

Deney 2: Gösterge Uygulamaları

Deneyin Amacı:

Arduino kartı ile deney tablası kullanarak gösterge bağlantısını yapmayı, uygulama devresini kurmayı, giriş/çıkış portlarının kullanılmasını öğrenmek için gösterge devre ve modüllerinin çalışma yöntemlerinin, donanımlarının incelenmesi ve programlanmasının yapılmasıdır.

Deney Öncesi Yapılacak İşlemler:

Bu deneyde kullanılan LED göstergelerin, sıvı kristal göstergelerin (LCD) elektronik devre malzemelerinin işlevleri, kullanım özellikleri, malzemenin fiziksel görünümü, adının ve değerinin üzerindeki verilerden okunarak elde edilmesi öğrenilecektir. Burada verilen uygulama programları incelenecek, her komut satırının işlevi, açıklaması anlaşılacaktır.

Deneyde Yapılacak İşlemler:

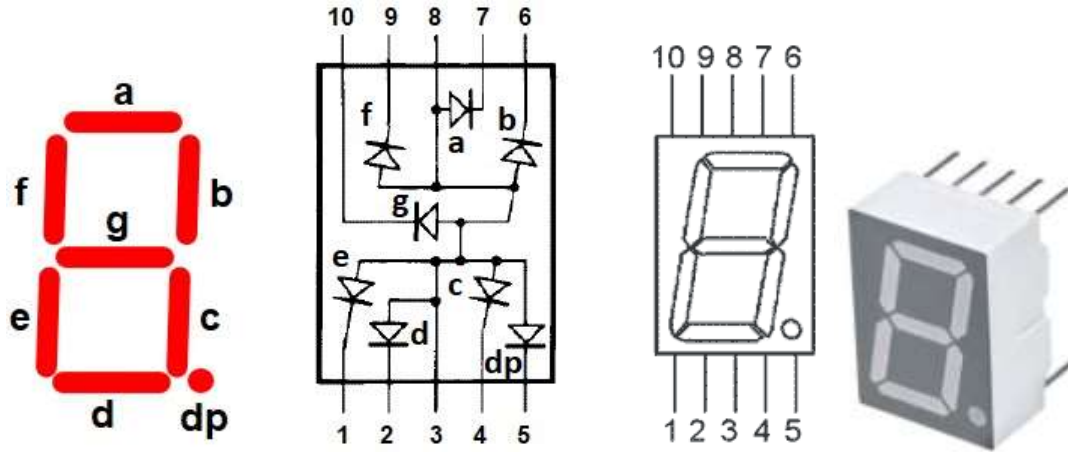
Burada verilen 7-parça LED gösterge ve 16x2 LCD ekranın nasıl bağlanacağı ve bir Arduino ile nasıl sürüleceği, örnek uygulama programları ile gösterilecektir.

2.a) LED Gösterge Uygulaması

Arduino ve 7-parça LED Gösterge Donanımı:

7-parça LED gösterge basit bir cihazdır. Şekil 1'de 7-parça LED göstergenin parça tanımları, iç bağlantısı, uç tanımları ve görünümü verilmiştir. Bu şekilde görüldüğü gibi birer uçları bağlanarak ortak yapılmış 8 LED (ondalık nokta (decimal point, dp 8. parçadır!) ile oluşmuştur. LED'lerin anotları ortak ise ortak anot (Common Anode, CA), katotları ortak ise ortak katot (Common Cathode, CC), 7-parça LED gösterge olarak adlandırılır. Sayıları oluşturmak için

Şekil 4'te gösterilen şekilde farklı parça kombinasyonlar kullanılarak düzenlenebilir.



Şekil 1. 7-parça LED göstergenin parça tanımları, iç bağlantısı, uç tanımları ve görünümü

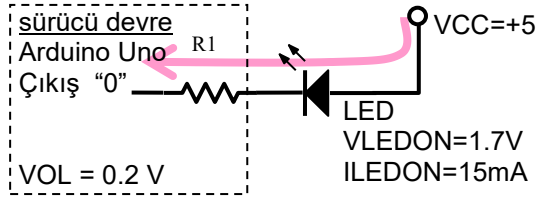
1. Bu uygulamada Arduino UNO kartı ile 7- parça LED göstergelyi bağlamak için Tablo 1'de verilen eşleşmeleri sağlamak için deney tablasını kullanarak aşağıda Şekil 3a'da verilen elektrik devresi Şekil 3b'deki montaj görüntüsündeki gibi kurulacaktır.

Tablo 1. Arduino UNO ile 7-parça LED gösterge uç eşleşmeleri

Arduino UNO	2	3	4	5	6	7	8
7-parça LED Gösterge	7 (a)	6 (b)	4 (c)	2 (d)	1 (e)	9 (f)	10 (g)

LED'leri doğrudan Arduino G/Ç uçlarına bağlamak, karta (mikrodenetleyiciye) veya LED'lere zarar verecektir! Zarar vermeyecek, doğru direnç değerini hesaplamak için Şekil 2'teki Ortak

Anot LED'in çalışması için eşdeğer devre kullanılır. LED 'lerin ortak anot ucu VCC'ye bağlanır. Bunun beraber, Arduino kart ile diğer bağlantıların her biri için parça uçları arasında Şekil 3'te verilen şekilde direnç kullanılır.

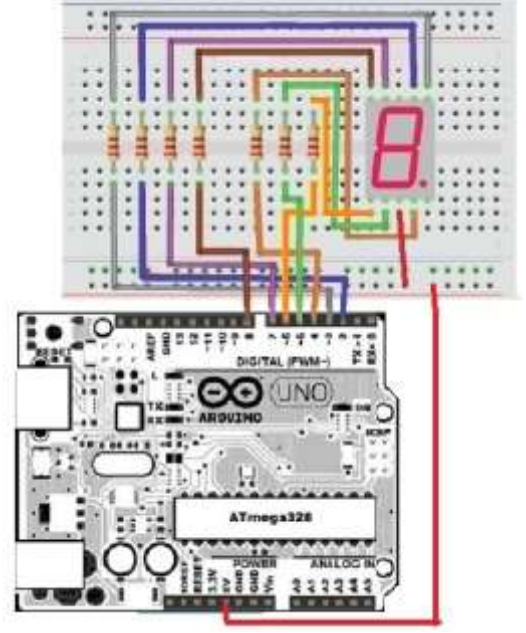
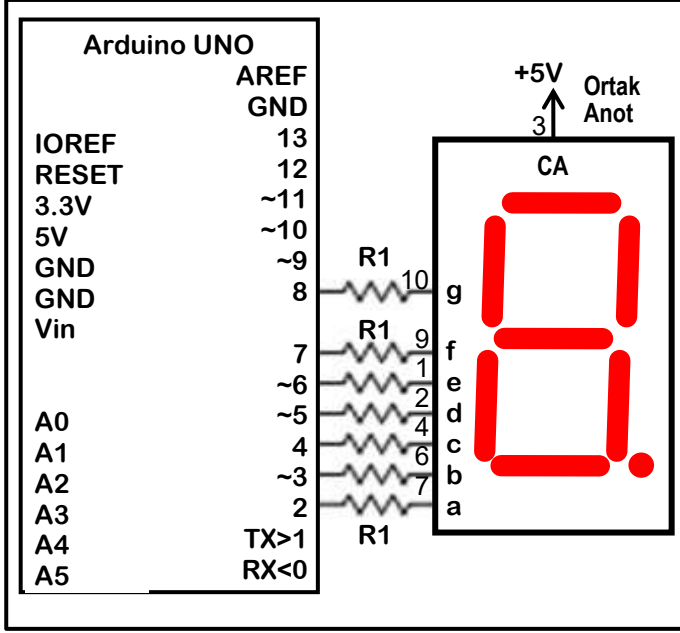


$$R1 = \frac{VCC - VLEDON - VOL}{ILEDON} = \frac{5V - 1.7V - 0.2V}{15mA} = 207\Omega$$

En yakın standart direnç $R1=220\Omega$

Şekil 2. Ortak Anot LED'in çalışması için eşdeğer devre

Arduino kart ile ortak anot LED gösterge eşlemeleri aşağıda Şekil-2'de verilmiştir.

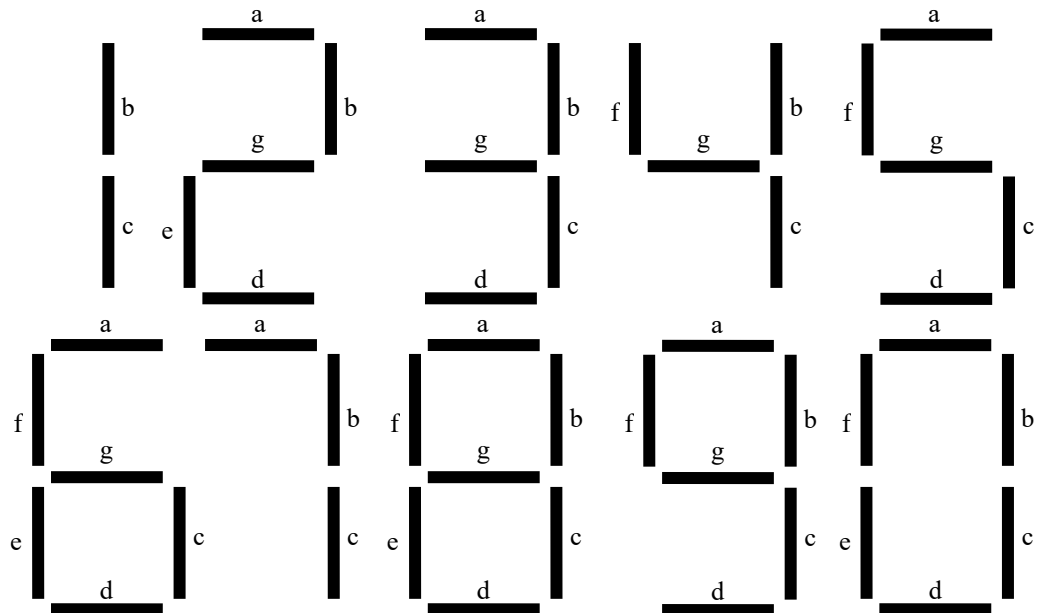


Şekil 3. (a) Elektrik Devresi

(b) Montaj Görüntüsü

Tablo 2. Ortak Anot 7-parça LED gösterge için veri dönüşüm tablosu

	Ortak Anot						
Sayı	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0



Şekil 4. Sayıların 7-parça LED göstergedeki şekil ve tanımları.

Arduino ve 7-parça LED Gösterge Yazılımı:

Bir basamak 7-parça LED göstergede, sürekli olarak 0'dan 9'a kadar sayan bir program.

1.Uygulama programı:

Program uzun, ayrıntılı bir yöntem kullanan ancak basit bir programdır.

```
// İSTÜN Bilgisayar/Yazılım/Mekatronik Müh.
// Arduino ile ortak anot LED gösterge "0-9" sayan basit bir program
// BL306 Mikroişlemciler ve Gömülü Sist. Lab.
// Deney 5: Gösterge Uygulamaları,
// 5.a) Arduino ile 7-parça LED Gösterge Uygulaması
// (c)Y.Doc.Dr.Tuncay UZUN

// Önce 7-parça LED Göstergeye bağlı Arduino uçları çıkış yapılıır
void setup() {
Serial.begin(9600);
  pinMode(2, OUTPUT); // (a) parçası
  pinMode(3, OUTPUT); // (b) parçası
  pinMode(4, OUTPUT); // (c) parçası
  pinMode(5, OUTPUT); // (d) parçası
  pinMode(6, OUTPUT); // (e) parçası
  pinMode(7, OUTPUT); // (f) parçası
  pinMode(8, OUTPUT); // (g) parçası
}

// Ortak anot için 0 = LED ON, 1 = LED OFF:
void loop() {
// Göstergeye '0' yaz
  digitalWrite(2,0);digitalWrite(3,0);digitalWrite(4,0);digitalWrite(5,0);
  digitalWrite(6,0);digitalWrite(7,0);digitalWrite(8,1);
  delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '1' yaz
  digitalWrite(2,1);digitalWrite(3,0);digitalWrite(4,0);digitalWrite(5,1);
  digitalWrite(6,1);digitalWrite(7,1);digitalWrite(8,1);
  delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '2' yaz
  digitalWrite(2,0);digitalWrite(3,0);digitalWrite(4,1);digitalWrite(5,0);
  digitalWrite(6,0);digitalWrite(7,1);digitalWrite(8,0);
  delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '3' yaz
  digitalWrite(2,0);digitalWrite(3,0);digitalWrite(4,0);digitalWrite(5,0);
  digitalWrite(6,1);digitalWrite(7,1);digitalWrite(8,0);
  delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '4' yaz
  digitalWrite(2,1);digitalWrite(3,0);digitalWrite(4,0);digitalWrite(5,1);
  digitalWrite(6,1);digitalWrite(7,0);digitalWrite(8,0);
  delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '5' yaz
  digitalWrite(2,0);digitalWrite(3,1);digitalWrite(4,0);digitalWrite(5,0);
  digitalWrite(6,1);digitalWrite(7,0);digitalWrite(8,0);
  delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '6' yaz
  digitalWrite(2,0);digitalWrite(3,1);digitalWrite(4,0);digitalWrite(5,0);
  digitalWrite(6,0);digitalWrite(7,0);digitalWrite(8,0);
  delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '7' yaz
  digitalWrite(2,0);digitalWrite(3,0);digitalWrite(4,0);digitalWrite(5,1);
  digitalWrite(6,1);digitalWrite(7,1);digitalWrite(8,1);
  delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '8' yaz
```

```

digitalWrite(2,0);digitalWrite(3,0);digitalWrite(4,0);digitalWrite(5,0);
digitalWrite(6,0);digitalWrite(7,0);digitalWrite(8,0);
delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
// Göstergeye '9' yaz
digitalWrite(2,0);digitalWrite(3,0);digitalWrite(4,0);digitalWrite(5,0);
digitalWrite(6,1);digitalWrite(7,0);digitalWrite(8,0);
delay(1000); //yazılan sayının görülmesi için 1 saniye bekle
}

```

2.Uygulama programı:

Yukarıdaki programda 7-parça LED gösterge bit desenleri için tekrarlanan yapıda satırlar içermektedir. Bu tekrarlayan satırları oluşturmak için 2 boyutlu bir dizi ve sayma işleminde "for" döngüsü kullanılarak daha kısa ve güzel bir programdır.

```

// İSTÜN Bilgisayar/Yazılım/Mekatronik Müh.
// Arduino ile ortak anot LED gösterge "0-9" sayan iyi bir program
// BL306 Mikroişlemciler ve Gömülü Sist. Lab.
// Deney 5: Gösterge Uygulamaları,
// 5.a) Arduino ile 7-parça LED Gösterge Uygulaması
// (c)Y.Doc.Dr.Tuncay UZUN

// Önce 7-parça LED Göstergeye bağlı Arduino uçları çıkış yapılıır
void setup() {
Serial.begin(9600);
pinMode(2, OUTPUT); // (a) parçası
pinMode(3, OUTPUT); // (b) parçası
pinMode(4, OUTPUT); // (c) parçası
pinMode(5, OUTPUT); // (d) parçası
pinMode(6, OUTPUT); // (e) parçası
pinMode(7, OUTPUT); // (f) parçası
pinMode(8, OUTPUT); // (g) parçası
}

// 0'dan 9'a kadar LED gösterge kod desenlerinin tablosu
// Ortak anot için 0 = LED ON, 1 = LED OFF:
byte seven_seg_digits[10][7] = {
// a b c d e f g parça kodu
{ 0, 0, 0, 0, 0, 0, 1 }, // "0" için parça deseni
{ 1, 0, 0, 1, 1, 1, 1 }, // "1" için parça deseni
{ 0, 0, 1, 0, 0, 1, 0 }, // "2" için parça deseni
{ 0, 0, 0, 0, 1, 1, 0 }, // "3" için parça deseni
{ 1, 0, 0, 1, 1, 0, 0 }, // "4" için parça deseni
{ 0, 1, 0, 0, 1, 0, 0 }, // "5" için parça deseni
{ 0, 1, 0, 0, 0, 0, 0 }, // "6" için parça deseni
{ 0, 0, 0, 1, 1, 1, 1 }, // "7" için parça deseni
{ 0, 0, 0, 0, 0, 0, 0 }, // "8" için parça deseni
{ 0, 0, 0, 0, 1, 0, 0 } // "9" için parça deseni
// 2 3 4 5 6 7 8 Arduino UNO uç numaraları
};

// Tablodan bakma yöntemiyle sayılan değer, 7-parça desen tanımını
// matristen alarak porta yazan altprogram
void sevenSegWrite(byte digit) {
byte pin = 2;
for (byte segCount = 0; segCount < 7; ++segCount) {
digitalWrite(pin, seven_seg_digits[digit][segCount]);
++pin;
}
}

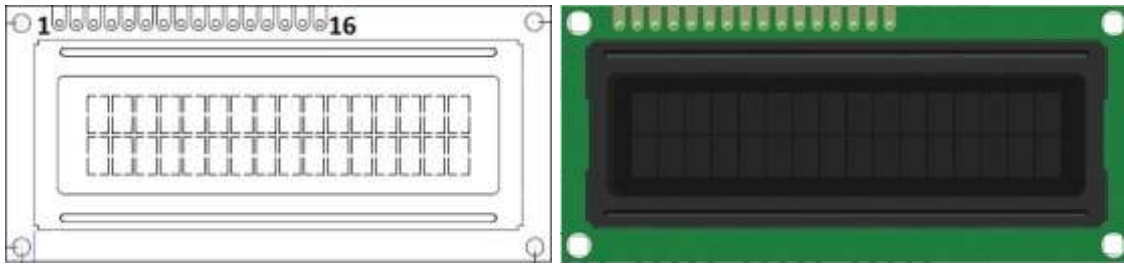
```

```
// 7-parça LED göstergede 1 saniye aralıkla "0-9" sayılması
void loop() {
  for (byte count = 0; count < 10; ++count) { // 0-9 Saymaya başla
    sevenSegWrite(count); // Sayılan değeri göstergeye yaz
    Serial.println(count);
    delay(1000); // Sayılan değerin görülebilmesi için 1 saniye bekle
  }
}
```

2.b) LCD Ekran Uygulaması

Arduino ve LCD Ekran Donanımı:

LCD ekran karmaşık, programlanabilir modül, elektronik bir cihazdır. Şekil 5'de 16x2, 2 satırda 16 karakter yazılabilen LCD göstergenin uç numaraları ve görünümü verilmiştir. Tablo 3'te 16x2 LCD gösterge uç tanımları verilmiştir. Tablo 4'te ise Arduino UNO ile gösterge uç eşleşmeleri verilmiştir.



Şekil 5. 16x2 LCD göstergenin uç numaraları ve görünümü

Tablo 3. 16x2 LCD gösterge uç tanımları

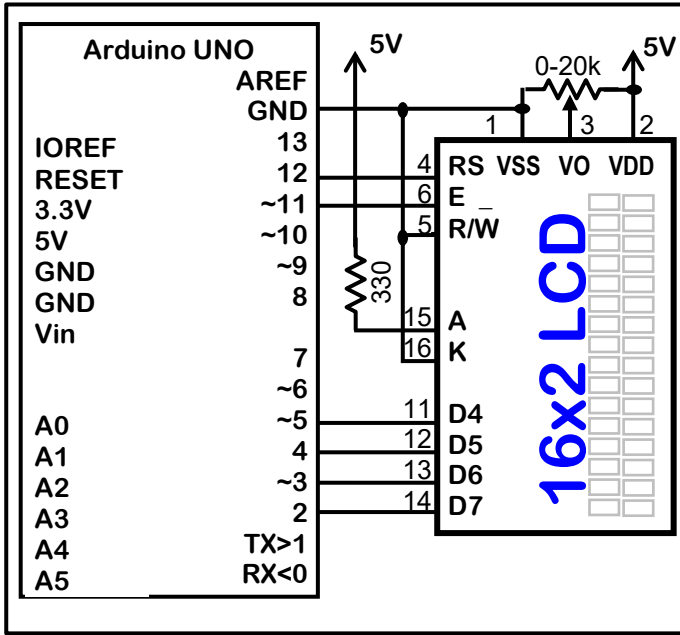
16x2 LCD	Uç Adı	Açıklama
1	VSS	GND
2	VDD	5V
3	VO	LCD için Güç Kaynağı
4	RS	Yazmaç Seçimi (H:Veri / L:Komut
5	R/W	Okuma/Yazma (H:Read / L:Write)
6	E	İzin (Enable)
7-14	D0-D7	Veri Yolu uçları
15 / 16	A / K	LCD Arka Işık Anot / Katot

Tablo 4. Arduino UNO ile gösterge uç eşleşmeleri

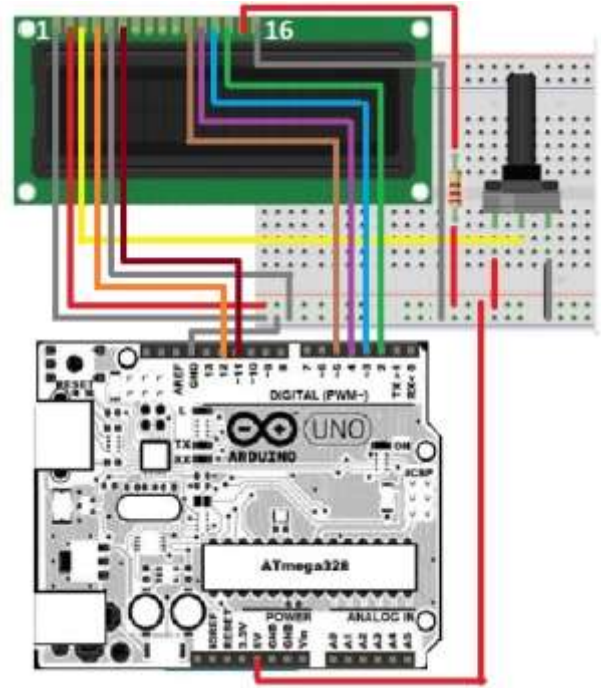
Arduino UNO	2	3	4	5	11	12
16x2 LCD	14 (D7)	13 (D6)	12 (D5)	11 (D4)	6 (E)	4 (RS)

Arduino ve LCD ekran Yazılımı:

16 karakter 2 satır LCD ekranda ilk olarak "Merhaba Dünya" yazan, sonra sürekli olarak geçen süreyi saniye olarak ekrana ve seri porta yazan bir programdır. Bu program, Arduino IDE içinde hazır bulunan, yükleme gerektirmeyen "LiquidCrystal" kitaplığını kullanan LCD ekran kullanımı için bir başlangıç programıdır. Bu programın benzeri değişik örnek programlara Arduino IDE ana menüden File>>Examples>> LiquidCrystal alt menüleri kısmında bulunmaktadır. Ayrıca <https://docs.arduino.cc/learn/electronics/lcd-displays/> adresinde LCD ile Arduino donanımı ve yazılımı hakkında ek bilgileri de görebilirsiniz. Buradaki "LiquidCrystal" kitaplığı, metin tabanlı LCD'lerin çoğunda bulunan Hitachi HD44780 yonga setini temel alan LCD ekranları kontrol etmenize olanak tanır.



Şekil 6. (a) Elektrik Devresi



(b) Montaj Görüntüsü

1.Uygulama programı:

Aşağıda, Tablo 5'te "LiquidCrystal" kitaplığında bulunan komutlar verilmiştir. Bu komutların işlevlerine deney öncesinde kısaca bakınız. Programda ilk olarak LCD ekranın çalışma şekli ve LCD ekran ile Arduino bağlantısı için uç tanımları yapılır. Sonra kütüphane komutlarını kullanarak ekran boyut ayarını 16x1 olarak yapar ve ekrana "Merhaba, Dünya!" yazar.

Tablo 5. Arduino UNO ile gösterge uç eşleşmeleri

LiquidCrystal()	LiquidCrystal(rs, enable, d4, d5, d6, d7) LiquidCrystal(rs, rw, enable, d4, d5, d6, d7) LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7) LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)
begin()	lcd.begin(cols, rows, charsize)
clear()	lcd.clear()
home()	lcd.home()
setCursor()	lcd.setCursor(col, row)
write()	lcd.write(data)
print()	lcd.print(data) lcd.print(data, BASE)
cursor()	lcd.cursor()
noCursor()	lcd.noCursor()
blink()	lcd.blink()
noBlink()	lcd.noBlink()
display()	lcd.display()
noDisplay()	lcd.noDisplay()
scrollDisplayLeft()	lcd.scrollDisplayLeft()
scrollDisplayRight()	lcd.scrollDisplayRight()
autoscroll()	lcd.autoscroll()
noAutoscroll()	lcd.noAutoscroll()
leftToRight()	lcd.leftToRight()
rightToLeft()	lcd.RightToLeft()
createChar()	lcd.createChar(num, data)

Örnek program:

```
#include <LiquidCrystal.h>
// LDC ekran çalışma şeklinin tanımlanması
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
  lcd.begin(16,1);
  lcd.print("Merhaba, Dünya!");
}
void loop() {}
```

2.Uygulama programı:

Aşağıdaki programda ilk olarak "LiquidCrystal" kütüphanesi ile Arduino uç tanımlarını yaparak bağlantısını sağlar. Sonra kütüphane komutlarını kullanarak ekran boyut ayarını yapar ve ekrana "Merhaba Dünya" yazar. Ana programda ise ekranın alt satır başına saniye olarak geçen zamanı yazar.

```
// İSTÜN Bilgisayar/Yazılım/Mekatronik Müh.
// Arduino ile LCD ekranda "Merhaba Dünya" yazan uygulama programı
// BL306 Mikroişlemciler ve Gömülü Sist. Lab.
// Dene 5: Gösterge Uygulamaları,
// 5.b) Arduino ile LCD Uygulaması
// (c)Y.Doc.Dr.Tuncay UZUN
#include <LiquidCrystal.h>
// ilk koşullandırmada kütüphane LCD arayüzü tanımlar
// Aynı zamanda Arduino uç bağlantılarını da belirler
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // LCD ekranın sütun ve satır sayısı:
  lcd.begin(16, 2);
  // LCD ekrana ilk mesajın ilk satıra yazılması:
  lcd.print("Merhaba, Dünya!");
}

void loop() {
  // imleci sütun 0, satır 1'e alt satırın başına ayarla
  lcd.setCursor(0, 1); lcd.print("t=");
  lcd.setCursor(2, 1); delay(100);
  // saniye olarak geçen zamanın ekrana yazılması
  lcd.print(millis() / 1000); lcd.print(" saniye");
}
```

Sorular:

1. 7-parça LED göstergelerin türler, çalışma yöntemleri ve özellikleri nelerdir?
2. Çok basamaklı 7-parça LED gösterge Arduino karta nasıl bağlanır? Nasıl Çalışır?
3. LCD ekran türleri, çalışma yöntemleri ve özellikleri nelerdir?
4. Uygulama programları çalışırken hangi uçtan hangi işaret gönderilir?
5. Uygulama programının başlangıcında bulunan "setup" fonksiyonu ne için kullanılır?
6. Uygulama programının son kısmında bulunan "loop" fonksiyonu ne için kullanılır?
7. "LiquidCrystal" komutunun özelliği ve çalışma şekli nasıldır?
8. "lcd.begin" komutunun özelliği ve çalışma şekli nasıldır?
9. "lcd.print()"komutunun özelliği ve çalışma şekli nasıldır?
- 10."lcd.setCursor(0, 1)" komutunun özelliği ve çalışma şekli nasıldır?