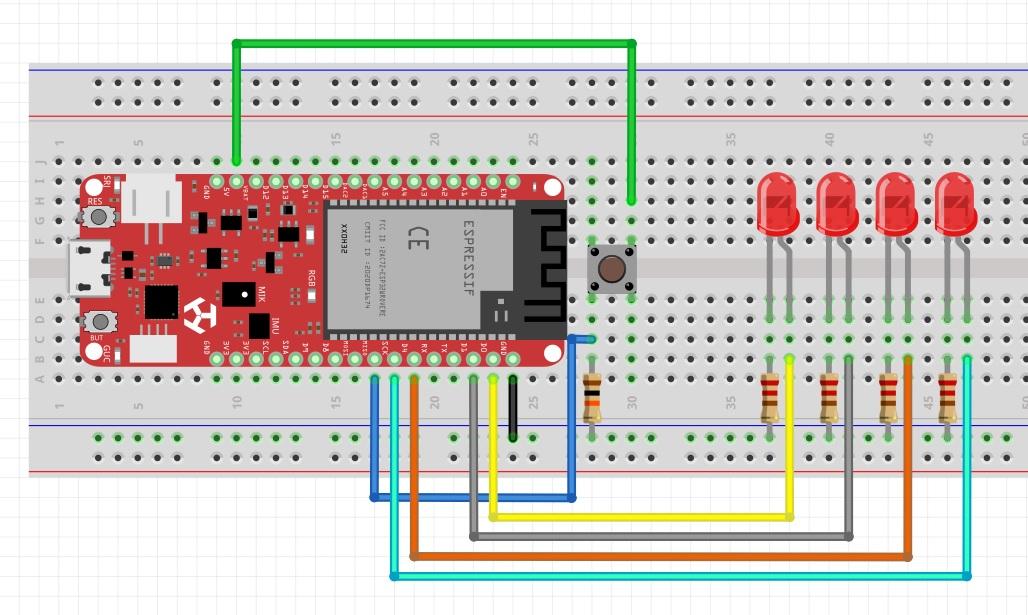
(ii) Butona ikinci defa basıldığında 1, 2, 3 ve 4 numaralı LED’ler sırasıyla yanıp söner. Devamında sadece 2 numaralı LED yanar ve yanık kalır.

(iii) Butona üçüncü defa basıldığında 1,2,3 ve 4 numaralı LED’ler sırasıyla yanıp söner. Devamında sadece 3 numaralı LED yanar ve yanık kalır.

(iv) Butona dördüncü defa basıldığında 1,2,3 ve 4 numaralı LED yanıp söner. Devamında sadece 4 numaralı LED yanar ve yanık kalır.

(v) Butona beşinci defa basıldığında bütün LED’ler söner ve butona bir defa basılmış gibi birinci adımdan yeniden başlanır.

Öğrenciler gruplar halinde oluşturacakları devreyi ve programı tartışırlar. Eğitmen gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir. Aşağıdaki resimde etkinlik örnek devre şeması görülmektedir.



*Resim 2.26: Tasarla Etkinliği Örnek Devre Şeması*

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

* Devrenin fiziksel olarak kurulumunda buton ve LED’ler için kullanılacak olan dirençlerin ohm değerleri neler olmalıdır?
* Devrenin fiziksel olarak kurulumunda LED’lerin Deneyap Kart pinlerine bağlantılarında nasıl bir sıra izlenmelidir?
* Arduino IDE içinde yazılacak olan kodlarda hangi tür kontrol yapıları kullanılmalıdır?
* Arduino IDE içerisinde yazılacak kodlarda butonların sıra ile basılması durumlarında LED’lerin yanma ve yanık kalma durumlarının algoritması nasıl olmalıdır?

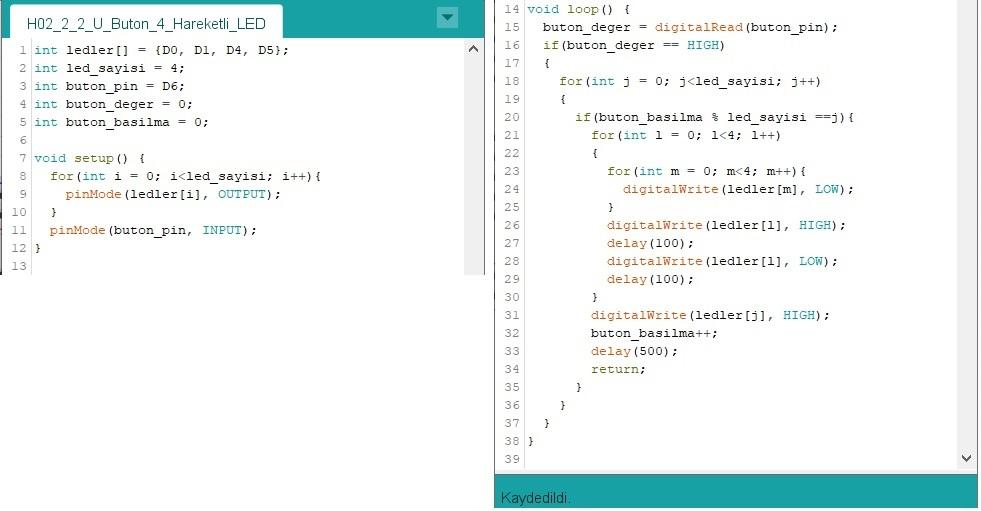
Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir.

* Devrenin fiziksel kurulumu için 220 ve 10 Kohm’luk dirençler kullanılmalı ve Arduino UNO üzerindeki dijital pin bağlantıları ile Void fonksiyonları öncesindeki tanımlamalar bağlantılardaki pin değerleri ile örtüşmelidir.
* Başlangıçta LED’lerin sönmesi ve daha sonra sıra ile yanma ve butonun basılma sayısı ile yanık kalacak LED’lerin sıralaması iç içe planlanmalıdır.
* Butonun basılma sayısı ve butonun basıldığında aktif çalışması durumlarının gerektireceği ifadelerin if kontrol yapısı ile tanımlanabileceği düşünülebilir.
* Butonun basılma sayısı ile LED’leri yakma ve söndürme işlemleri için for döngü yapıları kullanılmalıdır.

Bu etkinlik için kullanılabilecek devre tasarımı yukarıda verilmiştir.

### 2.2 Üret- Bir Buton Dört Hareketli LED Uygulaması

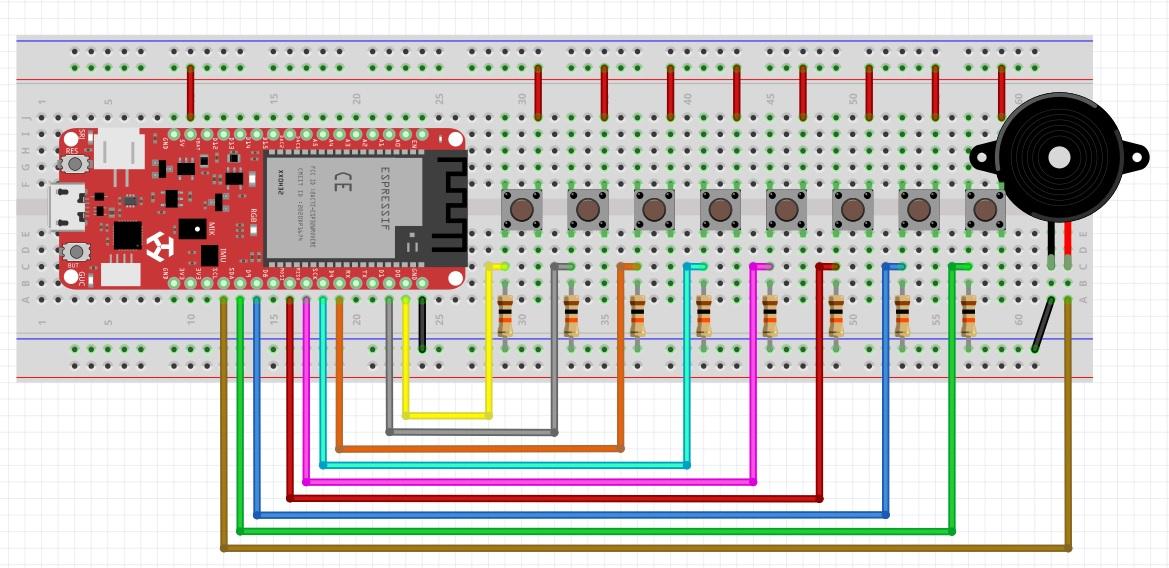
Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Özellikle “Arduino IDE” içerisinde yazılacak program kodlarında iç içe döngülerin tasarlanmasında ve çözüm için algoritmanın oluşturulma sürecinde eğitmenin rehberliği öğrencilerin etkinliği gerçekleştirmesinde önemli olacaktır. Öğrencilerin aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenir.



*Resim 2.27: Üret Etkinliği Örnek Kod*

### 2.3 Tasarla- Butonlar ile Piyano Yapımı

Bu etkinlikteki amaç setler içerisinde yer alan butonları ve Buzzer’ı kullanarak bir piyano tasarlamaktır. Bunun için C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La), B (Si), C\_ (ince Do) notalarının her birinin frekans değerlerini çalacak şekilde butonlar ve Buzzer ile Denayap Kart bağlantıları gerçekleştirilerek devre kurulur. Aşağıdaki resimde örnek bir devre şeması verilmiştir. Devrenin kurulumunda öğrencilerin butonlar ve Buzzer için GND ve dijital pin değerlerinin bağlantısını doğru bir şekilde gerçekleştirmeleri önemlidir. Arduino IDE içerisinde öğrencilerin yazacakları kodlarda değişken tanımlarına, if kontrol yapısının kurulmasına ve tone fonksiyonunun doğru şekilde kullanılmasına dikkat edilmelidir.



*Resim 2.28: Tasarla Etkinliği Örnek Devre Şeması*

Öğrenciler gruplar halinde oluşturacakları devreyi ve programı tartışırlar. Eğitmen gerekli noktalarda yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat çözümü öğrencilere hazır olarak vermemelidir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

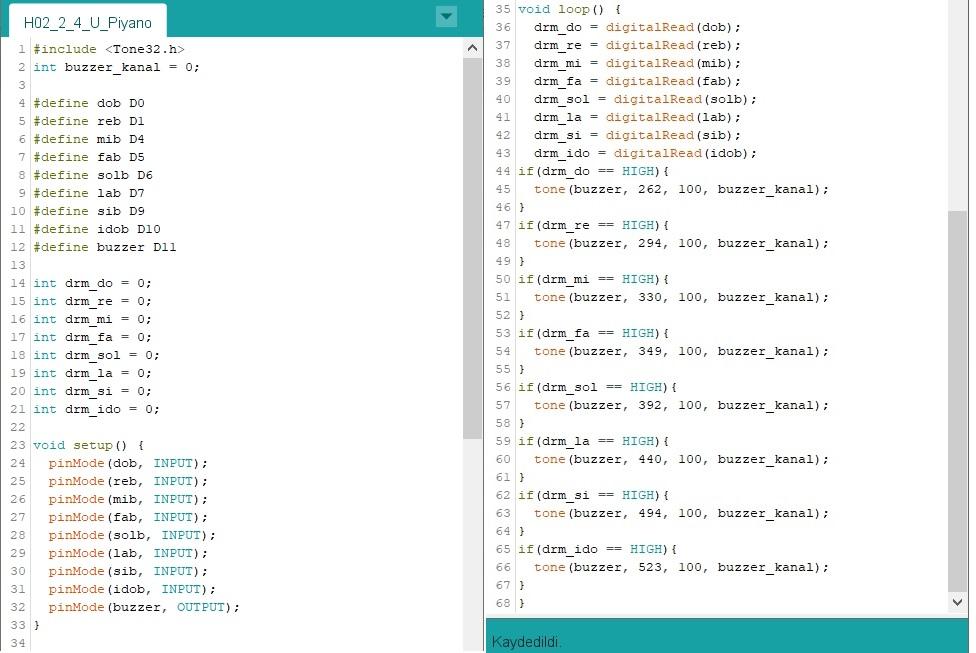
* Devre oluşturulurken butonların dirençleri ve “5V” güç bağlantıları Breadboard ve Deneyap Kart arasında nasıl yapılmalıdır?
* Devre oluşturulurken “Buzzer” bağlantısı “Breadboard” ve “Deneyap Kart” arasında nasıl yapılmalıdır?
* Arduino IDE içerisinde “Void” fonksiyonlarından önce Buzzer, buton bağlantı pinleri ve notaların ilk değer değişkenlerinin tanımlamaları nasıl yazılmalıdır?
* Set up fonksiyonu içerisinde hangi değişkenler giriş ve çıkış olarak tanımlanmalıdır?
* Loop fonksiyonu içerisinde notaların frekanslarını doğru şekilde Buzzer’da çalınmasını sağlayacak kontrol yapısı nasıl yazılmalıdır?
* tone fonksiyonu değerleri neler olmalıdır?

Fikir Üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

* Devre için buton bağlantıları Breadboard üzerinde Deneyap Kart içerisindeki pinlere sıralı şekilde bağlanmalıdır.
* Arduino IDE içerisinde nota değerlerinin başlangıç değerleri tanımlanmalıdır.
* Arduino IDE programında her bir buton için if kontrol yapısı her bir nota değeri için butonlardan okunmalıdır.
* Arduino IDE programında her bir butona karşılık olarak tone fonksiyonu ilgili Hertz cinsinden frekans değeri vermelidir.

### 2.4 Üret- Butonlar ile Piyano Yapımı

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin yukarıdaki devreye ilişkin olarak aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenir.



*Resim 2.29: Üret Etkinliği Örnek Kodu*

## 3. ADIM: DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Bugün tasarlayıp geliştirdikleri soldan sağa-sağdan sola yanıp sönen LED ve butonlar ile piyano uygulamaları için sınıf içerisinden öğrenciler seçilir ve aşağıdaki sorular üzerinden tartışma ortamı yaratılır:

* Devreler oluşturulurken neden farklı dirençler kullanıldı?
* Devrelerin fiziki kurulumunda en fazla nerelerde zorlandınız?
* Program algoritmalarını oluştururken hangi süreçlerde zorlandınız?
* Arduino IDE içerisinde yazılan kodlarda “#for” deyiminin kullanılmasının avantajları nelerdir?
* Arduino IDE içerisinde yazılan kodlarda “#if-else” deyiminin kullanılmasının avantajları nelerdir?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilebilir.