

Bu kitaba sığmayan daha neler var!

eba
www.eba.gov.tr



Karekodu okut, bu kitapla
ilgili EBA içeriklerine ulaş!

Teknoloji Yolculuğumuz



**BU DERS KİTABI MİLLÎ EĞİTİM BAKANLIĞINCA
ÜCRETSİZ OLARAK VERİLMİŞTİR.
PARA İLE SATILAMAZ.**

ISBN: 978-975-11-7988-3

Bandrol Uygulamasına İlişkin Usul ve Esaslar Hakkında Yönetmeliğin Beşinci Maddesinin
İkinci Fıkrası Çerçevesinde Bandrol Taşınması Zorunlu Değildir.

ELEKTRİK - ELEKTRONİK TEKNOLOJİSİ ALANI İLERİ MİKRODENETLEYİCİ UYGULAMALARI 11-12 DERS MATERYALI

ELEKTRİK - ELEKTRONİK TEKNOLOJİSİ ALANI

İLERİ

MİKRODENETLEYİCİ

UYGULAMALARI

DERS MATERYALI

11-12



MESLEKİ VE TEKNİK ANADOLU LİSESİ



MESLEKİ VE TEKNİK ANADOLU LİSESİ

ELEKTRİK - ELEKTRONİK TEKNOLOJİSİ ALANI

İLERİ MİKRODENETLEYİCİ UYGULAMALARI

11

Ders Materyali

YAZARLAR

Dr. Zafer ÖZTÜRK

Esra ÖVEZ

Mehmet GÖVERDİK

Ünsal Süleyman KESER

Yalçın TIRAŞ



MİLLÎ EĞİTİM BAKANLIĞI YAYINLARI	9375
YARDIMCI VE KAYNAK KİTAPLAR DİZİSİ	3035

Her hakkı saklıdır ve Millî Eğitim Bakanlığına aittir.
Ders materyalinin metin, soru şekilleri kısmen de olsa hiçbir surette alınıp yayımlanamaz.

HAZIRLAYANLAR

DİL UZMANI	Zülfü MUTLU
PROGRAM GELİŞTİRME UZMANI	Ergül SİRKINTI
REHBERLİK UZMANI	Fatih DÜĞENCİ
ÖLÇME VE DEĞERLENDİRME UZMANI	Mustafa ÇELİK
GÖRSEL TASARIM UZMANI	Birgül AKBABA

ISBN: 978-975-11-7988-3

Millî Eğitim Bakanlığının 24.12.2020 gün ve 18433886 sayılı oluru ile
Meslekî ve Teknik Eğitim Genel Müdürlüğünce ders materyali olarak hazırlanmıştır.



İSTİKLÂL MARŞI

Korkma, sönmez bu şafaklarda yüzen al sancak;
Sönmeden yurdumun üstünde tüten en son ocak.
O benim milletimin yıldızıdır, parlayacak;
O benimdir, o benim milletimindir ancak.

Çatma, kurban olayım, çehreni ey nazlı hilâl!
Kahraman ırkıma bir gül! Ne bu şiddet, bu celâl?
Sana olmaz dökülen kanlarımız sonra helâl.
Hakkıdır Hakk'a tapan milletimin istiklâl.

Ben ezelden beridir hür yaşadım, hür yaşarım.
Hangi çılgın bana zincir vuracakmış? Şaşarım!
Kükremiş sel gibiyim, bendimi çiğner, aşarım.
Yırtarım dağları, enginlere sığmam, taşarım.

Garbın âfâkını sarmışsa çelik zırhlı duvar,
Benim iman dolu göğsüm gibi serhaddim var.
Ulusun, korkma! Nasıl böyle bir imanı boğar,
Medeniyet dediğin tek dişi kalmış canavar?

Arkadaş, yurduma alçakları uğratma sakın;
Siper et gövdeni, dursun bu hayâsızca akın.
Doğacaktır sana va'dettiği günler Hakk'ın;
Kim bilir, belki yarın, belki yarından da yakın.

Bastığın yerleri toprak diyerek geçme, tanı:
Düşün altındaki binlerce kefensiz yatanı.
Sen şehit oğlusun, incitme, yazıktır, atanı:
Verme, dünyaları alsan da bu cennet vatanı.

Kim bu cennet vatanın uğruna olmaz ki feda?
Şüheda fışkıracak toprağı sıksan, şüheda!
Cânı, cânânı, bütün varımı alsın da Huda,
Etmesin tek vatanımdan beni dünyada cüda.

Ruhumun senden İlâhî, şudur ancak emeli:
Değmesin mabedimin göğsüne nâmahrem eli.
Bu ezanlar -ki şehadetleri dinin temeli-
Ebedî yurdumun üstünde benim inlemeli.

O zaman vecd ile bin secde eder -varsa- taşım,
Her cerâhamdan İlâhî, boşanıp kanlı yaşım,
Fışkırır ruh-ı mücerret gibi yerden na'sım;
O zaman yükselerek arşa değer belki başım.

Dalgalan sen de şafaklar gibi ey şanlı hilâl!
Olsun artık dökülen kanlarımın hepsi helâl.
Ebediyyen sana yok, ırkıma yok izmihlâl;
Hakkıdır hür yaşamış bayrağımın hürriyet;
Hakkıdır Hakk'a tapan milletimin istiklâl!

Mehmet Âkif Ersoy

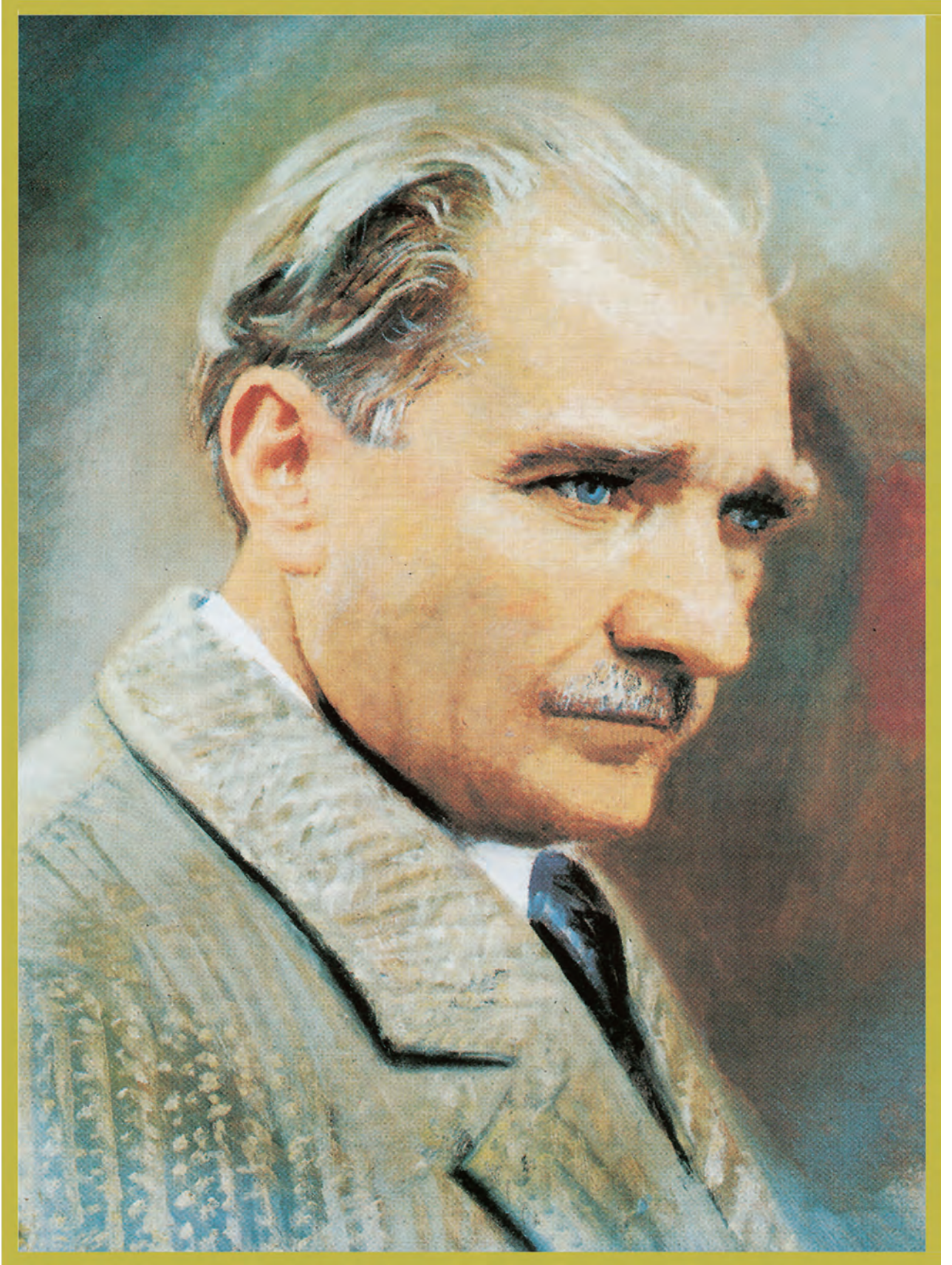
GENÇLİĞE HİTABE

Ey Türk gençliği! Birinci vazifen, Türk istiklâlini, Türk Cumhuriyetini, ilelebet muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin en kıymetli hazinendir. İstikbalde dahi, seni bu hazineden mahrum etmek isteyecek dâhilî ve hâricî bedhahların olacaktır. Bir gün, istiklâl ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için, içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin! Bu imkân ve şerait, çok namüsaît bir mahiyette tezahür edebilir. İstiklâl ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elîm ve daha vahim olmak üzere, memleketin dâhilinde iktidara sahip olanlar gaflet ve dalâlet ve hattâ hıyanet içinde bulunabilirler. Hattâ bu iktidar sahipleri şahsî menfaatlerini, müstevlîlerin siyasî emelleriyle tevhit edebilirler. Millet, fakr u zaruret içinde harap ve bîtap düşmüş olabilir.

Ey Türk istikbalinin evlâdı! İşte, bu ahval ve şerait içinde dahi vazifen, Türk istiklâl ve cumhuriyetini kurtarmaktır. Muhtaç olduğun kudret, damarlarındaki asil kanda mevcuttur.

Mustafa Kemal Atatürk



MUSTAFA KEMAL ATATÜRK

İÇİNDEKİLER

DERS MATERYALİNİN TANITIMI	12
1. ÖĞRENME BİRİMİ: MİKRODENETLEYİCİ İLE DİJİTAL UYGULAMALAR	16
1.1. Işık Efekt Devresi	24
1.2. LED Animasyon Devresi	29
1.3. Şerit RGB LED Kontrol Devresi	33
1.4. LED'li Ay Yıldız Devresi	37
1.5. Kayan Yazı Devresi	43
1.6. Havada Yazı Devresi	49
1.7. Trafik Lamba Devresi	54
1.8. LCD Uygulamaları	60
1.8.1. Paralel İletişimli LCD Uygulaması	60
1.8.2. Seri İletişimli LCD Uygulaması	65
1.8.3. Grafik LCD Uygulaması	70
1.9. Kronometre Devresi	77
1.10. Dijital Saat Devresi	83
1.11. Alkışla Yanan Lamba Devresi	95
1.12. Renk Algılama Devresi	99
1.13. Şifreli Kilit Devresi	105
1.14. Dijital Hesap Makinesi Devresi	113
1.15. Elektronik Piyano Devresi	120
1.16. Elektronik Plc Devresi	126
ÖLÇME VE DEĞERLENDİRME	135
2. ÖĞRENME BİRİMİ: MİKRODENETLEYİCİ İLE ANALOG UYGULAMALAR ...	138
2.1. Dijital Termometre Devresi	140
2.2. Mesafe Ölçer Devresi	144
2.3. Gaz Dedektörü Devresi	147
2.4. Motor Hız Kontrol Devresi	154
2.5. Dijital Voltmetre Devresi	159
ÖLÇME VE DEĞERLENDİRME	165
EK-A: PROJE UYGULAMALARI	167
1. AVometre Projesi	169
2. Engel Algılayan Robot Projesi	177
3. PV Panel ile Sokak Aydınlatması Projesi	181
4. SMD Lehimleme ve Sökme Cihazı Projesi	186
5. Göstergeli Zaman Ayarlı Dijital Grup Priz Projesi	191
6. Dimmerlenen Uzaktan Kontrollü Lamba Projesi	196
7. Robot Kol Kontrol Projesi	200
8. Termometreli Kayan Yazılı Saat Devre Projesi	204
9. Ayarlı DC Güç Kaynağı Projesi	212
10. ARDUINO Deney Seti Projesi	218

EK-B: PROJE UYGULAMALARI	223
1. Proje: Gelişmiş Dijital Saat	226
2. Proje: Dijital Pano	228
3. Proje: Uzaktan Kumandalı Robot	230
4. Proje: Bilimsel Hesap Makinesi	232
5. Proje: PV'li Aydınlatma	234
6. Proje: Direnç Ölçümü	236
7. Proje: Alarmlı Şifreli Kilit	237
8. Proje: Trafik Kavşak Kontrol	238
9. Proje: Işık ve Ses Etki	239
10. Proje: Bilgisayar Kontrollü Kayan Yazı	240
11. Proje: Dron	241
 EK-C: ARM32 MİKRODENETLEYİCİLER	 245
ARM SERİSİ MİKROİŞLEMCİLER	246
C.1. ARM (Advanced RISC Machine)	246
C.2. ARM32 Serisi Mikroişlemcilerin Yapısı ve Çeşitleri	246
C.3. Cortex – M Serisi Mikrodenetleyiciler	246
C.4. ARMKEIL Yazılımın İndirilmesi ve Kurulması	248
C.5. ARM32 Kontrol Kartı Sürücüsünün Yüklenmesi	250
C.6. TIVAWARE Yazılımının Yüklenmesi ve Örnek Dosyanın Açılması	252
C.7. ARMKEIL Yazılımının Kaynak Kodlarının İncelenmesi	255
C.8. ARM32 ile İlgili Örnek Uygulamalar	257
1. Uygulama: LED Animasyon	257
2. Uygulama: RGB LED	259
3. Uygulama: Zamanlayıcı (Timer)	259
4. Uygulama: Buton LED	263
5. Uygulama: ADC Sensör	264
6. Uygulama: PWM	265
7. Uygulama: Nokia 5110 LCD Display	267
8. Uygulama: LCD'de Sensör Bilgilerinin Gösterilmesi	269
9. Uygulama: LCD'de DHT11 Sıcaklık ve Nem Sensör Bilgilerinin Gösterilmesi	270
10. Uygulama: LCD'de Dijital Saat	274
11. Uygulama: Kronometre	275
 UYGULAMA ETKİNLİĞİ	 276
 CEVAP ANAHTARI	 277
 KAYNAKÇA	 278

GİRİŞ

Teknolojinin gelişmesi ile birlikte elektronik sistemleri içerisinde kontrol sistemlerine gereksinimin artması sonucu mikrodenetleyicilere olan ihtiyaç gün geçtikçe artmaktadır. İlk zamanlarda PIC gibi mikrodenetleyiciler ile sistemlerin tasarımı yapılırken son yıllarda ARDUINO, PINGUINO, ARM32 gibi kart modellenmiş mikrodenetleyiciler ile tasarımlar yapılmaktadır. Teknolojinin ve yazılımların gelişmesi ile birlikte kullanıcılara oldukça pratik bir sistem tasarımının oluşturulmasına katkıda bulunmaktadır. Bu derste önceki sınıfta öğrenilen temel mikrodenetleyici konuları çeşitli projeler eşliğinde uygulanarak çalışmalar pekiştirilecektir. Aşağıda verilen tablolarda her iki dönem için yapılacak uygulamaların hangi haftada yapılacağı ve hangi puan ile değerlendirileceği görülmektedir. Uygulamalar ile ilgili yönergeler, öğrenme birimi içerisinde (giriş bölümünde) detaylı olarak verilmiştir.

HAFTA	PROJE	PROJE ADI	Mikrodenetleyici			Montaj		PUANLAMA	
			PIC Serisi	ARDUINO	ARM32	Breadboard	Baskı devre		
1	1	Uygulama 1.1. Işık Etki Devresi	X	X	-	X	X	30	100
1		Uygulama 1.2. LED Animasyon Devresi	X	X	-	X	X	40	
1		Uygulama 1.3. Şerit RGB LED Kontrol Devresi	X	X	-	X	X	30	
1	2	Uygulama 1.4. LED'li Ay Yıldız Devresi	X	X	-	X	X	40	100
1		Uygulama 1.5. Kayan Yazı Devresi	X	X	-	X	X	30	
1		Uygulama 1.6. Havada Yazı Devresi	X	X	-	X	X	30	
1	3	Uygulama 1.7. Trafik Lamba Devresi	X	X	-	X	X	40	100
2		Uygulama 1.8. LCD Uygulamaları	X	X	-	X	X	60	
1	4	Uygulama 1.9. Kronometre Devresi	X	X	-	X	X	50	100
1		Uygulama 1.10. Dijital Saat Devresi	X	X	-	X	X	50	
1	5	Uygulama 1.11. Alkışla Yanan Lamba Devresi	X	X	-	X	X	30	100
1		Uygulama 1.12. Renk Algılama Devresi	X	X	-	X	X	30	
1		Uygulama 1.13. Şifreli Kilit Devresi	X	X	-	X	X	40	
1	6	Uygulama 1.14. Dijital Hesap Makinesi Devresi	X	X	-	X	X	30	100
1		Uygulama 1.15. Elektronik Piyano Devresi	X	X	-	X	X	30	
2		Uygulama 1.16. Elektronik PLC Devresi	-	X	-	X	X	40	

HAFTA	PROJE	PROJE ADI	Mikrodenetleyici			Montaj		PUANLAMA	
			PIC Serisi	ARDUINO	ARM32	Breadboard	Baskı devre		
1	7	Uygulama 2.1. Dijital Termometre Devresi	X	X	-	X	X	60	100
1		Uygulama 2.2. Mesafe Ölçer Devresi	X	X	-	X	X	40	
1	8	Uygulama 2.3. Gaz Dedektörü Devresi	X	X	-	X	X	30	100
1		Uygulama 2.4. Motor Hız Kontrol Devresi	X	X	-	X	X	30	
1		Uygulama 2.5. Dijital Voltmetre Devresi	X	X	-	X	X	40	
1	9	Uygulama A.1. AVOMETRE Projesi	-	X	-	-	X	50	100
1		Uygulama A.2. Engel Algılayan Robot Projesi	-	X	-	-	X	50	
2	10	Uygulama A.3. PV Panel ile Sokak Aydınlatması Projesi	-	X	-	-	X	50	100
1		Uygulama A.4. SMD Lehimleme ve Sökme Cihazı Proje	-	X	-	-	X	50	
1	11	Uygulama A.5. Göstergeli Zaman Ayarlı Dijital Grup Priz P.	-	X	-	-	X	50	100
1		Uygulama A.6. Dimmerlenen Uzaktan Kontrollü Lamba P.	-	X	-	-	X	50	
1	12	Uygulama A.7. Robot Kol Kontrol Projesi	-	X	-	-	X	50	100
1		Uygulama A.8. Termometreli Kayan Yazılı Saat Devre P.	-	X	-	-	X	50	
1	13	Uygulama A.9. Ayarlı DC Güç Kaynağı Projesi	-	X	-	-	X	50	100
1		Uygulama A.10. ARDUINO Deney Seti Projesi	-	X	-	-	X	50	
3	14	Uygulama B. Proje Uygulaması	X	X	X	-	X	100	100
1	15	Uygulama C. ARM32 Uygulaması	-	-	X	X	-	100	100

DERS MATERYALİNİN TANITIMI

Öğrenme biriminin kapağını gösterir.

Öğrenme biriminin numarasını ve adını gösterir.

Öğrenme biriminin ana konularını gösterir.

Etkileşimli kitap, video, ses, animasyon, uygulama, oyun, soru vb. ilave kaynaklara ulaşabileceğiniz karekodu gösterir.



Öğrenme biriminde neler öğrenileceğini gösterir.

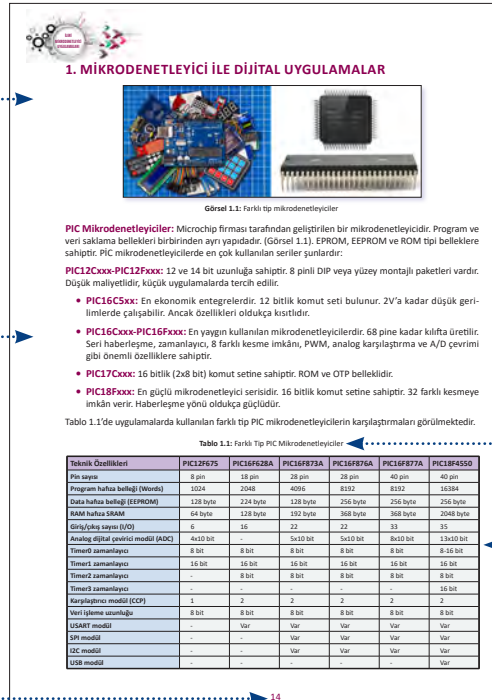
Konuyla ilgili hazırlık çalışmalarını gösterir.

Öğrenme birimiyle ilgili temel kavramları gösterir.

Öğrenme birimi başlığını gösterir.

Konu içeriğini gösterir.

Sayfa numarasını gösterir.



Konu ile ilgili tablo numarası ve adını gösterir.

Konuyla ilgili tabloyu gösterir.

Konuyla ilgili yan başlıkları ve açıklamaları gösterir.

ARDUINO Kartları: İçinde mikrodeneleyici entegresini, bir giriş/çıkış kartını ve Processing/Wiring dilinin uygulamasını barındıran fiziksel programlama platformudur. Kart donanımında ATMEAL AVR tipli mikrodeneleyiciler (Atmega328, Atmega2560 gibi) bulunur (Görsel 1.1). Kart içinde 5 V'luk regüle entegresi ve 16 MHz'lik osilatör vardır, bu nedenle osilatör gibi elemanlara ihtiyaç duymaz. En çok kullanılan Arduino kartları şunlardır:

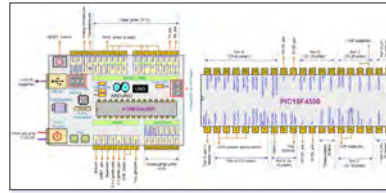
ARDUINO NANO: Atmega328 işlemciye sahiptir. 14'er adet dijital giriş ve çıkış pinine, 8 adet analog giriş pinine sahiptir. Besleme gerilimi USB port üzerinden sağlanır. Ancak Vin pini üzerinden de 7-12V arası DC giriş gerilimi uygulanabilir. 3,3V ve 5V DC voltaj çıkışları vardır. ARDUINO IDE üzerinden rahatlıkla programlanabilir. Tasarımı küçük boyutlu olduğundan breadboard ile beraber kolaylıkla kullanılır.

• **ARDUINO UNO:** Atmega328 işlemciye sahiptir. 14'er adet dijital giriş ve çıkış pinine, 6 adet analog giriş pinine sahiptir. Besleme gerilimi USB port üzerinden sağlanır. Ancak harici DC besleme girişinden veya Vin pini üzerinden de 7-12V arası DC giriş gerilimi uygulanabilir. 3,3V ve 5V DC voltaj çıkışları vardır. ARDUINO IDE üzerinden rahatlıkla programlanabilir. Tasarımı, doğrudan breadboard üzerine yerleştirilip kullanılmaya uygun değildir.

• **ARDUINO MEGA:** Atmega2560 işlemciye sahiptir. 54'er adet dijital giriş ve çıkış pinine, 16 adet analog giriş pinine sahiptir. Besleme gerilimi USB port üzerinden sağlanır. Ancak harici DC besleme girişinden veya Vin pini üzerinden de 7-12V arası DC giriş gerilimi uygulanabilir. 3,3V ve 5V DC voltaj çıkışları vardır. ARDUINO IDE üzerinden rahatlıkla programlanabilir. Tasarımı, doğrudan breadboard üzerine yerleştirilip kullanılmaya uygun değildir. Görsel 1.2'de ARDUINO UNO ile PIC18F4550 mikrodeneleyicilerin pin yapıları, Tablo 1.2'de farklı tip ARDUINO kartlarının karşılaştırmaları görülmektedir.

Tablo 1.2: Farklı Tip ARDUINO Kartları

Özellik/Özellikler	ARDUINO NANO	ARDUINO UNO
Kullanılan mikrodeneleyici	Atmega328	Atmega328
Dijital giriş ve çıkış (GPIO) pin sayısı	14 (5-pin)	14 (5-pin)
Analog giriş pin sayısı	8	6
Flash bellek	16 KB	16 KB
SRAM	2 KB	2 KB
EEPROM	1 KB	1 KB
Baskı hızı	10 MHz	16 MHz
Uzunluk	45 mm	68,6 mm
Geniřlik	18 mm	25,4 mm
Ağırlık	7 g	25 g



Görsel 1.2: ARDUINO UNO ve PIC18F4550 mikrodeneleyicilerin pin yapıları

15

Konu ile ilgili görseli gösterir.

Konuyla ilgili görselin adını ve numarasını gösterir.

Konuyla ilgili uygulamalar için yönerge ikonunu ve yönerge içeriğini gösterir.

UYGULAMA YÖNERGESİ

Bu öğrenme biriminde ARDUINO UNO ile PIC mikrodeneleyiciler kullanılarak çeşitli dijital uygulamalar gerçekleştirilecektir. Bu doğrultuda aşağıda verilen uygulama yönergesini inceleyiniz ve çalışmalarını yönergeye uygun gerçekleştiriniz.

Bu öğrenme birimine altı 16 farklı uygulama (sıralı olarak) bulunmaktadır. Aşağıdaki tabloda üç farklı mikroişlemci modeline göre verilen projeler görülmektedir. Uygulamaları aşağıdaki işlem sırasına göre gerçekleştiriniz.

- İlgili uygulamada kullanılacak mikroişlemci modelini belirleyiniz.
- Uygulamayı yapacak grup sayısını (1 ila 3 kişilik) ve istenirse projelerdeki grup sayısını (2 ila 6 kişilik) belirleyiniz. Her grup üyesinin uygulama sürecindeki görevini belirleyiniz.
- Belirlenen mikroişlemci modeli için verilen uygulama devresinin simülasyonunu elektronik simülasyon yazılımı ile yapınız.
- Belirlenen mikroişlemci modeli için verilen uygulama devresini breadboard ya da baskı devre plaketi üzerine kurunuz. Bu doğrultuda malzeme listesi oluşturunuz ve malzemelerin sağlanmalarını kontrol ediniz.
- Devrenin kodlarını ilgili editörde oluşturup derleyiniz ve mikroişlemciye yükleyiniz.
- Devrenin simülasyon ve programlama dosyalarını uygun isimde kaydediniz ve öğretmenimize teslim ediniz.
- İşlem sonunda devrenin raporunu öğretmeninizin yönergesi doğrultusunda (soruları da cevaplayarak) hazırlayıp öğretmenimize teslim ediniz.
- En fazla üç uygulamayı sıralı olarak, gruplar halinde ve eş güdümlü yürütebilirsiniz.

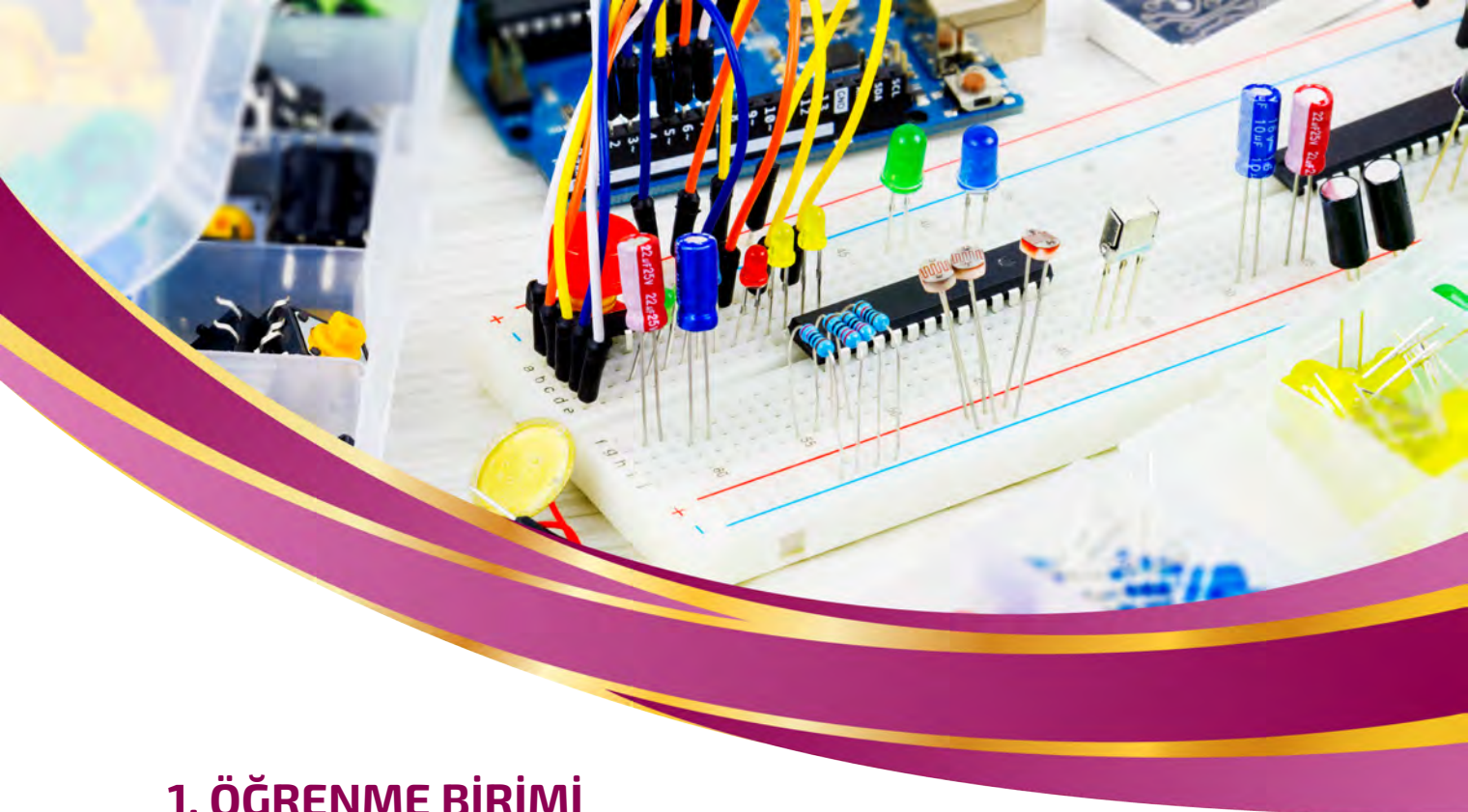
No.	Uygulama Adı	PIC Serisi	ARDUINO Serisi
01.	Yıkık Devresi	PIC16F628A	UNO kartı
02.	LED Animasyon Devresi	PIC16F628A	UNO kartı
03.	Şarj RGB LED Kontrol Devresi	PIC16F628A	UNO kartı
04.	LED'li Ay Yıldı Devresi	PIC16F873A	UNO kartı
05.	Kayan Yazı Devresi	PIC18F4550	UNO kartı
06.	Havada Yazı Devresi	PIC16F628A	UNO kartı
07.	Tufluk Lamba Devresi	PIC18F4550	UNO kartı
08.	LED Uygulamaları	PIC16F873A-PIC16F877A	UNO kartı
09.	Kronometre Devresi	PIC16F873A	UNO kartı
10.	Dijital Saat Devresi	PIC16F876A	UNO kartı
11.	Akıllıca Yanan Lamba Devresi	PIC12F675	UNO kartı
12.	Renk Algılama Devresi	PIC16F873A	UNO kartı
13.	Sesli Kilit Devresi	PIC16F877A	UNO kartı
14.	Dijital Hava Piyano Devresi	PIC16F873A	UNO kartı
15.	Elektronik Piyano Devresi	PIC16F877A	UNO kartı
16.	Elektronik PIC Devresi		NANO kartı

DİKKAT!

- Uygulama ile ilgili tüm dosyalara yanda verilen karekoddan ulaşabilirsiniz.
- Ekteki komut devrelerde kullanılan elemanların ayak bağlantılarına, baskı devre şemalarına ve detaylı teknik bilgilerine ulaşabilirsiniz.

21

Etkileşimli kitap, video, ses, animasyon, uygulama, oyun, soru vb. ilave kaynaklara ulaşabileceğiniz karekodu gösterir.



1. ÖĞRENME BİRİMİ

MİKRODENETLEYİCİ İLE DİJİTAL UYGULAMALAR

KONULAR

- 1.1. Işık Efekt Devresi
- 1.2. LED Animasyon Devresi
- 1.3. Şerit RGB LED Kontrol Devresi
- 1.4. LED'li Ay Yıldız Devresi
- 1.5. Kayan Yazı Devresi
- 1.6. Havada Yazı Devresi
- 1.7. Trafik Lamba Devresi
- 1.8. LCD Uygulamaları
- 1.9. Kronometre Devresi
- 1.10. Dijital Saat Devresi
- 1.11. Alkışla Yanan Lamba Devresi
- 1.12. Renk Algılama Devresi
- 1.13. Şifreli Kilit Devresi
- 1.14. Dijital Hesap Makinesi Devresi
- 1.15. Elektronik Piyano Devresi
- 1.16. Elektronik PLC Devresi

NELER ÖĞRENECEKSİNİZ?

- Farklı mikroişlemcileri kullanarak çeşitli dijital devreleri breadboard üzerinde kurma ve çalıştırma
- Farklı mikroişlemcileri kullanarak çeşitli dijital devreleri baskı devre plaketine yapma ve çalıştırma
- Farklı projeler içinden uygun projeleri seçme ve uygulama

HAZIRLIK ÇALIŞMALARI

1. Dijital entegreler yazılımla kontrol edilebilir mi? Fikirlerinizi sınıfta arkadaşlarınızla paylaşınız.
2. Tek çeşit mikrodnetleyici yerine yüzlerce çeşit mikrodnetleyici üretilmesinin sebebi ne olabilir? Fikirlerinizi sınıfta arkadaşlarınızla paylaşınız.

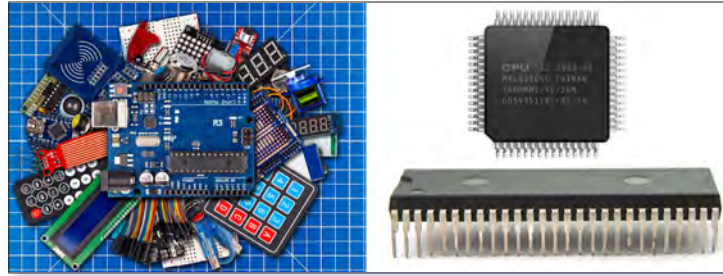
TEMEL KAVRAMLAR

ARDUINO, ARM32 mikroişlemciler, breadboard, el aletleri, elektronik devre, elektronik devre elemanları (direnç, display, sensör vb.), mikroişlemci yazılımları, PIC, plaket, ölçü aleti



1. MİKRODENETLEYİCİ İLE DİJİTAL UYGULAMALAR

Öğrenme birimlerindeki uygulamalarda “PIC” ve “ARDUINO” olmak üzere iki farklı mikrodeneleyici kullanılmıştır.



Görsel 1.1: Farklı tip mikrodeneleyiciler

PIC Mikrodeneleyiciler: Microchip firması tarafından geliştirilen bir mikrodeneleyicidir. Program ve veri saklama bellekleri birbirinden ayrı yapıdadır (Görsel 1.1). EPROM, EEPROM ve ROM tipi belleklere sahiptir. PIC mikrodeneleyicilerde en çok kullanılan seriler şunlardır:

- **PIC12Cxxx-PIC12Fxxx Serisi:** 12 veya 14 bitlik komut seti bulunur. 8 pinli DIP veya yüzey montajlı paketlere sahiptir. Düşük maliyetleri ile küçük uygulamalarda tercih edilirler.
- **PIC16C5xx Serisi:** En ekonomik entegrelerdir. 12 bitlik komut seti bulunur. 2-5 V DC gerilim aralığında çalışabilirler. Program belleği, RAM ve işlemci hızı düşüktür.
- **PIC16Cxxx-PIC16Fxxx:** En yaygın kullanılan mikrodeneleyicilerdir. 68 pine kadar kılıfta üretilir. Seri haberleşme, zamanlayıcı, 8 farklı kesme imkânı, PWM, analog karşılaştırma ve A/D çevrimi gibi önemli özelliklere sahiptir.
- **PIC17Cxxx:** 16 bitlik (2x8 bit) komut setine sahiptir. ROM ve OTP belleklidir.
- **PIC18Fxxx:** En güçlü mikrodeneleyici serisidir. 16 bitlik komut setine sahiptir. 32 farklı kesmeye imkân verir. Haberleşme yönü oldukça güçlüdür.

Tablo 1.1’de uygulamalarda kullanılan farklı tip PIC mikrodeneleyicilerin karşılaştırmaları görülmektedir.

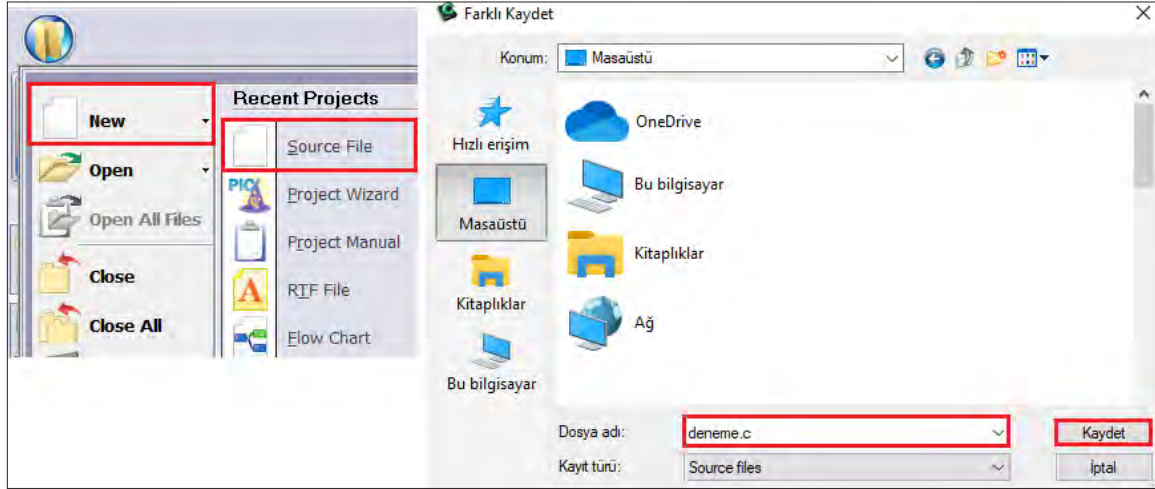
Tablo 1.1: Farklı Tip PIC Mikrodeneleyiciler

Teknik Özellikleri	PIC12F675	PIC16F628A	PIC16F873A	PIC16F876A	PIC16F877A	PIC18F4550
Pin sayısı	8 pin	18 pin	28 pin	28 pin	40 pin	40 pin
Program hafıza belleği (Words)	1024	2048	4096	8192	8192	16384
Data hafıza belleği (EEPROM)	128 byte	224 byte	128 byte	256 byte	256 byte	256 byte
RAM hafıza SRAM	64 byte	128 byte	192 byte	368 byte	368 byte	2048 byte
Giriş/çıkış sayısı (I/O)	6	16	22	22	33	35
Analog dijital çevirici modül (ADC)	4x10 bit	-	5x10 bit	5x10 bit	8x10 bit	13x10 bit
Timer0 zamanlayıcı	8 bit	8 bit	8 bit	8 bit	8 bit	8-16 bit
Timer1 zamanlayıcı	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit
Timer2 zamanlayıcı	-	8 bit	8 bit	8 bit	8 bit	8 bit
Timer3 zamanlayıcı	-	-	-	-	-	16 bit
Karşılaştırmacı modül (CCP)	1	2	2	2	2	2
Veri işleme uzunluğu	8 bit	8 bit	8 bit	8 bit	8 bit	8 bit
USART modül	-	Var	Var	Var	Var	Var
SPI modül	-	-	Var	Var	Var	Var
I2C modül	-	-	Var	Var	Var	Var
USB modül	-	-	-	-	-	Var



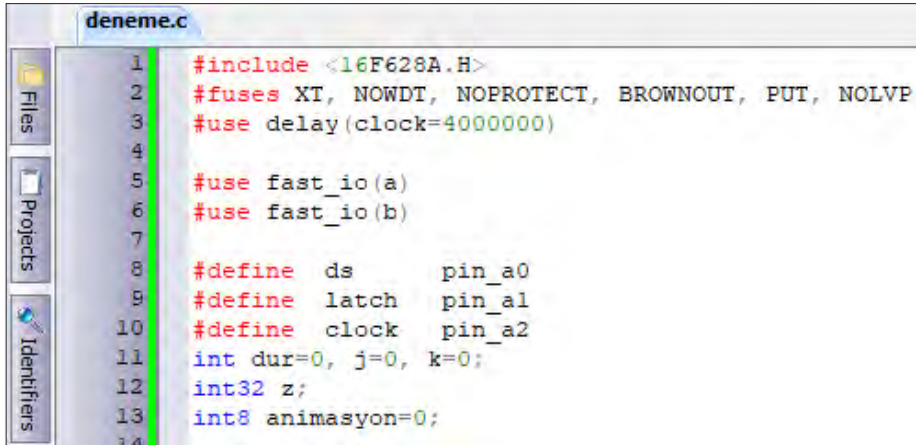
PIC mikrodeneleyici programlama editöründe C kaynak dosyası oluşturma ve yükleme şu adımlar doğrultusunda yapılır:

- Program çalıştırılır. Klasör simgesine tıklanır ve açılan menü sekmesinden **New (→)** **Source File (←)** seçilir (Görsel 1.3).



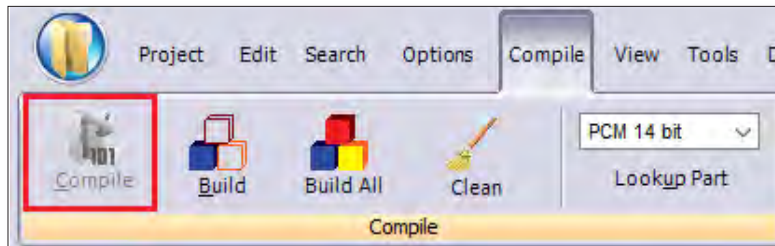
Görsel 1.3: Kaynak dosyası oluşturma ve dosyayı kaydetme

- Açılan **Farklı Kaydet** penceresinden dosya adı “deneme.c” olarak kaydedilir (Görsel 1.3).
- Kaynak dosyaya program kodları yazılır (Görsel 1.4).



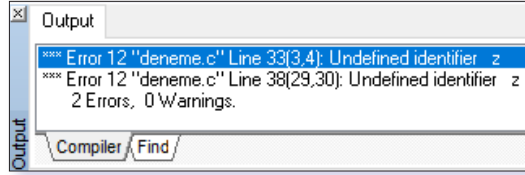
Görsel 1.4: Program kodlarının yazılması

- Program kodları yazıldıktan sonra derleme işlemi yapıp HEX dosyasının oluşturulması için **Compile** menüsünden **Compile** butonuna tıklanır (Görsel 1.5).



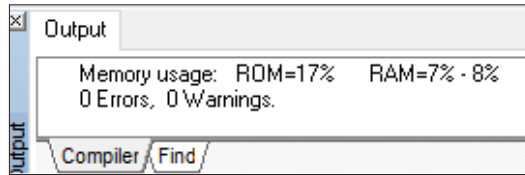
Görsel 1.5: Program kodlarının derlenmesi

- Program kodlarında hatalar varsa **Output** penceresinde hatanın olduğu satırlar ve hata tipleri görülür (Görsel 1.6). Hata uyarılarına çift tıklanarak hata satırlarına gidilir. Program kodları, hata tipine göre düzenlenerek tekrar derlenir.



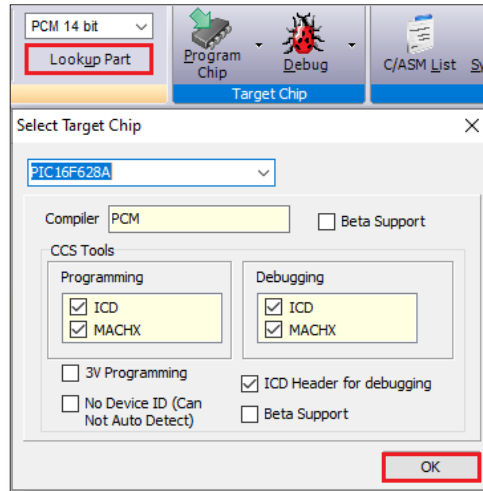
Görsel 1.6: Derleme hataları ve tipleri

- Program kodlarında hata yoksa **Output** penceresinde “0 Errors, 0 Warnings” uyarısı görülür (Görsel 1.7). Kaynak dosyasının bulunduğu klasöre derleme dosyaları oluşturulur (.HEX, .BAK, .COF, .ERR, .ESYM, .LST, .STA, .SYM, .TRE).



Görsel 1.7: Derlemenin tamamlanması

- **Compile** menüsünden **Lookup Part** tıklanır. Açılan pencereden PIC mikrodeneleyici seçilir ve ardından OK butonuna tıklanır (Görsel 1.8).



Görsel 1.8: PIC mikrodeneleyicinin seçilmesi

- PIC programlama araçlarının bilgisayara bağlantısı yapılır (Görsel 1.9).



Görsel 1.9: PIC programlayıcısı



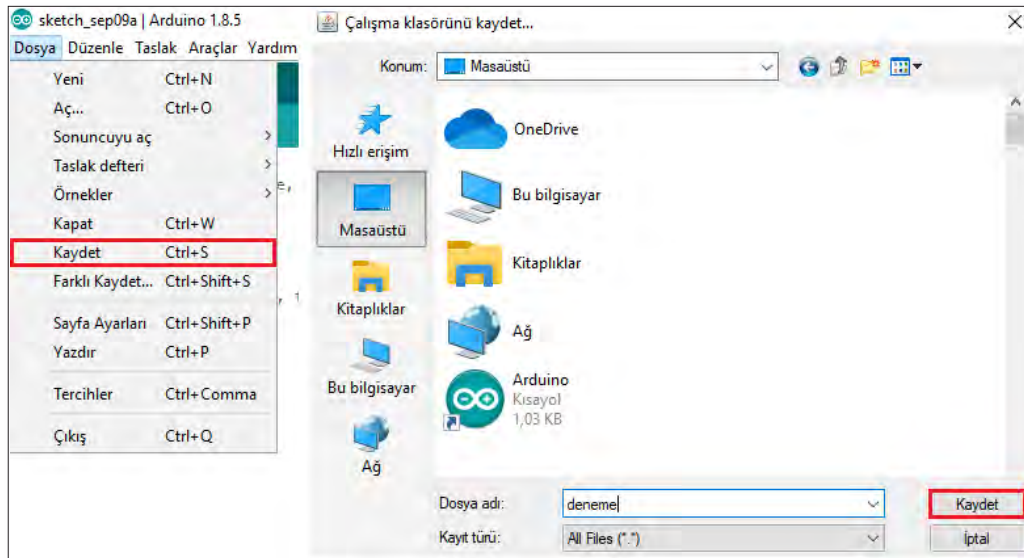
- **Compile** menüsünden **Program Chip** tıklanır. Açılan pencereden PIC programlama aracı (ICD veya MACHX) seçilir. COM port ayarları yapılarak HEX dosyası mikrodnetleyiciye yüklenir (Görsel 1.10).



Görsel 1.10: Programlama aracı seçimi

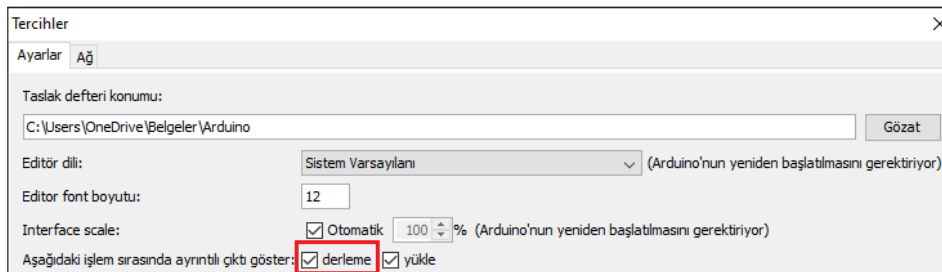
ARDUINO IDE programında kaynak dosyası oluşturma ve yükleme şu adımlar doğrultusunda yapılır:

- ARDUINO IDE programı çalıştırılır. Açılıştaki sketch isimli geçici proje dosyası otomatik olarak program tarafından oluşturulur.
- **Dosya** menüsüne tıklanır. Açılan dosya menüsünden **Kaydet** seçilir (Görsel 1.11). Açılan Çalışma klasörünü kaydet penceresinden **dosya adı (deneme)** verilir. Kaydet butonuna tıklanır.



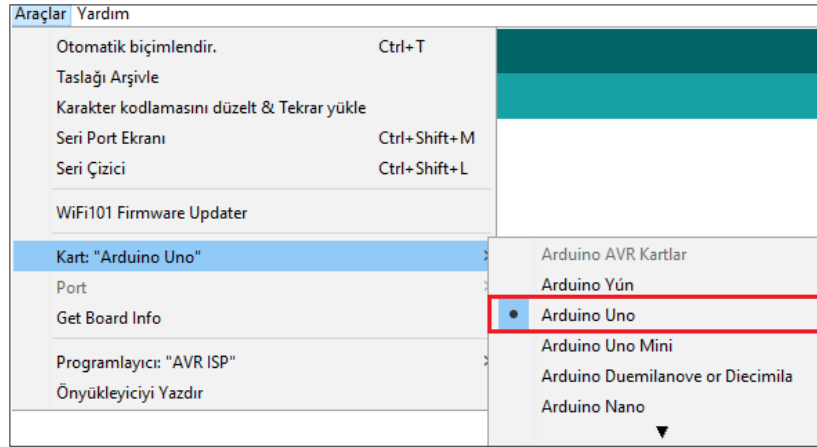
Görsel 1.11: ARDUINO IDE kaynak dosyası oluşturma ve kaynak dosyası kaydetme

- Program kodları derlendikten sonra HEX dosyasını görüntülemek için **Dosya menüsü/Tercihler** komutuna tıklanır. Açılan **Tercihler** penceresi **Ayarlar** sekmesinden derleme seçeneği işaretlenir (Görsel 1.12). Tamam butonuna tıklanarak pencere kapatılır.



Görsel 1.12: HEX dosyası görüntüleme ayarı

- Programlamada kullanılacak ARDUINO kartını seçmek için **Araçlar** menüsü/**Kart** komutuna gelinir. Kart: **“ARDUINO UNO”** menüsünden ARDUINO UNO kartı seçilir.

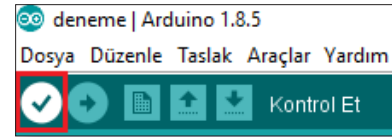


Görsel 1.13: ARDUINO kart seçimi

- Kaynak dosyaya program kodları yazılır (Görsel 1.14).
- Program kodları yazıldıktan sonra derleme işlemi yapıp HEX dosyasının oluşturulması için menü kısa yollarından **Kontrol Et** butonuna tıklanır (Görsel 1.15).

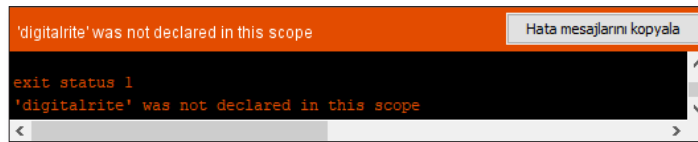


Görsel 1.14: Program kodlarının yazılması



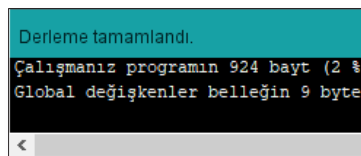
Görsel 1.15: Program kodlarının derlenmesi

- Program kodlarında hatalar varsa bildirim alanında hatanın olduğu satır ve hata şekli görülür (Görsel 1.16). Hatalı kod satırı pembe, hata mesajları turuncu renkle görüntülenir. Program kodları, hata tipine göre düzenlenerek tekrar derlenir.



Görsel 1.16: Derleme hataları ve şekli

- Program kodlarında hata yoksa bildirim alanında **“Derleme tamamlandı.”** uyarısı görülür (Görsel 1.17).



Görsel 1.17: Derlemenin tamamlanması

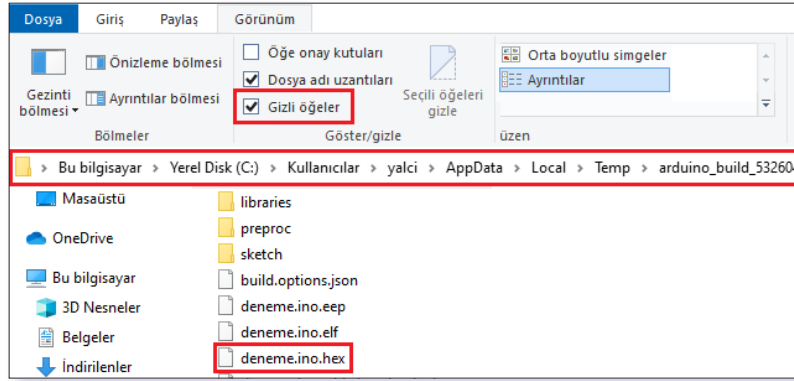


- HEX dosyasının bulunduğu klasör bilgi alanından tespit edilir (Görsel 1.18). Dahası taslak menüsünden “Derlenmiş Binary”i çıkar” kısmına tıklanarak HEX dosyası çalışma klasöründe oluşturulabilir.

```
Derleme tamamlandı.  
-warnings --change-section-1ma .eeprom=0 "C:\Users\yalci\AppData\Local\Temp\arduino_build_532604\deneme.ino.hex"
```

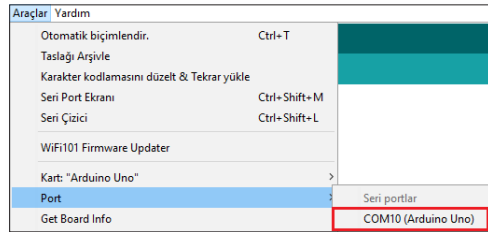
Görsel 1.18: HEX dosyasının bulunduğu klasör

- Dosya penceresi **Görünüm** menüsünden **Gizli öğeler** seçeneği işaretlenir. HEX dosyasının bulunduğu klasöre gidilerek HEX dosyası görüntülenir (Görsel 1.19). Elektronik simülasyon programlarında kullanılacak ARDUINO program dosyasının **deneme.ino.hex** dosyası olduğu unutulmamalıdır.

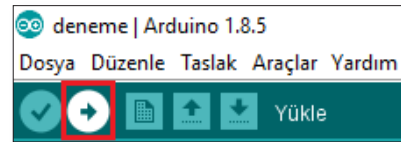


Görsel 1.19: HEX dosyasının klasörde görüntülenmesi

- Program kodları doğrudan ARDUINO karta yüklenecekse ARDUINO kart bilgisayarın USB portuna bağlanır.
- ARDUINO IDE programı **Araçlar** menüsünden **Port** komutuna gelinir. Açılan pencereden ARDUINO kartının takılı olduğu port seçilir (Görsel 1.20).
- Menü kısayollarından **Yükle** butonuna tıklanır ya da klavyeden “CTRL+U” tuşlarına aynı anda basılır (Görsel 1.21).



Görsel 1.20: Port seçimi



Görsel 1.21: Program kodlarının ARDUINO karta yüklenmesi

- ARDUINO karta program doğru yüklendiğinde bilgi alanında yüklemenin sonucu görüntülenir (Görsel 1.22).

```
Yükleme tamamlandı.  
avrdude: 924 bytes of flash verified  
avrdude done. Thank you.
```

Görsel 1.22: Programlama aracı seçimi

Bu öğrenme biriminde ARDUINO UNO ile PIC mikrodnetleyiciler kullanılarak çeşitli dijital uygulamalar gerçekleştirilecektir. Bu doğrultuda aşağıda verilen uygulama yönergesini inceleyiniz ve çalışmalarını yönergeye uygun gerçekleştiriniz.



UYGULAMA YÖNERGESİ



Bu öğrenme birimine ait 16 farklı uygulama bulunmaktadır. Aşağıdaki tabloda üç farklı mikrodnetleyici modeline göre verilen projeler görülmektedir. Uygulamaları aşağıdaki yönerge doğrultusunda gerçekleştiriniz.

- İlgili uygulamada kullanılacak mikrodnetleyici modelini belirleyiniz.
- Uygulamayı gerçekleştirilecek grubu (1 ila 6 kişi) oluşturunuz. Grup üyelerinin görevlerini belirleyiniz.
- Belirlenen mikrodnetleyici modeli için verilen uygulama devresinin simülasyonunu elektronik simülasyon yazılımı ile yapınız.
- Belirlenen mikrodnetleyici modeli için verilen uygulama devresini breadboard ya da baskı devre plaketi üzerine kurunuz. Bu doğrultuda malzeme listesi oluşturunuz ve malzemelerin sağlıklarını kontrol ediniz.
- Devrenin kodlarını ilgili editörde oluşturup derleyiniz ve mikrodnetleyici yükleyiniz.
- Devrenin simülasyon ve programlama dosyalarını uygun isimde kaydediniz ve öğretmeni- nize teslim ediniz.
- İşlem sonunda devrenin raporunu öğretmeninizin yönergesi doğrultusunda (soruları da cevaplayarak) hazırlayıp öğretmeninize teslim ediniz.

No.	Uygulama Adı	PIC Serisi	ARDUINO Serisi
01	Işık Efekt Devresi	PIC16F628A	UNO kartı
02	LED Animasyon Devresi	PIC16F628A	UNO kartı
03	Şerit RGB LED Kontrol Devresi	PIC16F628A	UNO kartı
04	LED'li Ay Yıldız Devresi	PIC16F873A	UNO kartı
05	Kayan Yazı Devresi	PIC18F4550	UNO kartı
06	Havada Yazı Devresi	PIC16F628A	UNO kartı
07	Trafik Lamba Devresi	PIC18F4550	UNO kartı
08	LCD Uygulamaları	PIC16F873A-PIC16F877A	UNO kartı
09	Kronometre Devresi	PIC16F873A	UNO kartı
10	Dijital Saat Devresi	PIC16F876A	UNO kartı
11	Alkışla Yanan Lamba Devresi	PIC12F675	UNO kartı
12	Renk Algılama Devresi	PIC16F873A	UNO kartı
13	Şifreli Kilit Devresi	PIC16F877A	UNO kartı
14	Dijital Hesap Makinesi Devresi	PIC16F873A	UNO kartı
15	Elektronik Piyano Devresi	PIC16F877A	UNO kartı
16	Elektronik PLC Devresi	-	NANO kartı

DİKKAT!

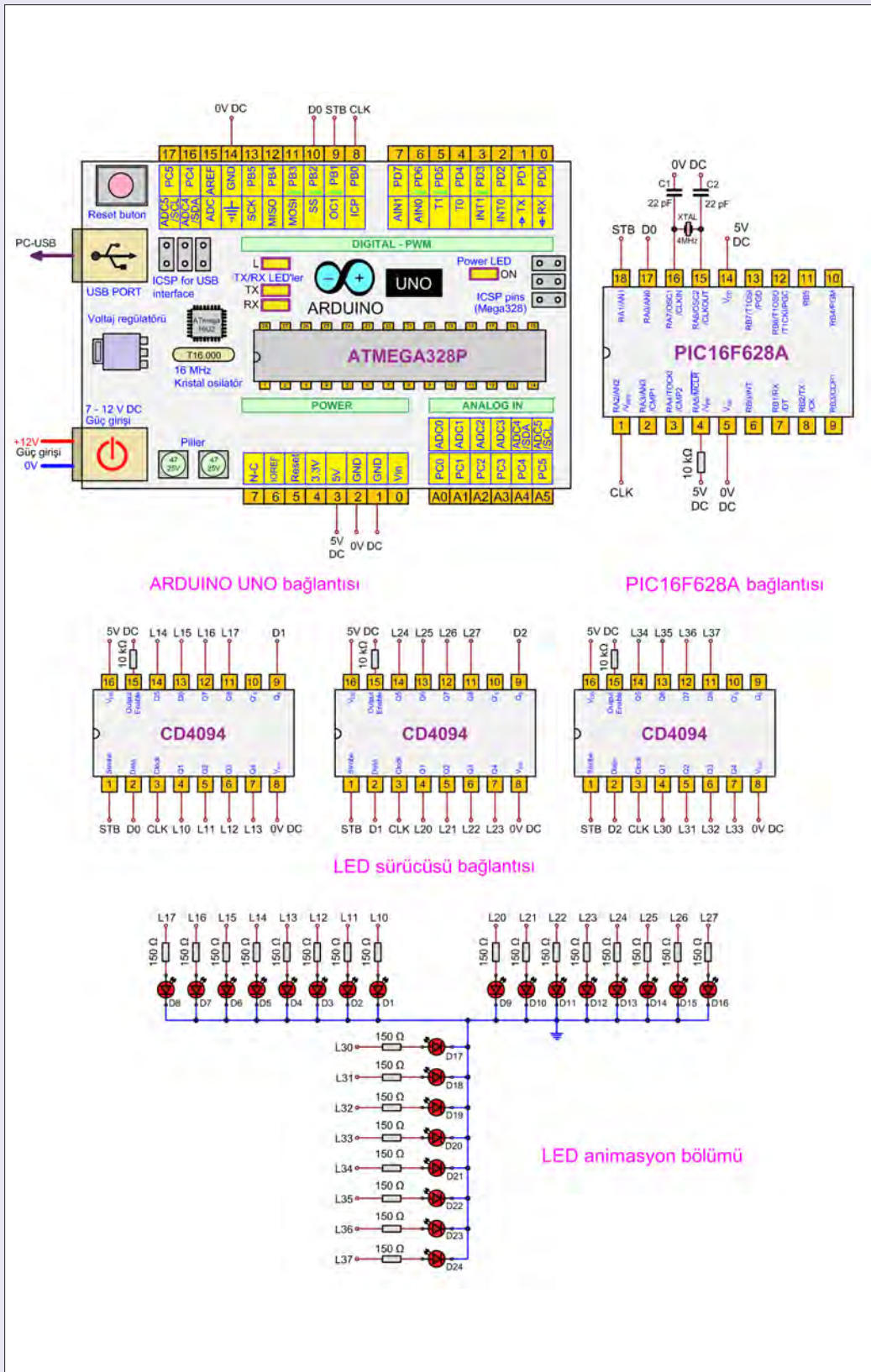
- Uygulama ile ilgili tüm dosyalara yanda verilen karekoddan ulaşabilirsiniz.
- Ekteki kısımdan devrelerde kullanılan elemanların ayak bağlantılarına, baskı devre şemalarına ve detaylı teknik bilgilerine ulaşabilirsiniz.



1.1. Işık Eft Devresi



Devre Şeması: Görsel 1.23'te ışık efekt devresi görölmektedir.



Görsel 1.23: Işık efekt devresi

Devre Hakkında: Işık efekt devresi, 3x8=24 adet LED diyot ve LED diyot sürücüsü olarak CD4094 seri girişli, paralel çıkışlı, kaymalı kaydedici (shift register) entegreleri kullanılarak tasarlanmıştır. Mikrodenetleyici, ışık efekt verilerini seri olarak CD4094 entegresine gönderir ve LED diyotlar üzerinde verileri görüntüler. Devreye enerji uygulandığında LED diyotların tamamı söndürülür. 1. efekt (efekt1) alt programı çalıştırılır. Veriler CD4094 entegresine gönderilir ve LED diyotlarda görüntülenir. Sırasıyla tüm alt programlar çalıştırılarak ışık efekti tamamlanır (efekt1-efekt6). LED diyotların tamamı söndürülür ve 1. efektte geçilir. Program sonsuz döngüde çalışır (Görsel 1.23).

Malzemeler: Tablo 1.3'te iki devrenin malzeme listesi görülmektedir.

Tablo 1.3: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F628A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Uygun ölçülerde	-
CD4094	Seri giriş, paralel çıkış, kaymalı, kaydedici	3 adet
LED diyot	5 mm kırmızı	24 adet
Direnç	150 Ω	24 adet
Direnç	10 k Ω	4 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	+5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları	Uygun çaplarda	-

Kod Yazılımları: Tablo 1.4'te iki devrenin kod yazılımları görülmektedir.

Tablo 1.4: Devrenin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F628A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <SPI.h>	2	#include <16F628A.H>
3	#define STB 9	3	#fuses XT, NOWDT, NOPROTECT, BROWNOUT, PUT, NOLVP
4	#define CLKS 8	4	#use delay(clock=4000000)
5	#define DATAS 10	5	#use fast_io(a)
6	//değişkenler	6	#use fast_io(b)
7	int dur=0, j=0, k=0;	7	#define ds pin_a0
8	byte animasyon=0, sil=0;	8	#define latch pin_a1
9	//alt programlar	9	#define clock pin_a2
10	void ekran_sil()	10	//değişkenler
11	{	11	int dur=0, j=0, k=0;
12	shiftOut(DATAS, CLKS, MSBFIRST, sil);	12	int32 z;



```

13 shiftOut(DATAS, CLKS, MSBFIRST, sil);
14 shiftOut(DATAS, CLKS, MSBFIRST, sil);
15 digitalWrite(STB, HIGH);
16 delayMicroseconds(1);
17 digitalWrite(STB, LOW);}
18 void LED_yaz(byte ent1, byte ent2, byte ent3)
19 {
20 shiftOut(DATAS, CLKS, MSBFIRST, ent3);
21 shiftOut(DATAS, CLKS, MSBFIRST, ent2);
22 shiftOut(DATAS, CLKS, MSBFIRST, ent1);
23 digitalWrite(STB, HIGH);
24 delayMicroseconds(1);
25 digitalWrite(STB, LOW);}
26 void efekt1()
27 {
28 efekt=1;
29 for(k=0;k<8;k++)
30 {
31 LED_yaz(efekt, efekt, efekt);
32 efekt=efekt<<1;
33 delay(250);}
34 efekt=64;
35 for(k=0;k<7;k++)
36 {
37 LED_yaz(efekt, efekt, efekt);
38 efekt=efekt>>1;
39 delay(250);}}
40 void efekt2()
41 {
42 efekt=3;
43 for(k=0;k<4;k++)
44 {
45 LED_yaz(efekt, efekt, efekt);
46 efekt=efekt<<2;
47 delay(250);}
48 efekt=48;
49 for(k=0;k<3;k++)
50 {
51 LED_yaz(efekt, efekt, efekt);
52 efekt=efekt>>2;
53 delay(250);}}
54 void efekt3()
55 {
56 efekt=15;
57 LED_yaz(efekt, efekt, efekt);
58 delay(250);

```

```

13 int8 efekt=0;
14 //alt programlar
15 void ekran_sil()
16 {
17 int j=0;
18 for(j=0;j<24;j++)
19 {
20 output_bit(ds,0);
21 output_high(clock);
22 delay_us(1);
23 output_low(clock);}
24 output_high(latch);
25 delay_us(1);
26 output_low(latch);}
27 void LED_yaz(int8 ent1, int8 ent2, int8 ent3)
28 {
29 z=make32(ent1,ent2,ent3);
30 for(j=0;j<24;j++)
31 {
32 output_bit(ds,bit_test(z,23-j));
33 output_high(clock);
34 delay_us(1);
35 output_low(clock);}
36 output_high(latch);
37 delay_us(1);
38 output_low(latch);}
39 void efekt1()
40 {
41 efekt=1;
42 for(k=0;k<8;k++)
43 {
44 LED_yaz(efekt,efekt,efekt);
45 efekt=efekt<<1;
46 delay_ms(250);}
47 efekt=64;
48 for(k=0;k<7;k++)
49 {
50 LED_yaz(efekt,efekt,efekt);
51 efekt=efekt>>1;
52 delay_ms(250);}}
53 void efekt2()
54 {
55 efekt=3;
56 for(k=0;k<4;k++)
57 {
58 LED_yaz(efekt,efekt,efekt);

```

59	efekt=efekt<<4;	59	efekt=efekt<<2;
60	LED_yaz(efekt, efekt, efekt);	60	delay_ms(250);}
61	delay(250);	61	efekt=48;
62	efekt=15;	62	for(k=0;k<3;k++)
63	LED_yaz(efekt, efekt, efekt);	63	{
64	delay(250);}	64	LED_yaz(efekt,efekt,efekt);
65	void efekt4()	65	efekt=efekt>>2;
66	{	66	delay_ms(250);}}
67	dur=1000;	67	void efekt3()
68	for(k=0;k<5;k++)	68	{
69	{	69	efekt=15;
70	LED_yaz(255,0,0);	70	LED_yaz(efekt,efekt,efekt);
71	delay(dur);	71	delay_ms(250);
72	LED_yaz(0,255,0);	72	efekt=efekt<<4;
73	delay(dur);	73	LED_yaz(efekt,efekt,efekt);
74	LED_yaz(0,0,255);	74	delay_ms(250);
75	delay(dur);	75	efekt=15;
76	dur=dur-200;}}	76	LED_yaz(efekt,efekt,efekt);
77	void efekt5()	77	delay_ms(250);}
78	{	78	void efekt4()
79	dur=500;	79	{
80	for(k=0;k<4;k++)	80	dur=1000;
81	{	81	for(k=0;k<5;k++)
82	LED_yaz(255,255,255);	82	{
83	delay(dur);	83	LED_yaz(255,0,0);
84	LED_yaz(0,0,0);	84	delay_ms(dur);
85	delay(dur);}}	85	LED_yaz(0,255,0);
86	void efekt6()	86	delay_ms(dur);
87	{	87	LED_yaz(0,0,255);
88	dur=250;	88	delay_ms(dur);
89	for(k=0;k<5;k++)	89	dur=dur-200;}}
90	{	90	void efekt5()
91	LED_yaz(0,0,255);	91	{
92	delay(dur);	92	dur=500;
93	LED_yaz(0,255,0);	93	for(k=0;k<4;k++)
94	delay(dur);	94	{
95	LED_yaz(255,0,0);	95	LED_yaz(255,255,255);
96	delay(dur);	96	delay_ms(dur);
97	{	97	LED_yaz(0,0,0);
98	shiftOut(DATAS, CLKS, MSBFIRST, sil);	98	delay_ms(dur);}}
99	shiftOut(DATAS, CLKS, MSBFIRST, sil);	99	void efekt6()
100	shiftOut(DATAS, CLKS, MSBFIRST, sil);	100	{dur=250;
101	digitalWrite(STB, HIGH);	101	for(k=0;k<5;k++)
102	delayMicroseconds(1);	102	{LED_yaz(0,0,255);



103	//ana program	103	delay_ms(dur);
104	void loop() {	104	LED_yaz(0,255,0);
105	ekran_sil();	105	delay_ms(dur);
106	delay(1000);	106	LED_yaz(255,0,0);
107	efekt1();	107	delay_ms(dur);
108	efekt2();	108	dur=dur+200;}}
109	efekt3();	109	void main()
110	efekt4();	110	{setup_timer_1(T1_DISABLED);
111	efekt5();	111	setup_CCP1(CCP_OFF);
112	efekt6();}	112	set_tris_a(0x00);
		113	output_a(0x00);
		114	while(TRUE)
		115	{ekran_sil();
		116	delay_ms(1000);
		117	efekt1();
		118	efekt2();
		119	efekt3();
		120	efekt4();
		121	efekt5();
		122	efekt6();}}

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Animasyon sırasının (efekt1-efekt6) değiştirilmesi programın çalışmasını nasıl etkiler?
2. LED sayısı 32'ye yükseltirirse devre ve programda hangi değişiklikler yapılır?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

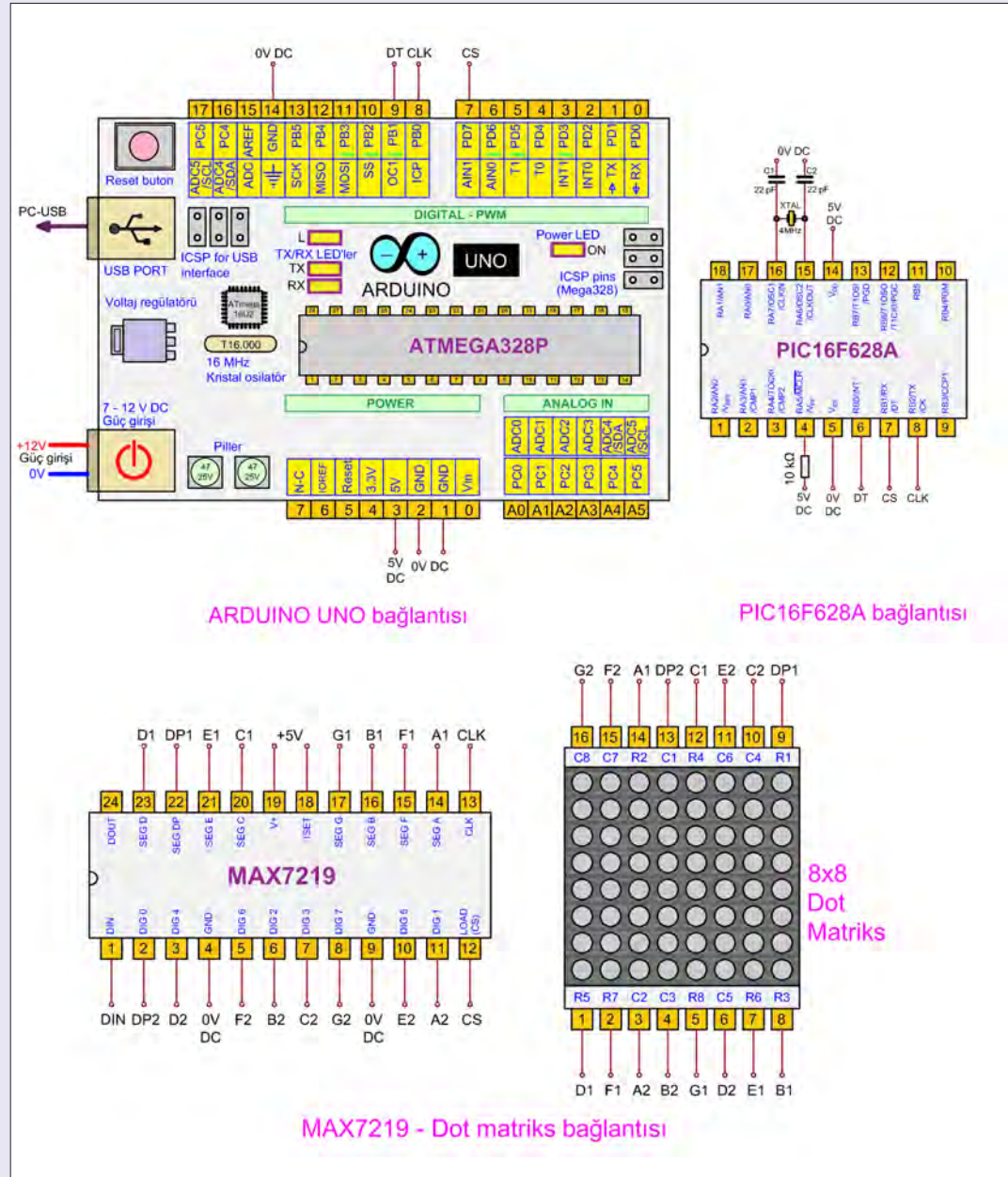
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



1.2. LED Animasyon Devresi



Devre Şeması: Görsel 1.24'te LED animasyon devresi görülmektedir.



Görsel 1.24: LED animasyon devresi

Devre Hakkında: LED animasyon devresi, genellikle LED diyotlarla oluşturulan ekran panellerde görsel şov amaçlı dizayn edilen devrelerdir. Uygulama devresi 8x8 LED dot matriks ve LED sürücüsü olarak MAX7219 entegresi kullanılarak tasarlanmıştır. Mikrodenetleyici, animasyona uygun verileri seri olarak MAX7219 entegresine gönderir ve animasyon 8x8 dot matrikste görüntülenir. Devreye enerji uygulandığında 8x8 dotmatrikste tüm LED diyotlar söndürülür ve ekran parlaklığı **parlaklik(10)** komutuyla (1-15 arası sayılar) ayarlanır. char animasyon[] dizisindeki veriler 0. adresten itibaren 8'li gruplar hâlinde alınarak MAX7219 entegresinin ilgili sütununa gönderilir ve 8x8 dot matrikste görüntülenir. Dizi içindeki 4 animasyon grubu tamamlanınca ekran silinir. 1. animasyon grubuna tekrar geçilir. Program sonsuz döngüde çalışır (Görsel 1.24).



Malzemeler: Tablo 1.5'te iki devrenin malzeme listesi görülmektedir.

Tablo 1.5: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyici kart	1 adet
PIC16F628A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Uygun ölçülerde	-
MAX7219	LED-display sürücü	1 adet
Direnç	10 kΩ	1 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları	Uygun çaplarda	-

Kod Yazılımları: Tablo 1.6'da iki devrenin kod yazılımları görülmektedir.

Tablo 1.6: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F628A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <SPI.h>	2	#include <16F628A.H>
3	#include <LEDControl.h>	3	#fuses XT, NOWDT, NOPROTECT, BROWNOUT, PUT, NOLVP
4	//değişkenler	4	#use delay(clock=4000000)
5	int i, dur;	5	#define DATA PIN_B0
6	LEDControl yt = LEDControl(9, 8, 7, 1);	6	#define LOAD PIN_B1
7	char animasyon[]={0x80, 0xC0, 0xE0, 0xF0, 0xF8, 0xFC, 0xFE, 0xFF, 0x01, 0x03, 0x07, 0x0F, 0x1F, 0x3F, 0x7F, 0xFF, 0xFF, 0x7E, 0x3C, 0x18, 0x18, 0x3C, 0x7E, 0xFF, 0x3C, 0x3C, 0x3C, 0x3C, 0xFF, 0x7E, 0x3C, 0x18}; ///ORTAK KATOT	7	#define CLK PIN_B2
8	//alt programlar	8	#include<MAX7219.c>
9	void flash(int sure)	9	#use fast_io(b)
10	{	10	#use fast_io(a)
11	for (i=0; i<sure; i++)	11	//değişkenler
12	{	12	char animasyon[]={ 0x80, 0xC0, 0xE0, 0xF0, 0xF8, 0xFC, 0xFE, 0xFF, 0x01, 0x03, 0x07, 0x0F, 0x1F, 0x3F, 0x7F, 0xFF, 0xFF, 0x7E, 0x3C, 0x18, 0x18, 0x3C, 0x7E, 0xFF, 0x3C, 0x3C, 0x3C, 0x3C, 0xFF, 0x7E, 0x3C, 0x18}; ///KATOT
13	yt.shutdown(0, true);	13	int8 i,j,k, fls, dur;
14	delay(250);	14	char data1[8];
15	yt.shutdown(0, false);	15	//alt programlar
16	delay(250);}}	16	void parlaklik(int8 parlak)
17	void parlaklik(int parlak)	17	{

```

18 {
19   yt.setIntensity(0, parlak);
20   //genel ayarlar
21   void setup() {
22     yt.shutdown(0, false);
23     yt.setIntensity(0, 15);
24     yt.clearDisplay(0);
25     delay(250);
26     //ana program
27     void loop() {
28       parlaklik(4);
29       dur=100;
30       for (i=0; i<8; i++)
31       {
32         yt.setRow(0,i,animasyon[i]);
33         delay(dur);
34         flash(2);
35         yt.clearDisplay(0);
36         for (i=0; i<8; i++)
37         {
38           yt.setRow(0,i,animasyon[i+8]);
39           delay(dur);
40           flash(2);
41           yt.clearDisplay(0);
42           for (i=0; i<4; i++)
43           {
44             yt.setRow(0,i,animasyon[i+16]);
45             yt.setRow(0,7-i,animasyon[i+16]);
46             delay(dur);
47             flash(2);
48             yt.clearDisplay(0);
49             for (i=0; i<8; i++)
50             {
51               yt.setRow(0,i,animasyon[i+24]);
52               delay(dur);
53               flash(2);
54               yt.clearDisplay(0);

```

```

18   display(0x0A,parlak);
19   void Lflash(int8 sure, int8 adr)
20   {
21     for (i=0; i<sure; i++)
22     {
23       for (k=0; k<8; k++)
24       {
25         data1[k]=0;
26         ekranda_goster(data1,0);
27         delay_ms(250);
28         for (j=0; j<8; j++)
29         {
30           data1[j]=animasyon[(adr*8)+j];
31           ekranda_goster(data1,0);
32           delay_ms(250);}}
33       void ekran_sil()
34       {
35         for (i=0; i<8; i++)
36         {
37           data1[i]=0;
38           ekranda_goster(data1,0);
39           //genel ayarlar
40           void main()
41           {
42             setup_timer_1(T1_DISABLED);
43             setup_CCP1(CCP_OFF);
44             ekran_ayar(0);
45             //ana program
46             while(1)
47             {
48               ekran_sil();
49               parlaklik(10);
50               dur=100;
51               for (i=0; i<8; i++)
52               {
53                 data1[i]=animasyon[i];
54                 ekranda_goster(data1,0);
55                 delay_ms(dur);
56                 fls=0;
57                 Lflash(2,0);
58                 ekran_sil();
59                 for (i=0; i<8; i++)
60                 {
61                   data1[i]=animasyon[i+8];
62                   ekranda_goster(data1,0);
63                   delay_ms(dur);
64                   fls=1;

```



	<pre> 65 Lflash(2,1); 66 ekran_sil(); 67 for (i=0; i<4; i++) 68 { 69 data1[i]=animasyon[i+16]; 70 data1[7-i]=animasyon[i+16]; 71 ekranda_goster(data1,0); 72 delay_ms(dur); 73 fls=2;}} 74 Lflash(2,2); 75 ekran_sil(); 76 for (i=0; i<8; i++) 77 { 78 data1[i]=animasyon[i+24]; 79 ekranda_goster(data1,0); 80 delay_ms(dur); 81 fls=3;}} 82 Lflash(2,3);}}</pre>
--	--

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonununu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Animasyon sayısının artırılması için programda yapılacak değişiklikler nelerdir?
2. Devreye ikinci dot matriks eklemek için devre ve programda yapılacak değişiklikler nelerdir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



Malzemeler: Tablo 1.7’de iki devrenin malzeme listesi görülmektedir.

Tablo 1.7: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F628A	Mikrodenetleyici entegre	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
CD4069	NOT (DEĞİL) kapısı	3 adet
RGB LED diyot	12’li şerit	1 adet
Direnç	10 kΩ	1 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları	Uygun çaplarda	-

Kod Yazılımları: Tablo 1.8’de iki devrenin kod yazılımları görülmektedir.

Tablo 1.8: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F628A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#define kirmizi_LED 11	2	#include <16F628A.h>
3	#define yesil_LED 10	3	#FUSES NOWDT, INTRC_IO, NOPUT, NOPROTECT, NOBROWNOUT, NOMCLR, NOLVP, NOCPD
4	#define mavi_LED 9	4	#use delay(clock=4000000)
5	//değişkenler	5	#include <string.h>
6	char kirmizi=0, yesil=0, mavi=0;	6	#include <math.h>
7	//genel ayarlar	7	#use fast_io(a)
8	void setup()	8	#use fast_io(b)
9	{	9	#define kirmizi_LED portb_0
10	pinMode(kirmizi_LED, OUTPUT);	10	#define yesil_LED portb_1
11	pinMode(yesil_LED, OUTPUT);	11	#define green_LED portb_2
12	pinMode(mavi_LED, OUTPUT);}	12	unsigned int16 count = 0;
13	//ana program	13	unsigned int16 say=0,say_pwm=0, pwm_kontrol=0;
14	void loop()	14	int8 kirmizi=50, yesil=60, mavi=100;
15	{	15	#int_timer0 // Timer0 kesmesi
16	rgb_LED_yak(255, 0, 0);	16	void timer0_kesme ()
17	delay(500);	17	{disable_interrupts(INT_timer0);
18	rgb_LED_yak(0, 255, 0);	18	disable_interrupts(INT_timer2);
19	delay(500);	19	set_timer0(55); // TMR0 değeri belirleniyor.
20	rgb_LED_yak(0, 0, 255);		

```

21 delay(500);
22 rgb_LED_yak(255, 255, 0);
23 delay(500);
24 rgb_LED_yak(0, 255, 255);
25 delay(500);
26 rgb_LED_yak(255, 0, 255);
27 delay(500);
28 rgb_LED_yak(255, 255, 255);
29 delay(500);
30 while(1)
31 {
32   kirmizi=kirmizi+5;
33   yesil=yesil+10;
34   mavi=mavi+15;
35   rgb_LED_yak(kirmizi, yesil, mavi);
36   delay(100);} }

```

```

20 say_pwm++;
21 if (say_pwm>=20)
22 {pwm_kontrol++;
23   if ((pwm_kontrol*5)>kirmizi){output_LOW(pin_B0);}
24   if ((pwm_kontrol*5)>yesil){output_LOW(pin_B1);}
25   if ((pwm_kontrol*5)>mavi){output_LOW(pin_B2);}
26   if (pwm_kontrol>=51){pwm_kontrol=0;}
27   say_pwm=0;}
28 enable_interrupts(INT_timer2);
29 enable_interrupts(INT_timer0);}
30 #int_timer2 // Timer2 kesmesi
31 void timer2_kesme ()
32 {disable_interrupts(INT_timer0);
33   disable_interrupts(INT_timer2);
34   set_timer2(0);
35   say++;
36   enable_interrupts(INT_timer2);
37   enable_interrupts(INT_timer0);}
38 void rgb_LED_yak(int r, int g, int b, int sure)
39 {if (r==1)
40 {output_high(pin_B0);}
41 else
42 {output_low(pin_B0);}
43 if (g==1)
44 {output_high(pin_B1);}
45 else
46 {output_low(pin_B1);}
47 if (b==1)
48 {output_high(pin_B2);}
49 else
50 {output_low(pin_B2);}
51 delay_ms(sure);
52 output_b(0x00);}
53 void main()
54 {setup_timer_0(RTCC_INTERNAL | RTCC_DIV_1);
55   setup_timer_1(T1_DISABLED);
56   setup_timer_2(T2_DIV_BY_16,255,5);
57   setup_comparator(NC_NC_NC_NC);
58   setup_vref(FALSE);
59   set_tris_b(0x00);
60   output_b(0x00);
61   rgb_LED_yak(1, 0, 0, 500);
62   rgb_LED_yak(0, 1, 0, 500);
63   rgb_LED_yak(0, 0, 1, 500);
64   rgb_LED_yak(1, 1, 0, 500);
65   rgb_LED_yak(0, 1, 1, 500);
66   rgb_LED_yak(1, 0, 1, 500);

```



		67	rgb_LED_yak(1, 1, 1, 500);
		68	set_timer0(55);
		69	set_timer2(0);
		70	enable_interrupts(GLOBAL);
		71	enable_interrupts(INT_timer2);
		72	enable_interrupts(INT_timer0);
		73	output_b(0xff);
		74	while(1)
		75	{if (say>20)
		76	{kirmizi=kirmizi+5;
		77	yesil=yesil+10;
		78	mavi=mavi+15;
		79	if (kirmizi>0)
		80	{output_high(pin_B0);}
		81	if (yesil>0)
		82	{output_high(pin_B1);}
		83	if (mavi>0)
		84	{output_high(pin_B2);}
		85	say=0;}}

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonununu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Renk skalasını daha da artırmak için program üzerinde yapılacak değişiklikler nelerdir?
2. RGB LED sayısı 24'e yükseltirirse devre ve programda hangi değişiklikler yapılır?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

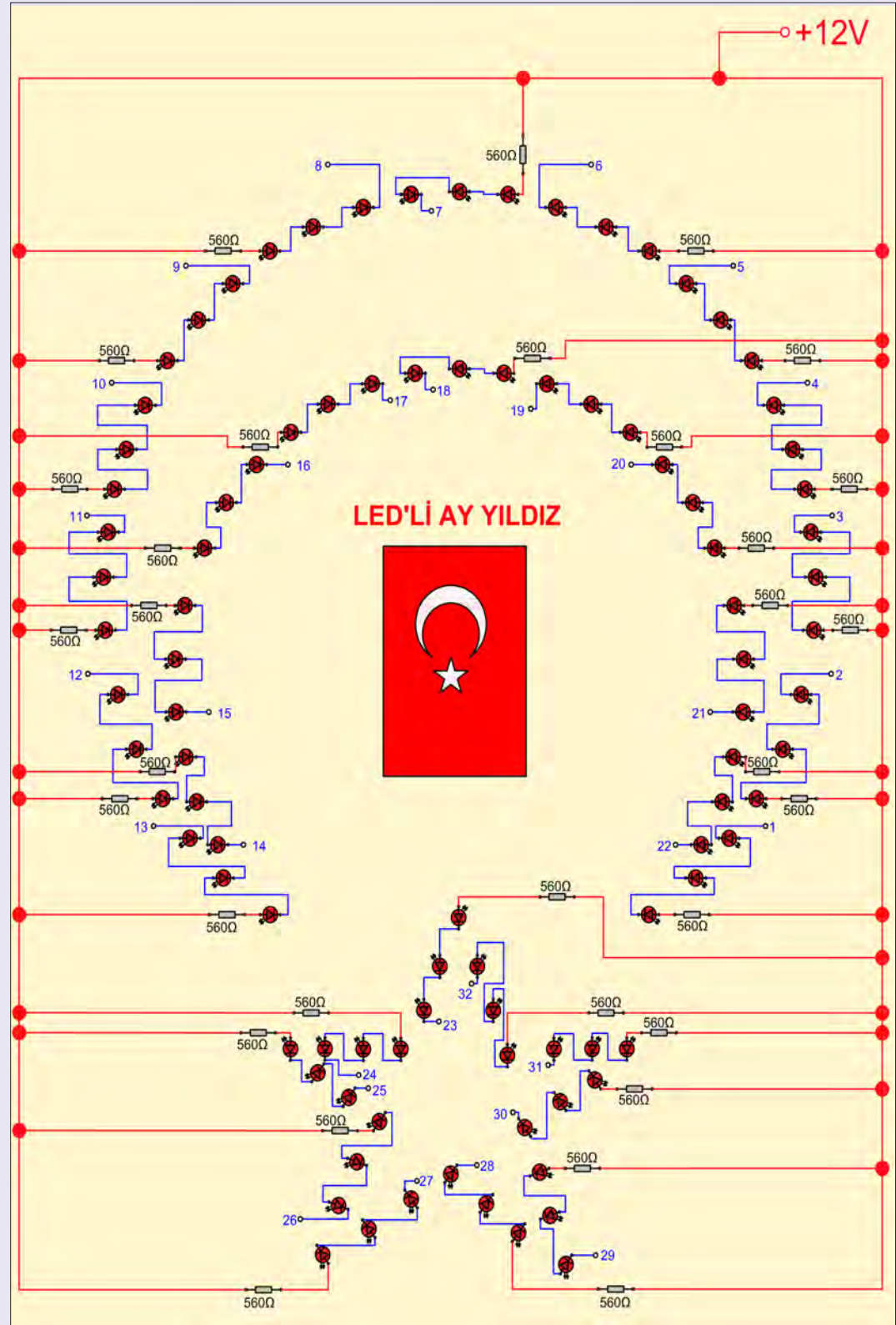
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



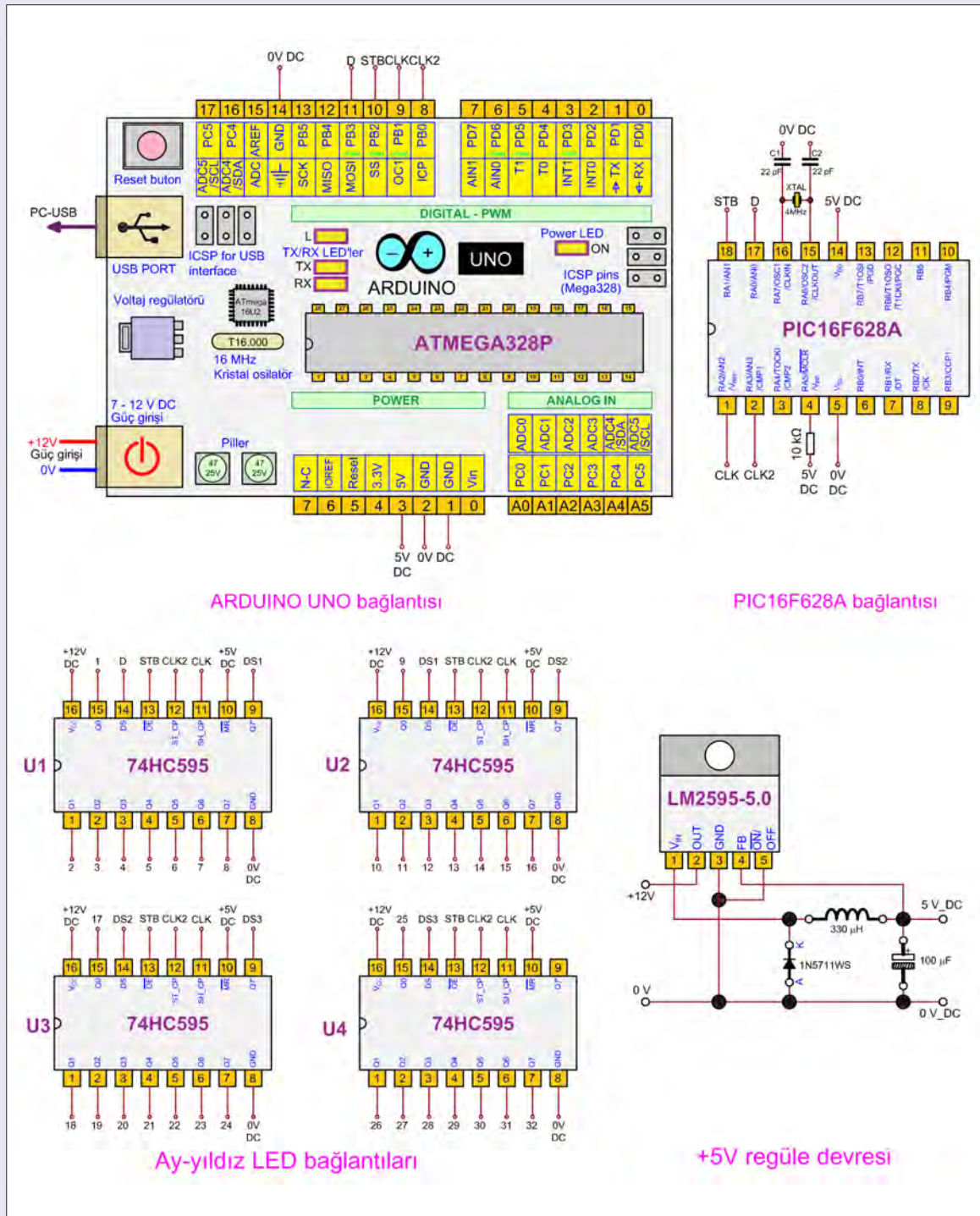
1.4. LED'li Ay Yıldız Devresi



Devre Şeması: Görsel 1.26'da LED'li ay yıldız devresinin bağlantısı, Görsel 1.27'de ise LED'li ay yıldız devresi için kontrol devre şeması görülmektedir.



Görsel 1.26: LED'li ay yıldız devresi



Görsel 1.27: LED'li ay yıldız kontrol devresi

Devre Hakkında: LED'li ay yıldız devresi 96 adet LED diyot ve LED diyot sürücülü olarak 74HC595 kaymalı kaydedici entegreleri kullanılarak tasarlanmıştır (Görsel 1.26). Mikrodenetleyici, LED verilerini seri olarak 74HC595 entegrelerine gönderir ve veriler LED diyotlar üzerinde görüntülenir. Devreye enerji uygulandığında LED diyotlardaki tüm veriler silinir. Bayrak animasyonu **bayrak_uclugit()** alt programından itibaren çalıştırılır. Veriler 74HC595 entegresine gönderilir ve LED diyotlarda görüntülenir. Sırasıyla tüm alt programlar çalıştırılarak bayrak animasyonu tamamlanır. Bayrak animasyonu sonsuz döngüde çalışır (Görsel 1.27).

Malzemeler: Tablo 1.9’da iki devrenin malzeme listesi görülmektedir.

Tablo 1.9: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F628A	Mikrodenetleyici entegre	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaketa	Yeterli ölçülerde	-
74HC595	Seri giriş paralel çıkışlı, kaymalı kaydedici	4 adet
LM2595-5	Voltaj regülatörü	1 adet
Diyot	1N5711WS	1 adet
Bobin	330 µH	1 adet
Kondansatör	100 µF elektrolitik	1 adet
LED diyot	Kırmızı veya beyaz	96 adet
Direnç	150 Ω (LED ön direnci olarak kullanılabilir.)	32 adet
Direnç	10 kΩ	1 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları	Uygun çaplarda	-

Kod Yazılımları: Tablo 1.10’da iki devrenin kod yazılımları görülmektedir.

Tablo 1.10: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F628A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <SPI.h>	2	#include <16F628A.H>
3	#define ds 11	3	#fuses XT, NOWDT, NOPROTECT,
4	#define latch 10		BROWNOUT, PUT, NOLVP
5	#define clk1 9	4	#use delay(clock=4000000)
6	#define clk2 8	5	#use fast_io(a)
7	//değişkenler	6	#use fast_io(b)
8	int j=0, k=0;	7	#define ds pin_a0
9	long int z;	8	#define latch pin_a1
10	//alt programlar	9	#define clock pin_a2
11	void bayrak_temizle()	10	#define clk2 pin_a3
12	{int j=0;	11	//değişkenler
13	digitalWrite(latch,HIGH);	12	int j=0, k=0;
14	for(j=0;j<32;j++)	13	int32 z;
15	{	14	//alt programlar



```

16 digitalWrite(ds,HIGH);
17 digitalWrite(clk1,HIGH);
18 delayMicroseconds(1);
19 digitalWrite(clk1,LOW);}
20 digitalWrite(latch,LOW);
21 delayMicroseconds(1);}
22 void bayrak_yaz(long int ent_data)
23 {
24 digitalWrite(latch,HIGH);
25 for(j=0;j<32;j++)
26 {
27 digitalWrite(ds,!bitRead(ent_data,31-j));
28 digitalWrite(clk1,HIGH);
29 digitalWrite(clk2,LOW);
30 delayMicroseconds(1);
31 digitalWrite(clk1,LOW);
32 digitalWrite(clk2,HIGH);;}
33 digitalWrite(latch,LOW);}
34 void bayrak_uclugit()
35 {z=1;
36 for(k=0;k<32;k++)
37 {
38 bayrak_yaz(z);
39 delay(100);
40 z=z<<1;}
41 z=0x40000000;
42 for(k=0;k<31;k++)
43 {
44 bayrak_yaz(z);
45 delay(100);
46 z=z>>1;}}
47 void bayrak_flas()
48 {
49 for(k=0;k<5;k++)
50 {
51 z=0xffffffff;
52 bayrak_yaz(z);
53 delay(500);
54 z=0x00000000;
55 bayrak_yaz(z);
56 delay(500);}}
57 void bayrak_dalgali()
58 {
59 for(k=0;k<5;k++)
60 {
61 z=0xAAAAAAAA;
62 bayrak_yaz(z);

```

```

15 void bayrak_temizle()
16 {int j=0;
17 output_high(latch);
18 for(j=0;j<32;j++)
19 {output_bit(ds,1);
20 output_high(clock);
21 delay_us(1);
22 output_low(clock);}
23 output_low(latch);
24 delay_us(1);}
25 void bayrak_yaz(int32 ent_data)
26 {output_high(latch);
27 for(j=0;j<32;j++)
28 {
29 output_bit(ds,!bit_test(ent_data,31-j));
30 output_high(clock);
31 output_low(clk2);
32 delay_us(1);
33 output_low(clock);
34 output_high(clk2);}
35 output_low(latch);}
36 void bayrak_uclugit()
37 {z=1;
38 for(k=0;k<32;k++)
39 {
40 bayrak_yaz(z);
41 delay_ms(100);
42 z=z<<1;}
43 z=0x40000000;
44 for(k=0;k<31;k++)
45 {
46 bayrak_yaz(z);
47 delay_ms(100);
48 z=z>>1;}}
49 void bayrak_flas()
50 {for(k=0;k<5;k++)
51 {
52 z=0xffffffff;
53 bayrak_yaz(z);
54 delay_ms(500);
55 z=0x00000000;
56 bayrak_yaz(z);
57 delay_ms(500);}}
58 void bayrak_dalgali()
59 {
60 for(k=0;k<5;k++)
61 {

```

63	delay(500);	62	z=0xAAAAAAAA;
64	z=0x55555555;	63	bayrak_yaz(z);
65	bayrak_yaz(z);	64	delay_ms(500);
66	delay(500);}}	65	z=0x55555555;
67	void ay_yildiz()	66	bayrak_yaz(z);
68	{	67	delay_ms(500);}}
69	for(k=0;k<5;k++)	68	void ay_yildiz()
70	{	69	{
71	z=0x003fffff;	70	for(k=0;k<5;k++)
72	bayrak_yaz(z);	71	{
73	delay(500);	72	z=0x003fffff;
74	z=0xffc00000;	73	bayrak_yaz(z);
75	bayrak_yaz(z);	74	delay_ms(500);
76	delay(500);}}	75	z=0xffc00000;
77	void bayrak_ruzgar()	76	bayrak_yaz(z);
78	{	77	delay_ms(500);}}
79	for(k=0;k<5;k++)	78	void bayrak_ruzgar()
80	{	79	{
81	z=0xffffffff;	80	for(k=0;k<5;k++)
82	bayrak_yaz(z);	81	{z=0xffffffff;
83	delay(500);}}	82	bayrak_yaz(z);
84	void bayrak_sabit()	83	delay_ms(500);}}
85	{z=0xffffffff;	84	void bayrak_sabit()
86	bayrak_yaz(z);	85	{ z=0xffffffff;
87	for(k=0;k<20;k++)	86	bayrak_yaz(z);
88	{	87	for(k=0;k<20;k++)
89	delay(250);}}	88	{delay_ms(250);}}
90	//genel ayarlar	89	//genel ayarlar
91	void setup()	90	void main()
92	{	91	{
93	pinMode(ds, OUTPUT);	92	setup_timer_1(T1_DISABLED);
94	pinMode(latch, OUTPUT);	93	setup_CCP1(CCP_OFF);
95	pinMode(clk1, OUTPUT);	94	set_tris_a(0x00);
96	pinMode(clk2, OUTPUT);	95	output_a(0x02);
97	bayrak_temizle();}	96	delay_ms(500);
98	//ana program	97	bayrak_temizle();
99	void loop()	98	//ana program
100	{	99	while(TRUE)
101	bayrak_uclugit();	100	{bayrak_uclugit();
102	bayrak_flas();	101	bayrak_flas();
103	bayrak_dalgali();	102	bayrak_dalgali();
104	ay_yildiz();	103	ay_yildiz();
105	bayrak_ruzgar();	104	bayrak_ruzgar();
106	bayrak_sabit();}	105	bayrak_sabit();}}



İşlem Basamakları

1. Uygun tipli mikrodeneleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Ay yıldız devresi tasarımını oluşturunuz ve LED'leri bu alana uygun şekilde yerleştiriniz.
6. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodeneleyiciye yükleyiniz. Devreyi çalıştırınız.
7. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

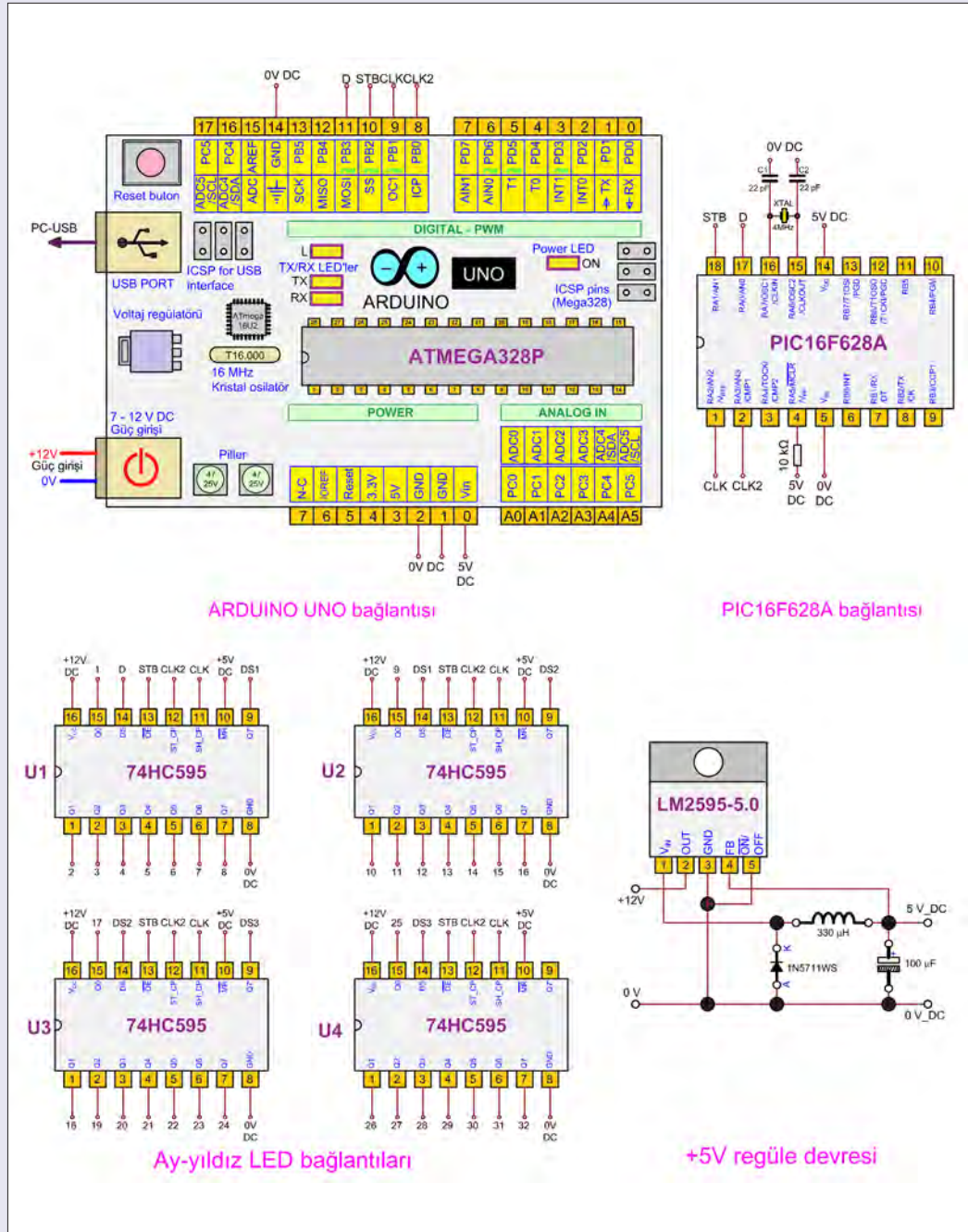
1. Bayrak animasyon sırasının yerleri değiştirildiğinde programın çalışmasında nasıl bir değişiklik olur?
2. Bayrak animasyonunda 32 bitlik veri kullanılmasının sebebi nedir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

1.5. Kayan Yazı Devresi

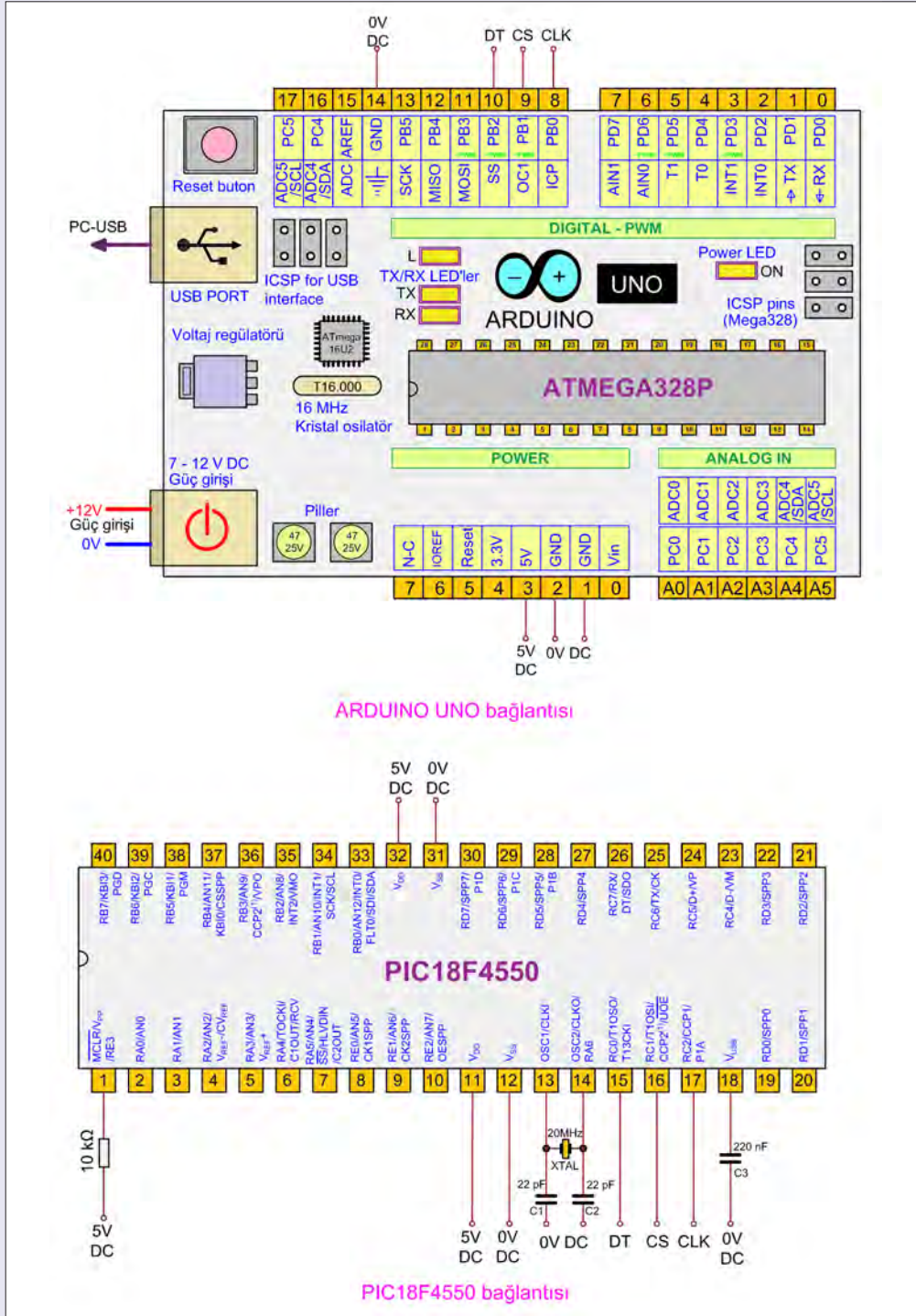
Devre Şeması: Görsel 1.28'de dot matrix ile tasarlanan kayan yazı devre şeması, Görsel 1.29'da kayan yazı kontrol devresi görülmektedir.



Görsel 1.28: Dot matrixli kayan yazı devresi

Devre Hakkında: Kayan yazı devresi 4 adet 8x8 dot matris ve 4 adet MAX7219 LED diyot sürücü entegresi kullanılarak tasarlanmıştır (Görsel 1.28). Mikrodnetleyici; harf, sembol ve sayılardan oluşan karakterlerin verilerini seri olarak MAX7219 entegrelerine gönderir ve veriler 8x8 dot matris üzerinde görüntülenir. Devreye enerji uygulandığında dot matrislerdeki tüm veriler silinir. **char *yazi** dizisindeki ilk karakter alınıp 5x7 font formatına dönüştürülür. 4. matris 7. sütuna 5x7 fontun birinci veri bloğu yazılır. Diğer 31 sütuna boşluk (0) verisi gönderilir ve "sure" ile tanımlanan değişken değeri kadar ekranda gösterilir.

Veriler bir sütun sola kaydırılır. 4. matris 7. sütuna 5x7 fontun ikinci veri bloğu yazılır. Diğer 30 sütuna boşluk verisi gönderilir “sure” ile tanımlanan değişken değeri kadar ekranda gösterilir. Veriler bir sütun sola kaydırılır. Bu işlem ilk karakterin tüm verileri tamamlanıncaya kadar devam ettirilir. Bir sütunluk boşluk bırakılır. Sırasıyla **char *yazi** dizisindeki diğer karakterler için de aynı işlemler uygulanır. Dizi içindeki tüm karakterler işlendikten sonra 32 adet boşluk işlemi uygulanarak yazının ekrandan kayıp gitmesi sağlanır. Tüm işlemler tamamlanınca reset fonksiyonuyla **char *yazi** dizisindeki ilk karakterden itibaren yazının tekrar ekranda kaydırılması sağlanır.



Görsel 1.29: Kayan yazı kontrol devresi

Malzemeler: Tablo 1.11’de iki devrenin malzeme listesi görülmektedir.

Tablo 1.11: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC18F4550	Mikrodenetleyici entegre	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
MAX7219	LED sürücü	4 adet
8x8 dot matris	Kırmızı	4 adet
Direnç	10 kΩ	5 adet
Kristal osilatör	20 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Kondansatör	220 nF, seramik tip, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları	Uygun çaplarda	-

Kod Yazılımları: Tablo 1.12’de iki devrenin kod yazılımları görülmektedir.

Tablo 1.12: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC18F4550)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <SPI.h>	2	#include <18F4550.h>
3	#include <yazikontrol.h>	3	#use delay(clock=20000000)
4	#include <ard_font.h>	4	#include <string.h>
5	yazikontrol yt = yazikontrol(10, 8, 9, 4); //dataPin,	5	#include <font.h>
6	clkPin, csPin, entegre sayısı	6	#include <math.h>
7	//değişkenler	7	#include <STDLIB.H>
8	const int chip_sayisi=4;	8	#include "MX7219.c"
9	int i=0, karakter_genislik=0, m=0;	9	#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG,
10	int k=0, b=0, sure=50;	10	USBDIV, PLL2, CPUDIV1, VREGEN, NOBROWNOUT
11	unsigned int len=0;	11	#use fast_io(b)
12	unsigned long karakter_adres;	12	#use fast_io(a)
13	uint8_t g_k=0, tr1=0, tr2=0;	13	#use fast_io(c)
14	uint16_t dn=0;	14	//değişkenler
15	char *yazi="İLERİ MIKRODENETLEYİCİ UYGULAMALARI";	15	const int8 chip_sayisi=4;
16	//alt programlar	16	int8 i=0, karakter_genislik=0, m=0;
17	void(* resetFunc) (void) = 0;	17	int8 k=0, b=0, sure=50;
18	void MAX7219_ayarla(int chip)	18	unsigned int len=0;
	{while(chip>0)		unsigned int16 g_k=0, karakter_adres;



19	{	19	byte t_data[]={0x00,0x00,0x00,0x00,0x00};
20	yt.kapat(chip-1, false);	20	char *yazi="İLERİ MİKRODENETLEYİCİ UYGULAMALARI";
21	yt.tarama(chip-1, 8);	21	//alt programlar
22	yt.parlaklik(chip-1, 15);	22	void max7219_ayarla(int chip)
23	yt.temizle(chip-1);	23	{while(chip>0)
24	chip--;}}	24	{output_high(CS);
25	void ekran_yaz (int sure)	25	MAX7219_yaz(durdurma_reg, calistirma_komutu,chip);
26	{int i=0;	26	MAX7219_yaz(kod_cozucu_reg, kod_cozme_kapali,chip);
27	for (i=0; i<8; i++)	27	MAX7219_yaz(tarama_reg, digit_0_7,chip);
28	{	28	MAX7219_yaz(parlaklik_reg, 0x0F,chip);
29	yt.satiryaz(0,8-i, ekran[i]);}	29	MAX7219_yaz(display_test_reg, test_komutu,chip);
30	for (i=0; i<8; i++)	30	delay_ms(100);
31	{	31	MAX7219_yaz(display_test_reg, testyok_komutu,chip);
32	yt.satiryaz(1,8-i, ekran[8+i]);}	32	chip--;}}
33	for (i=0; i<8; i++)	33	void ekran_kaydir()
34	{	34	{for (i=0; i<(chip_sayisi*8)-1; i++)
35	yt.satiryaz(2,8-i, ekran[16+i]);}	35	{ekran[(chip_sayisi*8)-1-i]=ekran[(chip_sayisi*8)-2-i];}
36	for (i=0; i<8; i++)	36	void ekran_sil()
37	{	37	{for (i=0; i<(chip_sayisi*8); i++)
38	yt.satiryaz(3,8-i, ekran[24+i]);}	38	{ekran[i]=0;}}
39	delay(sure);}	39	void bosluk_yaz()
40	void ekran_kaydir()	40	{for (b=0; b<(chip_sayisi*8); b++)
41	{for (i=0; i<(chip_sayisi*8)-1; i++)	41	{ekran_kaydir();
42	{	42	ekran[0]=0;
43	ekran[(chip_sayisi*8)-1-i]=ekran[(chip_sayisi*8)-2-i];}}	43	ekran_yaz(sure);}}
44	void ekran_sil()	44	void bir_bosluk()
45	{	45	{ekran_kaydir();
46	for (i=0; i<(chip_sayisi*8); i++)	46	ekran[0]=0;
47	{ekran[i]=0;}}	47	ekran_yaz(sure);}
48	void bosluk_yaz()	48	void turkce_kar(char c) {
49	{	49	switch (c)
50	for (b=0; b<(chip_sayisi*8); b++)	50	{
51	{	51	case 'Ç' : karakter_adres=5;
52	ekran_kaydir();	52	karakter_genislik=5; break;
53	ekran[0]=0;	53	case 'ç' : karakter_adres=10;
54	ekran_yaz(sure);}	54	karakter_genislik=5; break;
55	void turkce_kar(uint16_t c) {	55	case 'Ş' : karakter_adres=15;
56	switch (c)	56	karakter_genislik=5;; break;
57	{	57	case 'ş' : karakter_adres=20;
58	case 0xC387 : karakter_adres=5;	58	karakter_genislik=5; break;
59	karakter_genislik=5; break; //'Ç'	59	case 'İ' : karakter_adres=25;
60	case 0xC3A7 : karakter_adres=10;	60	karakter_genislik=3; break;
61	karakter_genislik=5; break; //'ç'	61	case 'I' : karakter_adres=30;
62	case 0xC59E : karakter_adres=15;	62	karakter_genislik=3;; break;
63	karakter_genislik=5;; break; //'Ş'	63	case 'Ö' : karakter_adres=35;
64	case 0xC59F : karakter_adres=20;	64	karakter_genislik=5;; break;
65	karakter_genislik=5; break; //'ş'	65	case 'ö' : karakter_adres=40;

66	case 0xC4B0 : karakter_adres=25;	66	karakter_genislik=5;; break;
67	karakter_genislik=3; break; //'i'	67	case 'Ğ' : karakter_adres=45;
68	case 0xC4B1 : karakter_adres=30;	68	karakter_genislik=5; break;
69	karakter_genislik=3;; break; //'i'	69	case 'ğ' : karakter_adres=50;
70	case 0xC396 : karakter_adres=35;	70	karakter_genislik=5; break;
71	karakter_genislik=5;; break; //'Ö'	71	case 'Ü' : karakter_adres=55;
72	case 0xC3B6 : karakter_adres=40;	72	karakter_genislik=5;; break;
73	karakter_genislik=5;; break; //'ö'	73	case 'ü' : karakter_adres=60;
74	case 0xC49E : karakter_adres=45;	74	karakter_genislik=5; break;
75	karakter_genislik=5; break; //'Ğ'	75	default :karakter_adres=0;
76	case 0xC49F : karakter_adres=50;	76	karakter_genislik=4; break; }}
77	karakter_genislik=5; break; //'ğ'	77	//genel ayarlar
78	case 0xC3BC : karakter_adres=55;	78	void setup()
79	karakter_genislik=5;; break; //'Ü'	79	{setup_timer_1(T1_DISABLED);
80	case 0xC39C : karakter_adres=60;	80	setup_timer_2(T2_DISABLED,0,1);
81	karakter_genislik=5; break; //'ü'	81	setup_comparator(NC_NC_NC_NC);
82	default :karakter_adres=0;	82	setup_vref(FALSE);
83	karakter_genislik=4; break; //'boşluk'}}	83	set_tris_c(0x00);
84	//genel ayarlar	84	output_c(0xff);}
85	void setup() {	85	//ana program
86	MAX7219_ayarla(chip_sayisi);	86	void main()
87	//ana program	87	{
88	void loop() {	88	MAX7219_ayarla(chip_sayisi);
89	len=strlen(yazi);	89	while(1)
90	for (k=0; k<len; k++)	90	{len=strlen(yazi);
91	{g_k=*yazi;	91	for (k=0; k<len; k++)
92	if(g_k>=32 && g_k<=126)	92	{g_k=*yazi;
93	{g_k=g_k-32;	93	if(g_k>=32 && g_k<=126)
94	if(g_k>=0 && g_k<=45)	94	{g_k=g_k-32;
95	{karakter_genislik=font_width[g_k];	95	if(g_k>=0 && g_k<=45)
96	karakter_adres=g_k*5;	96	{karakter_genislik=font_width[g_k];
97	for (m=0; m<karakter_genislik; m++)	97	karakter_adres=g_k*5;
98	{	98	for (m=0; m<karakter_genislik; m++)
99	ekran_kaydir();	99	{
100	ekran[0]=font[karakter_adres];	100	ekran_kaydir();
101	ekran_yaz(sure);	101	ekran[0]=font[karakter_adres];
102	karakter_adres++;}	102	ekran_yaz(sure);
103	bir_bosluk();}	103	karakter_adres++;}
104	else	104	bir_bosluk();}
105	{g_k=g_k-46;	105	else
106	karakter_genislik=font_width1[g_k];	106	{g_k=g_k-46;
107	karakter_adres=g_k*5;	107	karakter_genislik=font_width1[g_k];
108	for (m=0; m<karakter_genislik; m++)	108	karakter_adres=g_k*5;
109	{ekran_kaydir();	109	for (m=0; m<karakter_genislik; m++)
110	ekran[0]=font1[karakter_adres];	110	{
111	ekran_yaz(sure);	111	ekran_kaydir();



112	karakter_adres++;}	112	ekran[0]=font1[karakter_adres];
113	bir_bosluk();}}	113	ekran_yaz(sure);
114	else	114	karakter_adres++;}
115	{tr1=g_k;	115	bir_bosluk();}}
116	yazi++;	116	else
117	tr2=*yazi;	117	{
118	dn = ((uint16_t)tr1 << 8) tr2;	118	turkce_kar(g_k);
119	turkce_kar(dn);	119	for (m=0; m<karakter_genislik; m++)
120	for (m=0; m<karakter_genislik; m++)	120	{
121	{ekran_kaydir();	121	ekran_kaydir();
122	ekran[0]=font2[karakter_adres];	122	ekran[0]=font2[karakter_adres];
123	ekran_yaz(sure);	123	ekran_yaz(sure);
124	karakter_adres++;}	124	karakter_adres++;}
125	bir_bosluk();}	125	bir_bosluk();}
126	yazi++;}	126	yazi++;}
127	bosluk_yaz();	127	bosluk_yaz();
128	resetFunc();}	128	reset_cpu();}}

İşlem Basamakları

1. Uygun tipli mikrodenetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Mikrodenetleyici kontrol devresini breadboard veya plaket üzerinde kurunuz.
5. Dot matrixli kayan yazılı devreyi breadboard veya plaket üzerinde kurunuz.
6. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodenetleyiciye yükleyiniz. Devreyi çalıştırınız.
7. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

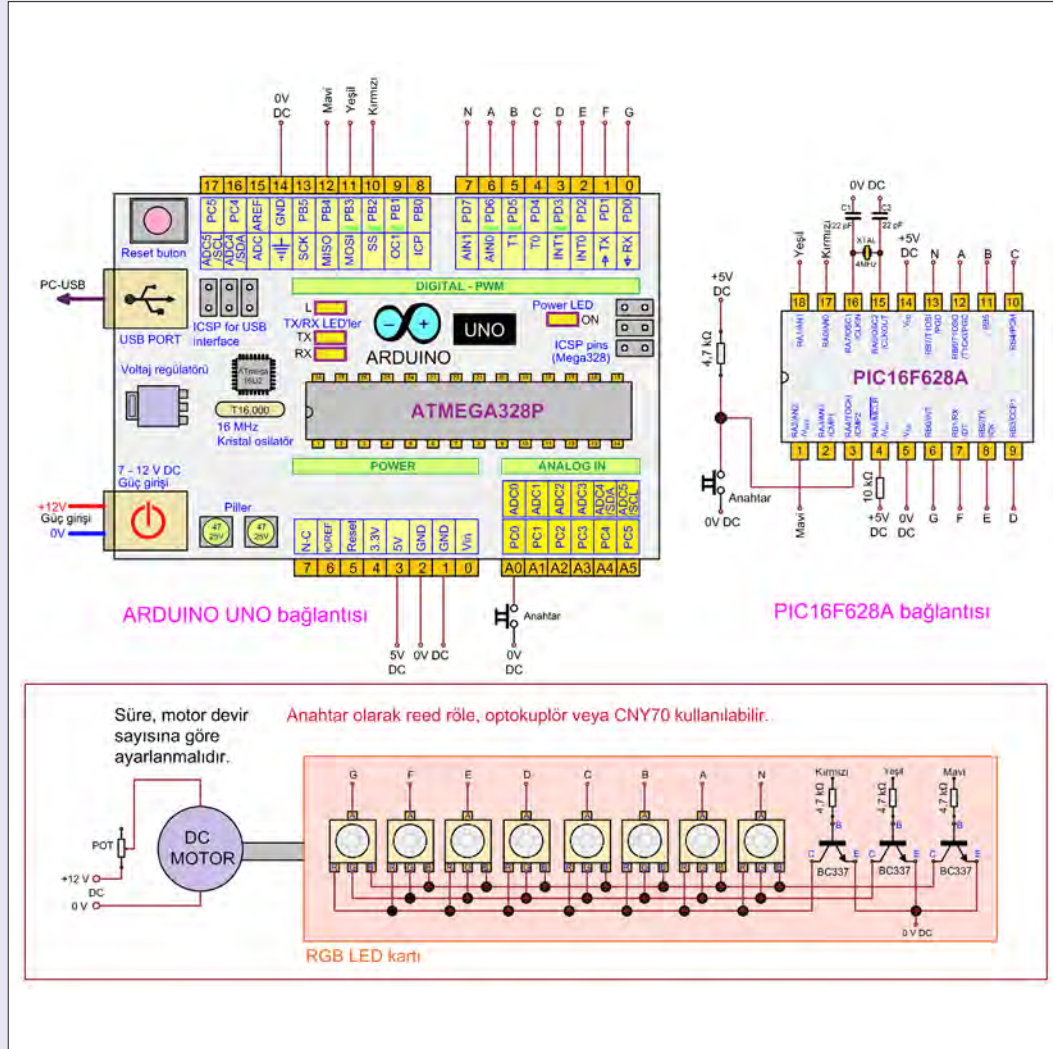
1. Yazının soldan sağa kayması için program üzerinde yapılacak değişiklikler nelerdir?
2. Yazının ekranda sabit kalması için program üzerinde yapılacak değişiklikler nelerdir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

1.6. Havada Yazı Devresi

Devre Şeması: Görsel 1.30'da havada yazı devresi görülmektedir.



Görsel 1.30: Havada yazı kontrol devresi

Devre Hakkında: Havada yazı devresi; RGB LED diyot, BC337 transistör ve 12V DC motor kullanılarak tasarlanmıştır. Mikrodenetleyici, yazı fontlarını paralel veri formatında RGB LED diyotlara gönderir ve seçili renge göre ilgili transistörü iletime geçirir. Böylece değişik renklerde yazı fontları elde edilir. Bu işlemler sırasında DC motor çalıştığından havada değişik renklerde yazı görüntüsü elde edilir. Devreye enerji uygulandığında RGB LED diyotlardaki tüm veriler silinir. DC motor dönmeye başlar. DC motor dönerken anahtar kapalı konuma geçince **char yazi[8]** dizisindeki ilk karakter alınıp 5x7 font formatına dönüştürülür. 5x7 fontun birinci veri bloğu RGB LED diyotlara paralel olarak gönderilir. “renk” değişkeninin değerine göre transistör veya transistörler iletime geçirilir. “sure” ile tanımlanan değişken değeri kadar RGB LED diyotlar iletimde kalır. Bu işlem ilk karakterin tüm verileri tamamlanıncaya kadar devam ettirilir. Bir sütunluk boşluk bırakılır. Sıra ile **char yazi[8]** dizisindeki diğer karakterler için de aynı işlemler uygulanır. Böylece havada değişik renklerde yazı elde edilmiş olur. Tüm işlemler tamamlanınca reset fonksiyonuyla program başlangıca döner. Anahtar kapalı konuma geçinceye kadar beklenir. Uygulamada motor devir sayısı ve “sure” değişkeni ayarlanarak yazının havada en uygun şekilde oluşturulması sağlanır (Görsel 1.30).



Malzemeler: Tablo 1.13’de iki devrenin malzeme listesi görülmektedir.

Tablo 1.13: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC18F4550	Mikrodenetleyici entegre	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
MAX7219	LED sürücü	4 adet
8x8 dot matris	Kırmızı	4 adet
Direnç	10 kΩ	5 adet
Kristal osilatör	20 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Kondansatör	220 nF, seramik tip, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.14’te iki devrenin kod yazılımları görülmektedir.

Tablo 1.14: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F628A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <SPI.h>	2	#include <16F628A.h>
3	#include <ard_font.h>	3	#FUSES NOWDT, INTRC_IO, NOPUT, NOPROTECT,
4	#define kırmızı 10		NOBROWNOUT, NOLVP, NOCPD
5	#define yeşil 11	4	#use delay(clock=4000000)
6	#define mavi 12	5	#include <string.h>
7	#define anahtar A0	6	#include <math.h>
8	//değişkenler	7	#include <hfont.h>
9	int karakter_genislik=0, m=0, renk=0;	8	#use fast_io(a)
10	int k=0, z=0;	9	#use fast_io(b)
11	unsigned int len=0;	10	#define anahtar pin_a4
12	unsigned long g_k=0, karakter_adres, sure=50;	11	//değişkenler
13	uint8_t tr1=0, tr2=0;	12	int8 karakter_genislik=0, m=0, renk=0;
14	uint16_t dn=0;	13	int8 k=0;
15	char yazı[8]="MIKRO";	14	unsigned int len=0;
16	//alt programlar	15	unsigned int16 g_k=0, karakter_adres, sure=100;
17	void RGB()	16	char yazı[8]="MIKRO";
18	{renk++;	17	//alt programlar
19	digitalWrite(kırmızı, bitRead(renk, 0));	18	void RGB()

```

20 digitalWrite(yesil,bitRead(renc,1));
21 digitalWrite(mavi,bitRead(renc,2));
22 delay(sure);}
23 void bir_bosluk()
24 {for (z=0; z<8; z++)
25 {digitalWrite(z,LOW);
26 delayMicroseconds(3);}
27 digitalWrite(kirmizi,LOW);
28 digitalWrite(yesil,LOW);
29 digitalWrite(mavi,LOW);
30 delay(sure);
31 delay(sure);
32 delay(sure);}
33 void turkce_kar( uint16_t c) {
34 switch (c)
35 {
36 case 0xC387 : karakter_adres=5;
37 karakter_genislik=5; break; //"Ç"
38 case 0xC3A7 : karakter_adres=10;
39 karakter_genislik=5; break; //"ç"
40 case 0xC59E : karakter_adres=15;
41 karakter_genislik=5;; break; //"Ş"
42 case 0xC59F : karakter_adres=20;
43 karakter_genislik=5; break; //"ş"
44 case 0xC4B0 : karakter_adres=25;
45 karakter_genislik=3; break; //"İ"
46 case 0xC4B1 : karakter_adres=30;
47 karakter_genislik=3;; break; //"i"
48 case 0xC396 : karakter_adres=35;
49 karakter_genislik=5;; break; //"Ö"
50 case 0xC3B6 : karakter_adres=40;
51 karakter_genislik=5;; break; //"ö"
52 case 0xC49E : karakter_adres=45;
53 karakter_genislik=5; break; //"Ğ"
54 case 0xC49F : karakter_adres=50;
55 karakter_genislik=5; break; //"ğ"
56 case 0xC3BC : karakter_adres=55;
57 karakter_genislik=5;; break; //"Ü"
58 case 0xC39C : karakter_adres=60;
59 karakter_genislik=5; break; //"ü"
60 default :karakter_adres=0;
61 karakter_genislik=4; break; //"boşluk"}}
62 //genel ayarlar
63 void setup() {
64 pinMode(kirmizi, OUTPUT);
65 pinMode(yesil, OUTPUT);
66 pinMode(mavi, OUTPUT);

```

```

19 {renc++;
20 output_bit(pin_a0,bit_test(renc,0));
21 output_bit(pin_a1,bit_test(renc,1));
22 output_bit(pin_a2,bit_test(renc,2));
23 delay_ms(sure);}
24 void bir_bosluk()
25 {output_b(0x00);
26 output_bit(pin_a0,0);
27 output_bit(pin_a1,0);
28 output_bit(pin_a2,0);
29 delay_ms(sure);
30 delay_ms(sure);
31 delay_ms(sure);}
32 void turkce_kar(char c) {
33 switch (c)
34 {
35 case 'Ç' : karakter_adres=5;
36 karakter_genislik=5; break;
37 case 'ç' : karakter_adres=10;
38 karakter_genislik=5; break;
39 case 'Ş' : karakter_adres=15;
40 karakter_genislik=5;; break;
41 case 'ş' : karakter_adres=20;
42 karakter_genislik=5; break;
43 case 'İ' : karakter_adres=25;
44 karakter_genislik=3; break;
45 case 'i' : karakter_adres=30;
46 karakter_genislik=3;; break;
47 case 'Ö' : karakter_adres=35;
48 karakter_genislik=5;; break;
49 case 'ö' : karakter_adres=40;
50 karakter_genislik=5;; break;
51 case 'Ğ' : karakter_adres=45;
52 karakter_genislik=5; break;
53 case 'ğ' : karakter_adres=50;
54 karakter_genislik=5; break;
55 case 'Ü' : karakter_adres=55;
56 karakter_genislik=5;; break;
57 case 'ü' : karakter_adres=60;
58 karakter_genislik=5; break;
59 default :karakter_adres=0;
60 karakter_genislik=4; break;}}
61 //genel ayarlar
62 void main()
63 {setup_timer_1(T1_DISABLED);
64 setup_timer_2(T2_DISABLED,0,1);
65 setup_comparator(NC_NC_NC_NC);

```



```

67 pinMode(anahtar, INPUT_PULLUP);
68 pinMode(0, OUTPUT);
69 pinMode(1, OUTPUT);
70 pinMode(2, OUTPUT);
71 pinMode(3, OUTPUT);
72 pinMode(4, OUTPUT);
73 pinMode(5, OUTPUT);
74 pinMode(6, OUTPUT);
75 pinMode(7, OUTPUT);}
76 //ana program
77 void loop() {
78 if(!digitalRead(anahtar))
79 {for (k=0; k<8; k++)
80 {g_k=yazi[k];
81 if(g_k>=32 && g_k<=126)
82 {g_k=g_k-32;
83 if(g_k>=0 && g_k<=45)
84 {
85 karakter_genislik=font_width[g_k];
86 karakter_adres=g_k*5;
87 for (m=0; m<karakter_genislik; m++)
88 {for (z=0; z<8; z++)
89 {digitalWrite(z,bitRead(font[karakter_adres],z));
90 delayMicroseconds(3);}
91 RGB();
92 delay(sure);
93 karakter_adres++;}
94 bir_bosluk();}
95 else
96 {g_k=g_k-46;
97 karakter_genislik=font_width1[g_k];
98 karakter_adres=g_k*5;
99 for (m=0; m<karakter_genislik; m++)
100 {for (z=0; z<8; z++)
101 {digitalWrite(z,bitRead(font1[karakter_adres],z));
102 delayMicroseconds(3);}
103 RGB();
104 delay(sure);
105 karakter_adres++;}
106 bir_bosluk();}
107 else
108 {tr1=g_k;
109 k++;
110 tr2=yazi[k];
111 dn = ((uint16_t)tr1 << 8) | tr2;
112 turkce_kar(dn);
113 for (m=0; m<karakter_genislik; m++)

```

```

66 setup_vref(FALSE);
67 set_tris_b(0x00);
68 output_b(0x00);
69 set_tris_a(0x10);
70 output_a(0x00);
71 //ana program
72 while(1)
73 {if(!input(anahtar))
74 {for (k=0; k<8; k++)
75 {g_k=yazi[k];
76 if(g_k>=32 && g_k<=126)
77 {g_k=g_k-32;
78 if(g_k>=0 && g_k<=45)
79 {karakter_genislik=font_width[g_k];
80 karakter_adres=g_k*5;
81 for (m=0; m<karakter_genislik; m++)
82 {output_b(font[karakter_adres]);
83 RGB();
84 delay_ms(sure);
85 karakter_adres++;}
86 bir_bosluk();}
87 else
88 {g_k=g_k-46;
89 karakter_genislik=font_width1[g_k];
90 karakter_adres=g_k*5;
91 for (m=0; m<karakter_genislik; m++)
92 {
93 output_b(font1[karakter_adres]);
94 RGB();
95 delay_ms(sure);
96 karakter_adres++;}
97 bir_bosluk();}
98 else
99 {
100 turkce_kar(g_k);
101 for (m=0; m<karakter_genislik; m++)
102 {
103 output_b(font2[karakter_adres]);
104 RGB();
105 delay_ms(sure);
106 karakter_adres++;}
107 bir_bosluk();}}}
108 reset_cpu();}

```

114	{for (z=0; z<8; z++)		
115	{digitalWrite(z,bitRead(font2[karakter_adres],z));		
116	delayMicroseconds(3);}		
117	RGB();		
118	delay(sure);		
119	karakter_adres++;}		
120	bir_bosluk();}}}		

İşlem Basamakları

1. Uygun tipli mikrodeneleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Dot matrixli kayan yazılı devreyi breadboard veya plaket üzerinde kurunuz.
6. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodeneleyiciye yükleyiniz. Devreyi çalıştırınız.
7. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Tek renkte havada yazı oluşturmak için programda yapılacak değişiklikler nelerdir?
2. Yazının maksimum 8 karakter (**char yazi[8]**) seçilmesinin sebebi nedir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



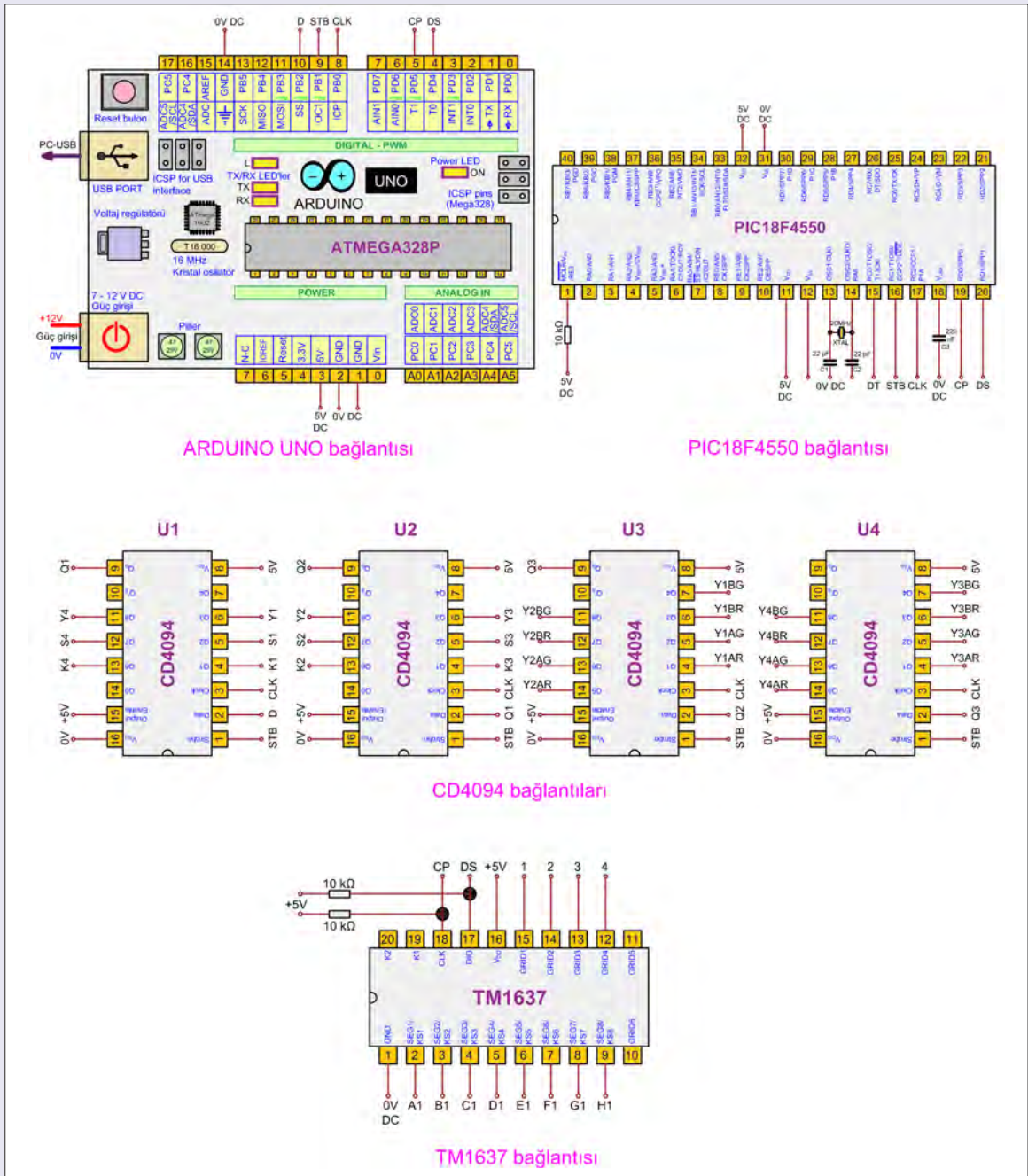
1.7. Trafik Lamba Devresi



Devre Şeması: Görşel 1.31'de dört kavşaklı trafik lamba devresi, Görşel 1.32'de trafik lamba kontrol devresi görölmektedir.



Görşel 1.31: Dört kavşaklı trafik lamba devresi



Görsel 1.32: Trafik lamba kontrol devresi

Devre Hakkında: Trafik lambası devresi; araç trafik ışıkları, yaya ışıkları, iki haneli sayıcılar, CD4094 kaymalı kaydedici entegreler ve TM1637 entegresi kullanılarak tasarlanmıştır. Mikrodenetleyici, kavşak verilerini seri olarak CD4094 entegrelerine gönderir ve veriler trafik ışıkları üzerinde görüntülenir. Mikrodenetleyici, sayıcı verilerini seri olarak TM1637 entegresine gönderir ve bu veriler sayıcılar üzerinde görüntülenir. Devreye enerji uygulandığında 1. ve 3. kavşak verileri referans alınır. Işık verileri **i_data[52]** dizisinden alınarak trafik lambalarında, ekran verileri **k1_3kavşak** ve **k2_4kavşak** değişkenlerinden alınarak 2 haneli sayıcılarda görüntülenir. Her saniyede yeni veriler trafik lambaları ve sayıcılarda görüntülenir. 1. ve 3. kavşak verileri işlenip bittiğinde 2. ve 4. kavşak verileri işlenmeye başlar. 2. ve 4. kavşak verileri işlenip bittiğinde program tekrar 1 ve 3. kavşak verilerini işlemeye başlar ve sonsuz döngüde çalışmaya devam eder (Görsel 1.32).

16	unsigned long int i_data[52]= {0x4481A5A5, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x4481A9A6, 0x84816595, 0x84826595, 0x84845555, 0x84845555, 0x84845555, 0x84845555, 0x84845555, 0x84445555, 0x84245956, 0x84245956, 0x82245956, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A 0x81246A9A, 0x82246A9A, 0x84246656, 0x84445656, 0x84845555, 0x84845555, 0x84845555, 0x84845555, 0x84845555, 0x84825555, 0x84816595, 0x84816595};	16	int k1_3kavsak=0, k2_4kavsak=0;
17	//alt programlar	17	int kavsak1_3=1, kavsak2_4=0;
18	void isik_yaz(int sayi)	18	int isik_kontrol=0, artir=0, i=0, isik_adres=0, s_kontrol=0;
19	{	19	unsigned int8
20	shiftOut(DATAS, CLKS, LSBFIRST, i_data[sayi]);	20	kavsak_sari[26]={1,0,0,0,0,0,0,0,0,0,0,0,0,0,1, 0,0,0,0,0,0,0,0,0,0};
21	shiftOut(DATAS, CLKS, LSBFIRST, i_data[sayi]>>8);	21	unsigned int32 i_data[52]= {0x4481A5A5, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x2481A9A6, 0x4481A9A6, 0x84816595, 0x84826595, 0x84845555, 0x84845555, 0x84845555, 0x84845555, 0x84845555, 0x84445555, 0x84245956, 0x84245956, 0x82245956, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x81246A9A, 0x82246A9A, 0x84246656 0x84445656, 0x84845555, 0x84845555, 0x84845555, 0x84845555, 0x84845555, 0x84825555, 0x84816595, 0x84816595};
22	shiftOut(DATAS, CLKS, LSBFIRST, i_data[sayi]>>16);	22	//alt programlar
23	shiftOut(DATAS, CLKS, LSBFIRST, i_data[sayi]>>24);	23	void isik_yaz(int sayi)
24	digitalWrite(STB, HIGH);	24	{int i=0,j=0;
25	delayMicroseconds(1);	25	for(i=0;i<1;i++)
26	digitalWrite(STB, LOW);}	26	{for(j=0;j<32;j++)
27	void sure_azalt()	27	{output_bit(ds,bit_test(i_data[sayi],j));
28	{k1_3kavsak--;	28	output_high(clock);
29	k2_4kavsak--;}	29	delay_us(1);
30	void ekran_yaz(int a,int b,int c,int d)	30	output_low(clock);}}
31	{ATM1637Display display(CLK, DIO);	31	output_high(latch);
32	display.clear();	32	delay_us(1);
33	display.setBrightness(0x0f);	33	output_low(latch);}
34	if (a==10)	34	void ekran_yaz(int a,int b,int c,int d)
35	{data[0]=64;	35	{Ekran(0);
36	data[1]=64;}	36	TM1637_display_segments(0, seg_data[a], 0);
37	else	37	TM1637_display_segments(1, seg_data[b], 0);
38	{data[0]=display.encodeDigit(a);	38	TM1637_display_segments(2, seg_data[c], 0);
39	data[1]=display.encodeDigit(b);}	39	TM1637_display_segments(3, seg_data[d], 0);}
40	if (c==10)	40	void sure_azalt()
41	{data[2]=64;		
42	data[3]=64;}		



```
43 else
44 {data[2]=display.encodeDigit(c);
45 data[3]=display.encodeDigit(d);}
46 display.setSegments(data);
47 display.setBrightness(0x0f);}
48 //genel ayarlar
49 void setup() {
50 pinMode(CLK, OUTPUT);
51 pinMode(DIO, OUTPUT);
52 pinMode(DATAS, OUTPUT);
53 pinMode(CLKS, OUTPUT);
54 pinMode(STB, OUTPUT);}
55 //ana program
56 void loop() {
57 if (kavsak1_3)
58 {k1_3kavsak=14;
59 k2_4kavsak=25;
60 while(isik_kontrol<26)
61 {isik_yaz(isik_adres);
62 if (kavsak_sari[s_kontrol])
63 {ekran_yaz(10,10,k2_4kavsak/10,k2_4kavsak%10);}
64 else
65 {ekran_yaz(k1_3kavsak/10, k1_3kavsak%10,
66 k2_4kavsak/10, k2_4kavsak%10);}
67 delay(1000);
68 sure_azalt();
69 isik_adres++;
70 isik_kontrol++;
71 if (isik_kontrol==15)
72 k1_3kavsak=35;
73 s_kontrol++;}}
74 kavsak1_3=0;
75 kavsak2_4=1;
76 s_kontrol=0;
77 isik_adres=26;
78 isik_kontrol=27;
79 if (kavsak2_4)
80 {k1_3kavsak=25;
81 k2_4kavsak=14;
82 while(isik_kontrol<53)
83 {isik_yaz(isik_adres);
84 if (kavsak_sari[s_kontrol])
85 {ekran_yaz(k1_3kavsak/10, k1_3kavsak%10,10,10);}
86 else
```

```
41 {k1_3kavsak--;
42 k2_4kavsak--;}
43 //genel ayarlar
44 void main()
45 {setup_timer_3(T3_DISABLED | T3_DIV_BY_1);
46 set_tris_a(0x00);
47 set_tris_b(0x00);
48 set_tris_c(0x00);
49 set_tris_d(0x00);
50 output_a(0x00);
51 output_b(0x00);
52 output_c(0x00);
53 output_d(0x00);
54 Ekran(0);
55 TM1637_init();
56 delay_ms(10);
57 //ana program
58 while(TRUE)
59 {if (kavsak1_3)
60 {k1_3kavsak=14;
61 k2_4kavsak=25;
62 while(isik_kontrol<26)
63 {isik_yaz(isik_adres);
64 if (kavsak_sari[s_kontrol])
65 {ekran_yaz(10,10, k2_4kavsak/10, k2_4kavsak%10);}
66 else
67 {ekran_yaz(k1_3kavsak/10, k1_3kavsak%10,
68 k2_4kavsak/10, k2_4kavsak%10);}
69 delay_ms(1000);
70 sure_azalt();
71 isik_adres++;
72 isik_kontrol++;
73 if (isik_kontrol==15)
74 k1_3kavsak=35;
75 s_kontrol++;}}
76 kavsak1_3=0;
77 kavsak2_4=1;
78 s_kontrol=0;
79 isik_adres=26;
80 isik_kontrol=27;
81 if (kavsak2_4)
82 {k1_3kavsak=25;
83 k2_4kavsak=14;
84 while(isik_kontrol<53)
```

87	{ekran_yaz(k1_3kavsak/10, k1_3kavsak%10, k2_4kavsak/10, k2_4kavsak%10);}	84	{isik_yaz(isik_adres);
88	delay(1000);	85	if (kavsak_sari[s_kontrol])
89	sure_azalt();	86	{ekran_yaz(k1_3kavsak/10, k1_3kavsak%10, 10, 10);}
90	isik_adres++;	87	else
91	isik_kontrol++;	88	{ekran_yaz(k1_3kavsak/10, k1_3kavsak%10, k2_4kavsak/10, k2_4kavsak%10);}
92	if (isik_kontrol==42)	89	delay_ms(1000);
93	k2_4kavsak=35;	89	sure_azalt();
94	s_kontrol++;}}	90	isik_adres++;
95	kavsak1_3=1;	91	isik_kontrol++;
96	kavsak2_4=0;	92	if (isik_kontrol==42)
97	s_kontrol=0;	93	k2_4kavsak=35;
98	isik_adres=0;	94	s_kontrol++;}}
99	isik_kontrol=0;}	95	kavsak1_3=1;
		96	kavsak2_4=0;
		97	s_kontrol=0;
		98	isik_adres=0;
		99	isik_kontrol=0;}}

İşlem Basamakları

1. Uygun tipli mikrodeneleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodeneleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Başlangıçta trafik lambaları ile sayıcıların 2 ve 4. kavşak bilgilerine göre çalışması için program üzerinde yapılacak değişiklikler nelerdir?
2. kavsak_sari[26] dizisinin program içindeki görevi nedir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



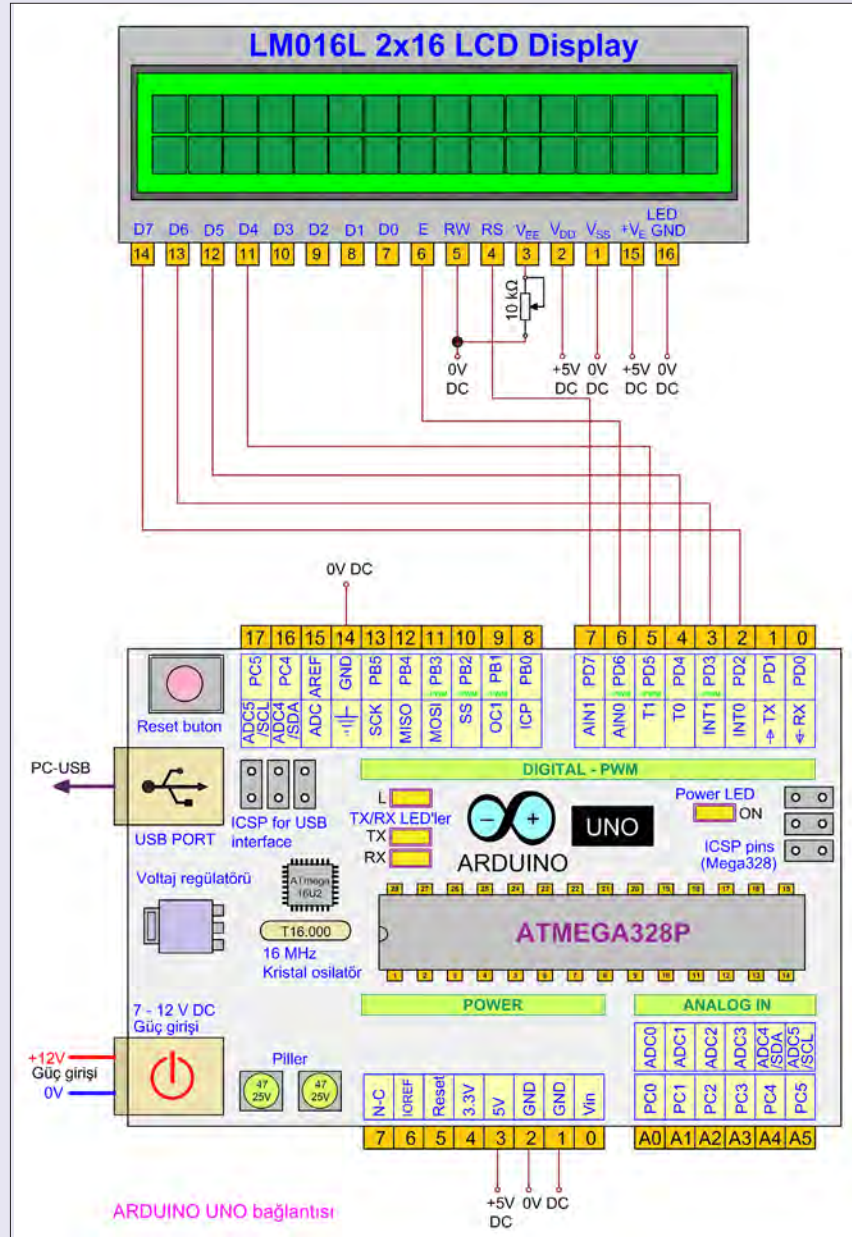
1.8. LCD UYGULAMALARI

LCD ekranlar bilgisayar veya mikrodeneleyicili sistemlerde görüntüleme amaçlı kullanılan karakter veya grafik tabanlı ekranlardır. Karakter tabanlı LCD'ler satır ve sütun sayısına göre isimlendirilir. 1x8, 2x8, 1x16, 2x16, 2x20, 4x16, 4x20 en çok kullanılan LCD türlerindedir. Karakter LCD ASCII formatında gönderilen verinin karşılığını 5x7 veya 5x8 font şeklinde istenilen satır ve sütunda görüntüler. Karakter LCD'lerle paralel veya seri haberleşme (I2C) yöntemleriyle iletişim sağlanır. Grafik tabanlı LCD'ler (GLCD) yatay (satır) ve dikey (sütun) piksellerden oluşur ve piksel sayısına göre isimlendirilir. 128x64, 240x64, 240x128, 320x240 en çok kullanılan GLCD türlerindedir. GLCD'lerle paralel veya seri haberleşme (I2C, SPI) yöntemleriyle iletişim sağlanır.

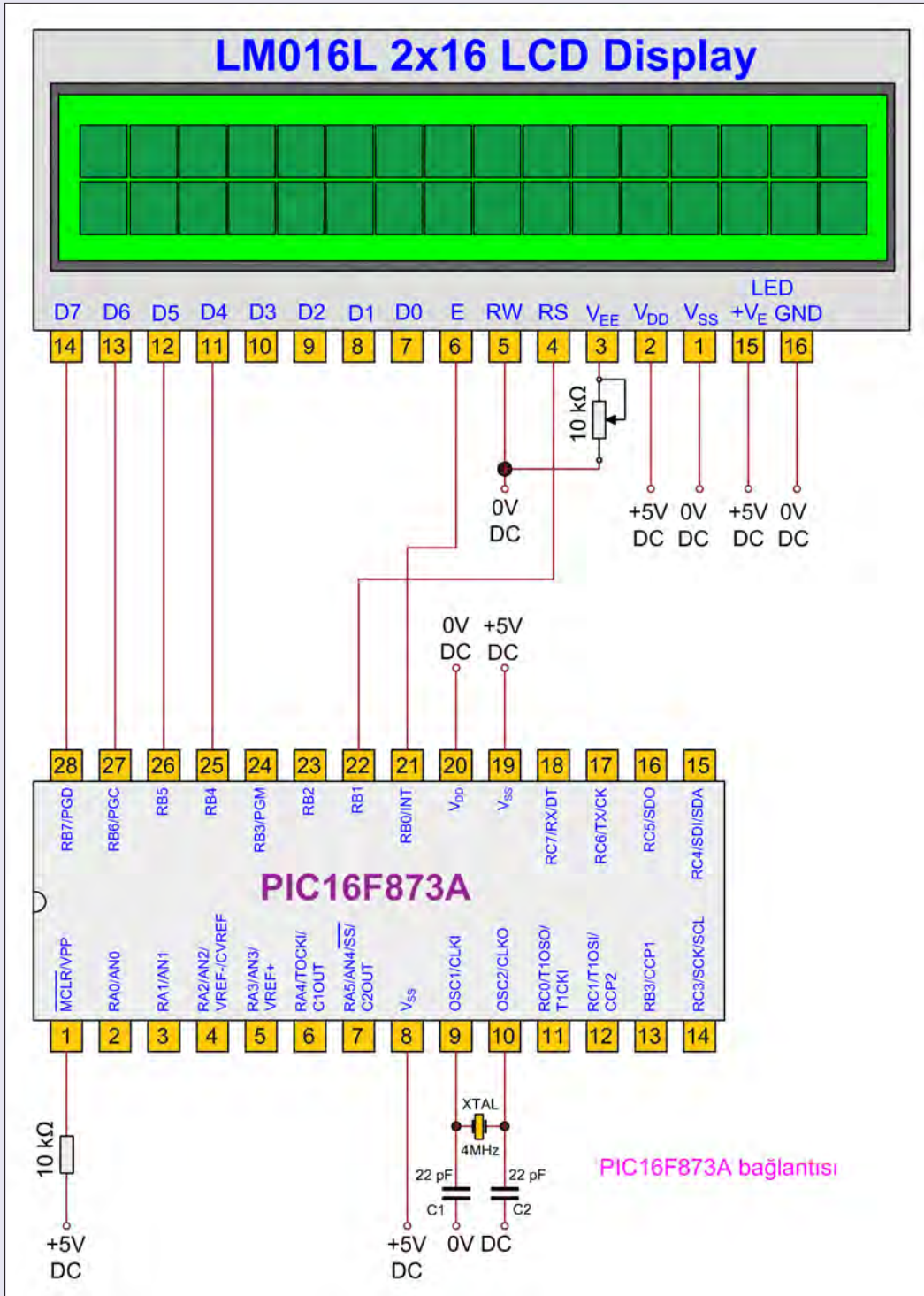


1.8.1. Paralel İletişimli LCD Uygulaması

Devre Şeması: Görsel 1.33'te ARDUINO UNO ile yapılan, Görsel 1.34'te PIC16F873A entegresi ile yapılan paralel iletişimli LCD devresi görülmektedir.



Görsel 1.33: ARDUINO UNO ile paralel iletişimli LCD devresi



Görsel 1.34: PIC16F873A entegresi ile paralel iletişimli LCD devresi

Devre Hakkında: Paralel iletişimli LCD devresi 4 bit data protokolünde veri alışverişini yapmaktadır. LCD bağlantıları doğrudan mikrodeneleyici pinlerine bağlanır. Devreye enerji uygulandığında imleç modları ekranda gösterilir. Satır sütun bilgisine göre adres konumlaması yapılır. Türkçe karakterler ekranda gösterilir. Birinci ve ikinci satıra yazı yazdırılır. Yazılan yazı sağa ve sola kaydırılır. LCD ekranda yapılabilen işlemlerin tamamının kodları programda yer almaktadır (Görsel 1.33-1.34).



Malzemeler: Tablo 1.17’de iki devrenin malzeme listesi görülmektedir.

Tablo 1.17: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F873A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
LM016L	2x16	1 adet
Potansiyometre	10 kΩ	1 adet
Direnç	10 kΩ	3 adet
Kristal osilatör	4 MHz, 20 MHz	2 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.18’de iki devrenin kod yazılımları görülmektedir.

Tablo 1.18: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F873A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <LiquidCrystal.h>	2	#include <16F873A.h>
3	#include <Wire.h>	3	#fuses XT, NOWDT, PUT, NOPROTECT, BROWNOUT, NOLVP
4	LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // (rs, enable, d4, d5, d6, d7)	4	#use delay(clock=4000000)
5	//alt programlar	5	#define use_portb_lcd TRUE
6	void sola_kaydir()	6	#include <lcd.c>
7	{lcd.scrollDisplayLeft();}	7	//alt programlar
8	void saga_kaydir()	8	void turkce_karakterler_yukle() {
9	{lcd.scrollDisplayRight();}	9	int j,k,cgram=64;
10	void turkce_karakterler_yukle()	10	const byte karakter_[8][8] = {
11	{byte ty_g[8] = {14, 0,15,17,15,1,14,0};	11	{0, 0, 0, 0, 0, 0, 0, 0}, //boş
12	byte ty_i[8] = {4,0,4,4,4,4,0};	12	{14, 0,15,17,15, 1,14, 0}, //ğ
13	byte ty_o[8] = {10,14,17,17,17,14,0};	13	{4,0,4,4,4,4,0}, //i
14	byte ty_s[8] = {15,16,16,14, 1,30, 4, 0};	14	{10,14,17,17,17,17,14,0}, //ö
15	byte ty_u[8] = {10,17,17,17,17,14,0};	15	{15,16,16,14, 1,30, 4, 0}, //ş
16	byte ty_c[8] = {14,17,16,16,21,14,4,0};	16	{10,17,17,17,17,17,14,0}, //ü
17	byte ty_bos[8] = {0, 0, 0, 0, 0, 0, 0, 0};	17	{14,17,16,16,21,14,4,0}, //ç
18	lcd.createChar(0, ty_g);	18	{0, 0, 0, 0, 0, 0, 0, 0} //boş};
19	lcd.createChar(1, ty_i);	19	for(j=0;j<=7;j++) {
20	lcd.createChar(2, ty_o);	20	lcd_send_byte(0,cgram);
21	lcd.createChar(3, ty_s);	21	for(k=0;k<=7;k++)
22	lcd.createChar(4, ty_u);	22	{lcd_send_byte(1,karakter_[j][k]);}

23	lcd.createChar(5, ty_c);	23	cgram=cgram+8; }}
24	lcd.createChar(6, ty_bos);}	24	void sola_kaydir()
25	//genel ayarlar	25	{lcd_send_byte(0, 0x18);}
26	void setup()	26	void saga_kaydir()
27	{lcd.begin(16, 2);	27	{lcd_send_byte(0, 0x1E);}
28	lcd.clear();	28	//genel ayarlar
29	turkce_karakterler_yukle();}	29	void main()
30	//ana program	30	{int a,i;
31	void loop()	31	lcd_init();
32	{lcd.clear();	32	turkce_karakterler_yukle();
33	lcd.print(" 4 BIT LCD KOD");	33	lcd_send_byte(0,0x28);
34	lcd.setCursor(0, 1);	34	lcd_putc("\f4 BIT LCD KOD");
35	lcd.print(" GONDERME UYG.");	35	lcd_putc("\n GONDERME UYG.");
36	delay(2000);	36	delay_ms(2000);
37	lcd.blink();	37	lcd_send_byte(0, 0x0D);
38	lcd.clear();	38	lcd_putc("\f FLASH IMLEC MODU");
39	lcd.print("FLASH IMLEC MODU");	39	lcd_putc("\n IMLEC ACIK");
40	lcd.setCursor(0, 1);	40	delay_ms(2000);
41	lcd.print(" IMLEC ACIK");	41	lcd_send_byte(0, 0x0C);
42	delay(2000);	42	lcd_send_byte(0, 0x0E);
43	lcd.noBlink();	43	lcd_putc("\f CIZGI IMLEC MODU");
44	lcd.clear();	44	lcd_putc("\n IMLEC SABIT");
45	lcd.cursor();	45	delay_ms(2000);
46	lcd.print("CIZGI IMLEC MODU");	46	lcd_send_byte(0, 0x0C);
47	lcd.setCursor(0, 1);	47	lcd_send_byte(0, 0x0F);
48	lcd.print(" IMLEC SABIT");	48	lcd_putc("\f CIZGI IMLEC MODU");
49	delay(2000);	49	lcd_putc("\n FLASH IMLEC");
50	lcd.noCursor();	50	delay_ms(2000);
51	lcd.command(0x0F);	51	lcd_send_byte(0, 0x0C);
52	lcd.clear();	52	lcd_gotoxy(1,1);
53	lcd.print("CIZGI IMLEC MODU");	53	lcd_putc("\f TURKCE KARAKTERLER");
54	lcd.setCursor(0, 1);	54	lcd_putc("\n ç ğ i ö ü ş");
55	lcd.print(" FLASH IMLEC");	55	delay_ms(2000);
56	delay(2000);	56	lcd_putc("\f ILERI MIKRODENETLEYICI UYGULAMALARI");
57	lcd.command(0x0C);	57	lcd_putc("\n EKRANDA GOSTER");
58	lcd.clear();	58	delay_ms(2000);
59	lcd.print("TURKCE KARAKTERLER");	59	//ana program
60	lcd.setCursor(3, 1);	60	while(1)
61	for(int i = 0; i < 7; i++) //40 karakter için kaydırma	61	{lcd_putc("\f ILERI MIKRODENETLEYICI UYGULAMALARI");
62	{lcd.write(byte(i));	62	lcd_putc("\n SAGA KAYDIR");
63	lcd.print(" ");}	63	delay_ms(1000);
64	delay(2000);	64	for(i = 0; i < 40; i++) //40 karakter için kaydırma
65	lcd.clear();	65	{saga_kaydir();



66	lcd.print("İLERİ MIKRODENETLEYİCİ UYGULAMALARI");	66	delay_ms(150); // Kayma hızı
67	lcd.setCursor(0, 1);	67	}
68	lcd.print("EKRANDA GOSTER");	68	lcd_putc("\fİLERİ MIKRODENETLEYİCİ UYGULAMALARI");
69	delay(2000);	69	lcd_putc("\n SOLA KAYDIR");
70	while(1)	70	delay_ms(1000);
71	{lcd.clear();	71	for(i = 0; i < 40; i++)
72	lcd.print("İLERİ MIKRODENETLEYİCİ UYGULAMALARI");	72	{sola_kaydir();
73	lcd.setCursor(0, 1);	73	delay_ms(150);}}
74	lcd.print(" SAGA KAYDIR");		
75	delay(1000);		
76	for(int i = 0; i < 40; i++) //40 karakter için kaydırma		
77	{saga_kaydir();		
78	delay(150); // Kayma hızı}		
79	lcd.clear();		
80	lcd.print("İLERİ MIKRODENETLEYİCİ UYGULAMALARI");		
81	lcd.setCursor(0, 1);		
82	lcd.print(" SOLA KAYDIR");		
83	delay(1000);		
84	for(int i = 0; i < 40; i++)		
85	{sola_kaydir();		
86	delay(150); }}		

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

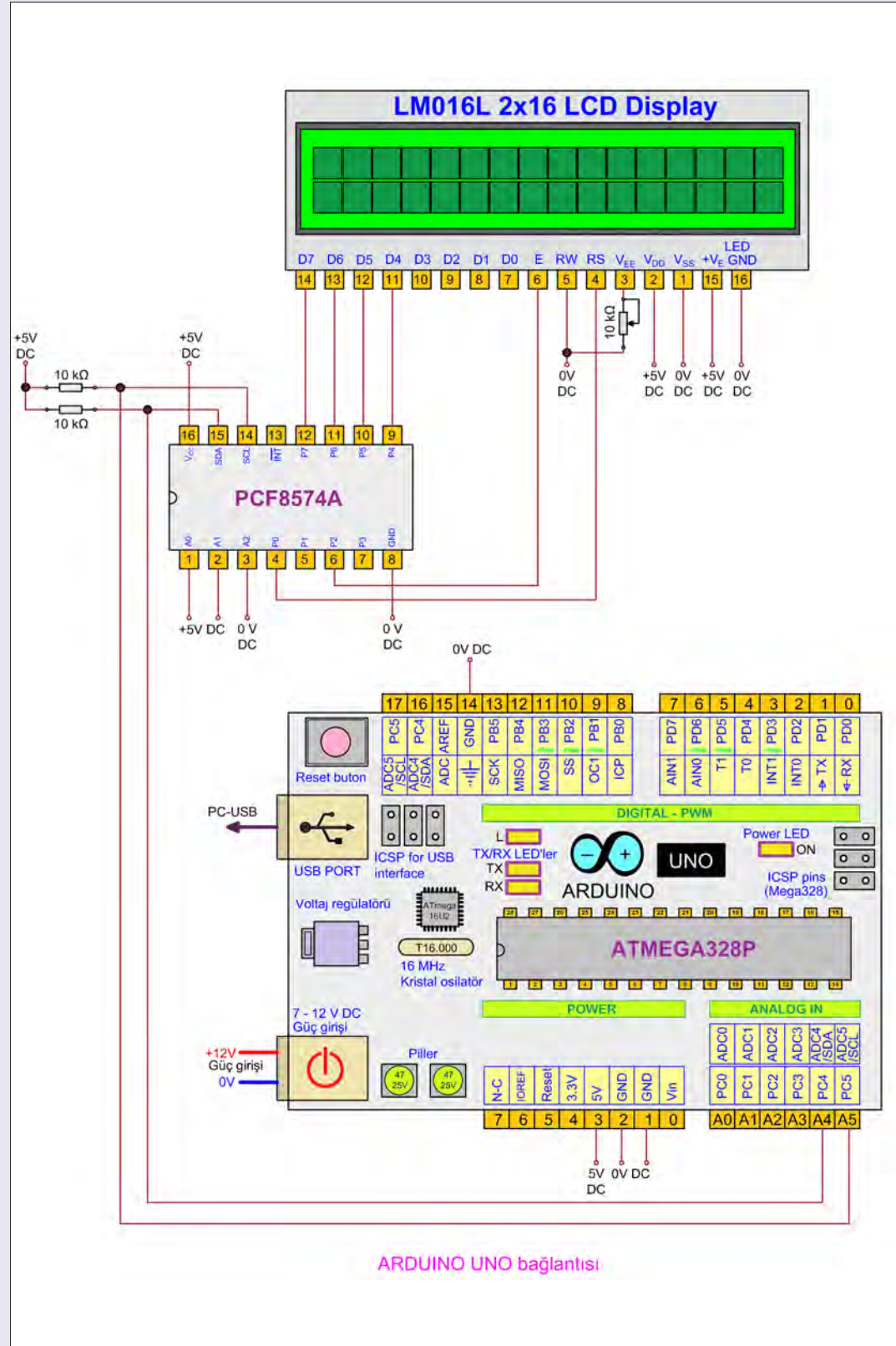
1. LCD modülde D0-D3 pinlerine bağlantı yapılmamasının sebebi nedir?
2. LCD modüle veri göndermeden önce **lcd.clear()** komutunun çalıştırılmasının sebebi nedir?



1.8.2. Seri İletişimli LCD Uygulaması

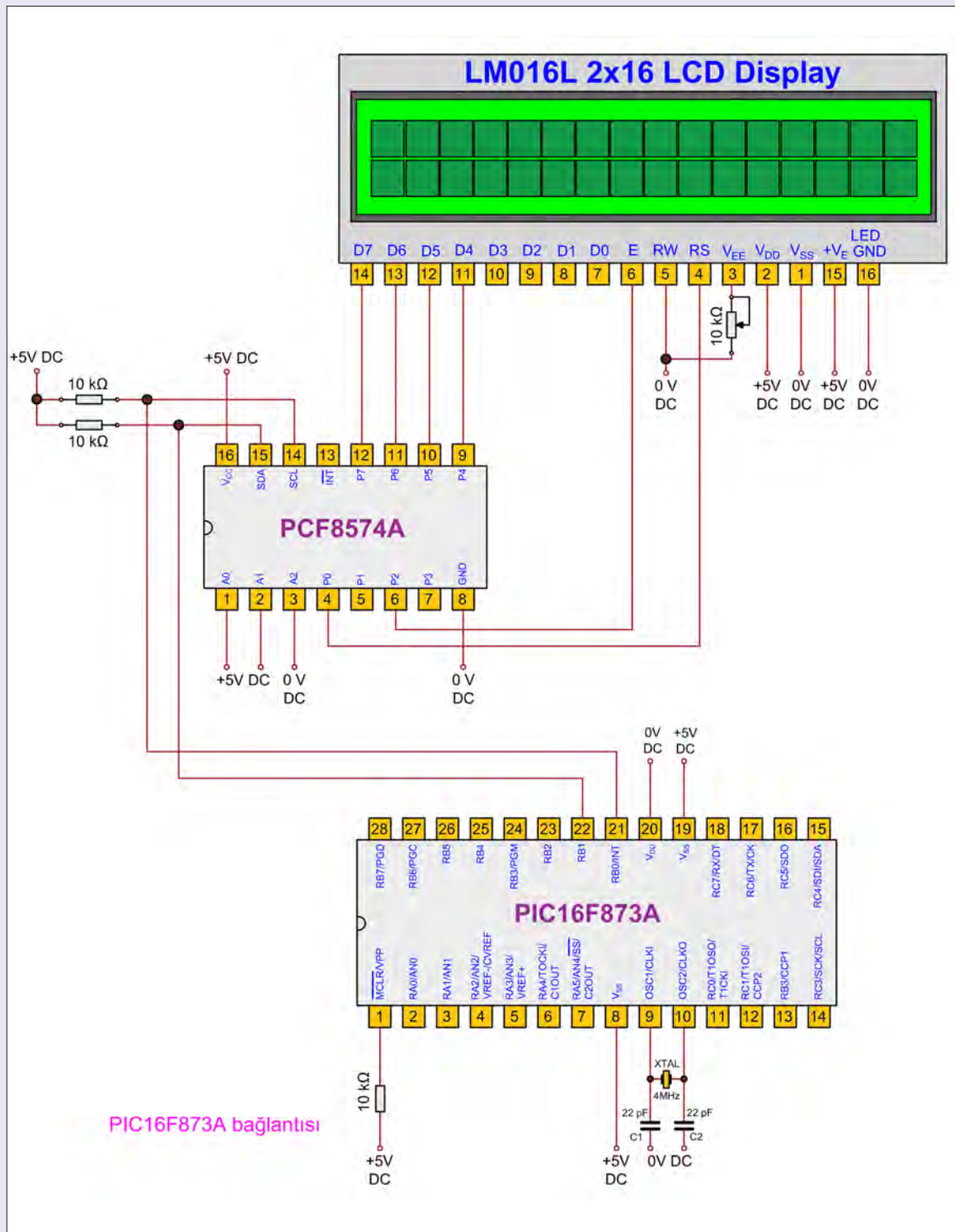


Devre Şeması: Görsel 1.35'te ARDUINO UNO ile yapılan, Görsel 1.36'da PIC16F873A entegresi ile yapılan seri iletişimli LCD devresi görülmektedir.



ARDUINO UNO bağlantısı

Görsel 1.35: ARDUINO UNO ile seri iletişimli LCD devresi



Görsel 1.36: PIC16F873A entegresi ile seri iletişimli LCD devresi

Devre Hakkında: Seri iletişimli LCD devresi I2C data protokolü kullanan PCF8574A entegresiyle iletişim yapmaktadır. LCD bağlantıları PCF8574A entegresinin P0-P7 pinlerine bağlanır. Devreye enerji uygulandığında imleç modları ekranda gösterilir. Satır sütun bilgisine göre adres konumlaması yapılır. Türkçe karakterler ekranda gösterilir. Birinci ve ikinci satıra yazı yazdırılır. Yazılan yazı sağa ve sola kaydırılır (Görsel 1.35-1.36).

Malzemeler: Tablo 1.19’da iki devrenin malzeme listesi görölmektedir.

Tablo 1.19: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodeneleyicili kart	1 adet
PIC16F628A	Mikrodeneleyici entegre	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
LM016L	2x16	1 adet
Potansiyometre	10 kΩ	1 adet
Direnç	10 kΩ	3 adet
Kristal osilatör	4 MHz, 20 MHz	2 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
PCF8574 entegresi		1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.20’de iki devrenin kod yazılımları görölmektedir.

Tablo 1.20: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F873A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <LiquidCrystal_I2C.h>	2	#include <16F873a.h>
3	LiquidCrystal_I2C lcd(0x23,16,2); //0x23 entegre tanımı ve adresi yanlış olduğunda iletişim kurulamaz.	3	#fuses XT, NOWDT, PUT, NOPROTECT, BROWNOUT, NOLVP
4	//alt programlar	4	#use delay(clock=4000000)
5	void sola_kaydir()	5	#define PCF_SDA PIN_B1
6	{lcd.scrollDisplayLeft();}	6	#define PCF_SCL PIN_B0
7	void saga_kaydir()	7	#use i2c(master, sda=PCF_SDA, scl=PCF_SCL)
8	{lcd.scrollDisplayRight();}	8	#define PCF8574_ID 3 // Entegre tanımı yanlış olduğunda iletişim kurulamaz.
9	void turkce_karakterler_yukle() {	9	#define PCF_ADD 64 // Entegre adresi yanlış olduğunda iletişim kurulamaz.
10	byte ty_g[8] = {14, 0,15,17,15, 1,14, 0};	10	#define RS P0
11	byte ty_i[8] = {4,0,4,4,4,4,0};	11	#define RW P1
12	byte ty_o[8] = {10,14,17,17,17,14,0};	12	#define E P2
13	byte ty_s[8] = {15,16,16,14, 1,30, 4, 0};	13	#define DB4 P4
14	byte ty_u[8] = {10,17,17,17,17,14,0};	14	#define DB5 P5
15	byte ty_c[8] = {14,17,16,16,21,14,4,0};	15	#define DB6 P6
16	byte ty_bos[8] = { 0, 0, 0, 0, 0, 0, 0, 0};	16	#define DB7 P7
17	lcd.createChar(0, ty_g);	17	#include <yt_lcd.c>
18	lcd.createChar(1, ty_i);	18	//alt programlar
19	lcd.createChar(2, ty_o);	19	void turkce_karakterler_yukle()
20	lcd.createChar(3, ty_s);		



21	lcd.createChar(4, ty_u);	20	{int j,k,cgram=64;
22	lcd.createChar(5, ty_c);	21	const byte karakter_[8][8] = {
23	lcd.createChar(6, ty_bos);}	22	{0,0,0,0,0,0,0,0}, //boş
24	//genel ayarlar	23	{14, 0,15,17,15, 1,14, 0}, //ğ
25	void setup()	24	{4,0,4,4,4,4,4,0}, //i
26	{lcd.begin();	25	{10,14,17,17,17,17,14,0}, //ö
27	lcd.clear();	26	{15,16,16,14, 1,30, 4, 0}, //ş
28	turkce_karakterler_yukle();}	27	{10,17,17,17,17,17,14,0}, //ü
29	//ana program	28	{14,17,16,16,21,14,4,0}, //ç
30	void loop()	29	{ 0, 0, 0, 0, 0, 0, 0, 0} //boş};
31	{lcd.clear();	30	for(j=0;j<=7;j++) {
32	lcd.print(" 4 BIT LCD KOD");	31	lcd_send_byte(0,cgram);
33	lcd.setCursor(0, 1);	32	for(k=0;k<=7;k++)
34	lcd.print(" GONDERME UYG.");	33	{lcd_send_byte(1,karakter_[j][k]);}
35	delay(2000);	34	cgram=cgram+8;}}
36	lcd.blink();	35	void sola_kaydir()
37	lcd.clear();	36	{lcd_send_byte(0, 0x18);}
38	lcd.print("FLASH IMLEC MODU");	37	void saga_kaydir()
39	lcd.setCursor(0, 1);	38	{lcd_send_byte(0, 0x1E);}
40	lcd.print("IMLEC ACIK");	39	//genel ayarlar
41	delay(2000);	40	void main()
42	lcd.noBlink();	41	{int a,i;
43	lcd.clear();	42	lcd_init();
44	lcd.cursor();	43	turkce_karakterler_yukle();
45	lcd.print("CIZGI IMLEC MODU");	44	lcd_send_byte(0,0x28);
46	lcd.setCursor(0, 1);	45	lcd_putc("\f 4 BIT LCD KOD");
47	lcd.print("IMLEC SABIT");	46	lcd_putc("\n GONDERME UYG.");
48	delay(2000);	47	delay_ms(2000);
49	lcd.noCursor();	48	lcd_send_byte(0, 0x0D);
50	lcd.command(0x0F);	49	lcd_putc("\f FLASH IMLEC MODU");
51	lcd.clear();	50	lcd_putc("\n IMLEC ACIK");
52	lcd.print("CIZGI IMLEC MODU");	51	delay_ms(2000);
53	lcd.setCursor(0, 1);	52	lcd_send_byte(0, 0x0C);
54	lcd.print(" FLASH IMLEC");	53	lcd_send_byte(0, 0x0E);
55	delay(2000);	54	lcd_putc("\f CIZGI IMLEC MODU");
56	lcd.command(0x0C);	55	lcd_putc("\n IMLEC SABIT");
57	lcd.clear();	56	delay_ms(2000);
58	lcd.print("TURKCE KARAKTERLER");	57	lcd_send_byte(0, 0x0C);
59	lcd.setCursor(3, 1);	58	lcd_send_byte(0, 0x0F);
60	for(int i = 0; i < 7; i++) //40 karakter için kaydırma	59	lcd_putc("\f CIZGI IMLEC MODU");
61	{lcd.write(byte(i));	60	lcd_putc("\n FLASH IMLEC");
62	lcd.print(" ");}	61	delay_ms(2000);
63	delay(2000);	62	lcd_send_byte(0, 0x0C);
64	lcd.clear();	63	lcd_gotoxy(1,1);

65	lcd.setCursor(0, 1);	64	lcd_putc("\f TURKCE KARAKTERLER");
66	lcd.print("EKRANDA GOSTER");	65	lcd_putc("\n ç ğ i ö ü ş");
67	delay(2000);	66	delay_ms(2000);
68	while(1)	67	lcd_putc("\f ILERI MIKRODENETLEYICI UYGULAMALARI");
69	{lcd.clear();	68	lcd_putc("\n EKRANDA GOSTER");
70	lcd.print("ILERI MIKRODENETLEYICI UYGULAMALARI");	69	delay_ms(2000);
71	lcd.setCursor(0,1);	70	//ana program
72	lcd.print("SAGA KAYDIR");	71	while(1)
73	delay(1000);	72	{lcd_putc("\f ILERI MIKRODENETLEYICI UYGULAMALARI");
74	for(int i = 0; i < 40; i++) //40 karakter için kaydırma	73	lcd_putc("\n SAGA KAYDIR");
75	{saga_kaydir();	74	delay_ms(1000);
76	delay(150); // Kayma hızı}	75	for(i = 0; i < 40; i++) //40 karakter için kaydırma
77	lcd.clear();	76	{saga_kaydir();
78	lcd.print("ILERI MIKRODENETLEYICI UYGULAMALARI");	77	delay_ms(150); // Kayma hızı}
79	lcd.setCursor(0, 1);	78	lcd_putc("\f ILERI MIKRODENETLEYICI UYGULAMALARI");
80	lcd.print("SOLA KAYDIR");	79	lcd_putc("\n SOLA KAYDIR");
82	delay(1000);	80	delay_ms(1000);
83	for(int i = 0; i < 40; i++)	81	for(i = 0; i < 40; i++)
84	{sola_kaydir();	82	{sola_kaydir();
85	delay(150);}}}	83	delay_ms(150); }}

İşlem Basamakları

1. Uygun tipli mikrodenetleyiciyi (PIC veya ARDUINO) seçiniz
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodenetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

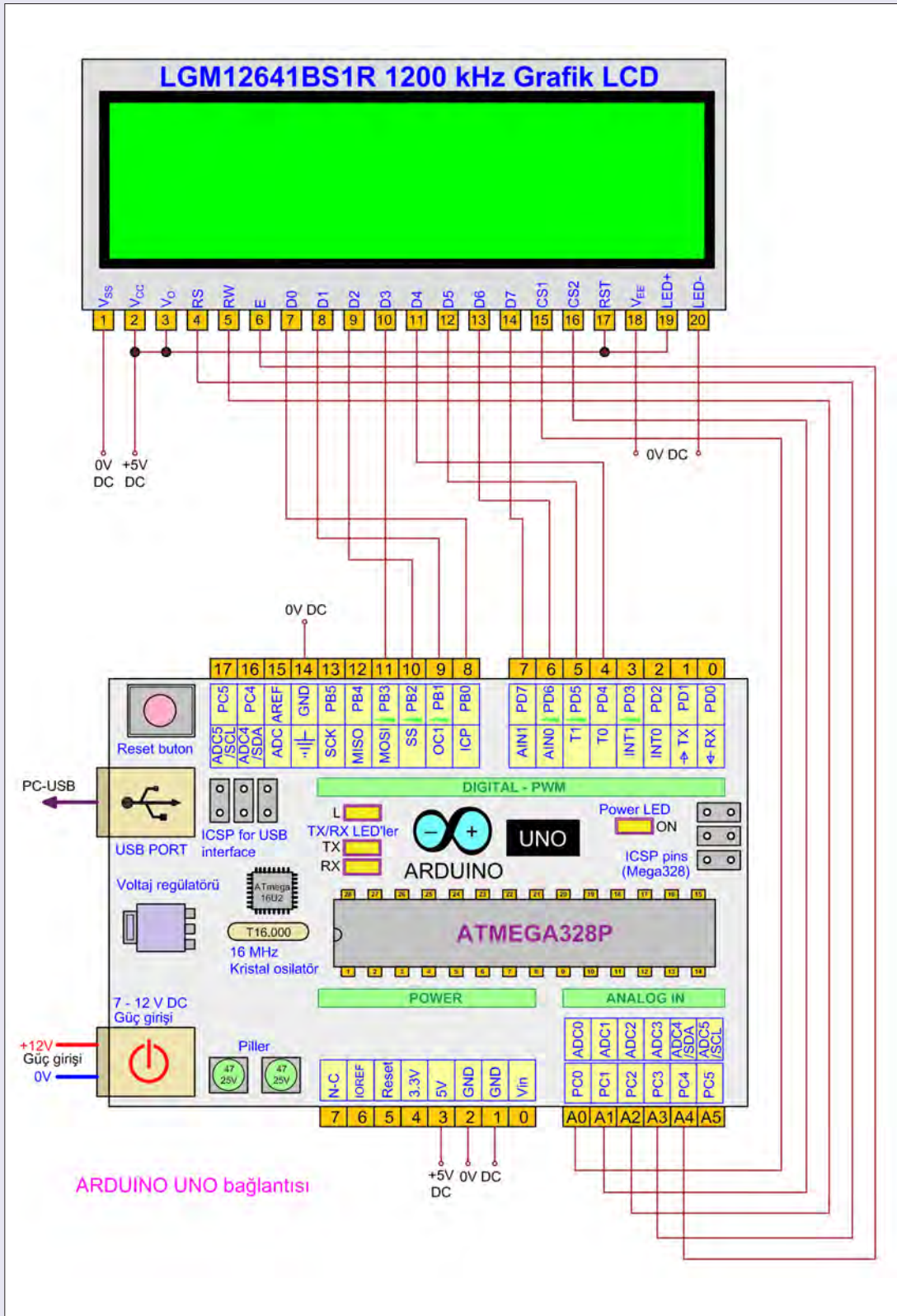
1. Seri LCD modülü ile paralel LCD modülü arasındaki donanım farkı nedir?.
2. Seri LCD modülü iletişimde hangi veri iletim yöntemi kullanılır?



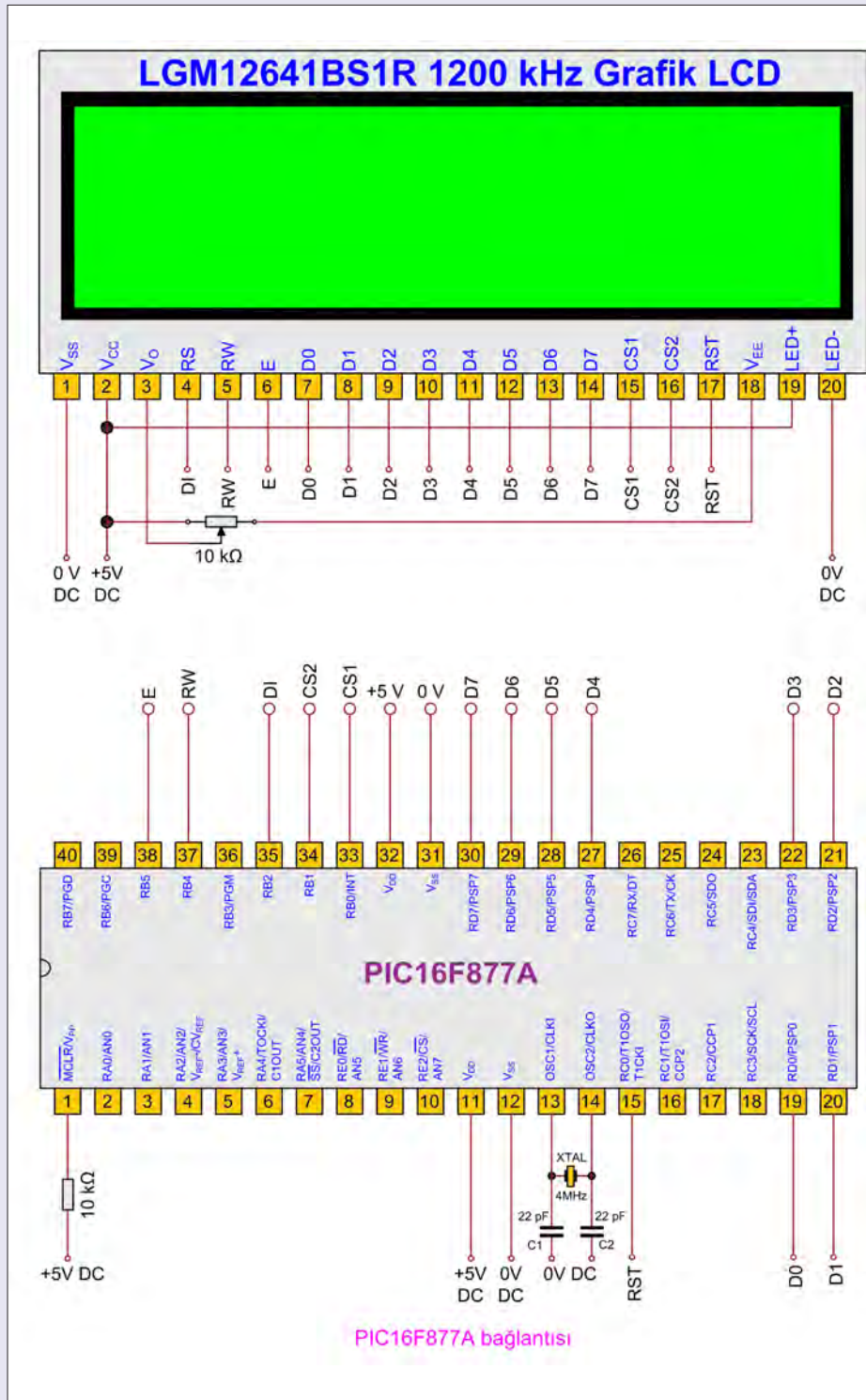
1.8.3. Grafik LCD Uygulaması



Devre Şeması: Görsel 1.37’de ARDUINO UNO ile yapılan, Görsel 1.38’de PIC16F877A entegresi ile yapılan paralel iletişimli grafik LCD (GLCD) devresi görülmektedir.



Görsel 1.37: ARDUINO UNO ile paralel iletişimli GLCD devresi



Görsel 1.38: PIC16F877A entegresi ile paralel iletişimli GLCD devresi

Devre Hakkında: Paralel iletişimli GLCD devresi 8 bit data protokolünde veri alışverişi yapmaktadır. GLCD bağlantıları doğrudan mikrodeneleyici pinlerine bağlanır. Devreye enerji uygulandığında ekranda yazı yazdırılır. Satır sütun bilgisine göre adres konumlaması yaptırılır. Bu ekranda çizgi, boş daire, dolu daire ve dikdörtgen çizdirilir. GLCD kütüphanesine yeni kodlar eklenerek daha farklı GLCD komutları oluşturmak mümkündür (Görsel 1.37-1.38).



Malzemeler: Tablo 1.21’de iki devrenin malzeme listesi görülmektedir.

Tablo 1.21: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F877A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
Grafik LCD	LGM12641BS1R, 128x64	1 adet
Direnç	10 kΩ	2 adet
Potansiyometre	10 kΩ	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.22’de iki devrenin kod yazılımları görülmektedir.

Tablo 1.22: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F877A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <glcd.h>	2	#include <16F873a.h>
3	#include "fonts/allFonts.h"	3	#fuses HS, NOWDT, NOPROTECT, NOBROWNOUT, NOLVP, NOPUT, NOWRT, NODEBUG, NOCPD
4	//değişkenler	4	#use delay (clock=20000000)
5	int i,j,x1,y1,x2,y2, xeksen,yeksen;	5	#include <HDM64GS12.c>
6	//genel ayarlar	6	#include <graphics.c>
7	void setup()	7	//değişkenler
8	{GLCD.Init();	8	char yazı_a[]="İLERİ";
9	GLCD.ClearScreen();	9	char yazı_b[]="MIKRODENETLEYİCİ";
10	GLCD.SelectFont(System5x7, BLACK); }	10	char yazı_c[]="UYGULAMA";
11	//ana program	11	char yazı_d[]="KİTABI";
12	void loop()	12	char yazı_e[]="GRAFİK LCD UYG.";
13	{GLCD.ClearScreen();	13	int i, j, x1, y1, x2, y2, xeksen, yeksen;
14	GLCD.CursorTo(8,0);	14	//genel ayarlar
15	GLCD.print("İLERİ");	15	while(1)
16	GLCD.SelectFont(Arial_bold_14);	16	{glcd_init(ON); // Ekran siliniyor.
17	GLCD.CursorTo(0,1);	17	glcd_text57(35, 0, yazı_a, 2, ON); // GLCD’de yazı yazdırılıyor.
18	GLCD.print("MIKRODENETLEYİCİ");	18	glcd_text57(18, 18, yazı_b, 1, ON);
19	GLCD.SelectFont(System5x7, BLACK);	19	glcd_text57(23, 28, yazı_c, 2, ON);
20	GLCD.CursorTo(5,5);	20	glcd_text57(45, 45, yazı_d, 1, ON);
21	GLCD.print("UYGULAMALARI");	21	glcd_text57(20, 55, yazı_e, 1, ON);
22	GLCD.CursorTo(8,6);	22	delay_ms(2000);
23	GLCD.print("KİTABI");		

24	GLCD.CursorTo(4,7);	23	glcd_init(ON); // Ekran siliniyor.
25	GLCD.print("GRAFIK LCD UYG.");	24	x1=5;
26	delay(2000);	25	y1=5;
27	GLCD.ClearScreen();	26	x2=59;
28	x1=5;	27	y2=59;
29	y1=5;	28	for(i=0;i<6;i++)
30	x2=54;	29	{glcd_rect(x1, y1, x2, y2, NO, ON); // GLCD'de dikdörtgen
31	y2=54;		çizdiriliyor.
32	for(i=0;i<6;i++)	30	y1=y1+5;
33	{GLCD.DrawRect(x1, y1, x2, y2, BLACK); // GLCD'de	31	x1=x1+5;
	dikdörtgen çizdiriliyor.	32	x2=x2-5;
34	y1=y1+5;	33	y2=y2-5;
35	x1=x1+5;	34	delay_ms(500);}
36	x2=x2-10;	35	x1=94;
37	y2=y2-10;	36	y1=30;
38	delay(500);}	37	x2=98;
39	x1=94;	38	y2=34;
40	y1=30;	39	for(i=0;i<6;i++)
41	x2=4;	40	{glcd_rect(x1, y1, x2, y2, YES, ON); // GLCD'de içi dolu
42	y2=4;		dikdörtgen çizdiriliyor.
43	for(i=0;i<6;i++)	41	y1=y1-5;
44	{GLCD.FillRect(x1, y1, x2, y2, BLACK); // GLCD'de içi dolu	42	x1=x1-5;
	dikdörtgen çizdiriliyor.	43	x2=x2+5;
45	y1=y1-5;	44	y2=y2+5;
46	x1=x1-5;	45	delay_ms(500);}
47	x2=x2+10;	46	glcd_init(ON); // Ekran siliniyor.
48	y2=y2+10;	47	for(i=0;i<=24;i=i+3)
49	delay(500);}	48	{glcd_circle(32, 32, i, NO, ON); // GLCD'de daire çizdiriliyor.
50	GLCD.ClearScreen();	49	delay_ms(1000);}
51	for(i=0;i<=24;i=i+3)	50	for(i=0;i<=24;i=i+3)
52	{GLCD.DrawCircle(32, 32, i+1, BLACK); // GLCD'de	51	{glcd_circle(96, 32, i, YES, ON); // GLCD'de içi dolu daire
	daire çizdiriliyor.		çizdiriliyor.
53	delay(500);}	52	delay_ms(1000);}
54	for(i=0;i<=24;i=i+3)	53	glcd_init(ON); // Ekran siliniyor.
55	{GLCD.FillCircle(96, 32, i+1, BLACK); // GLCD'de içi dolu	54	y1=48;
	daire çizdiriliyor.	55	x1=5;
56	delay(500);}	56	x2=5;
57	GLCD.ClearScreen();	57	y2=60;
58	y1=48;	58	for (i=0;i<12;i++)
59	x1=5;	59	{glcd_bar(x1, 0, x2, y2, 5, ON); // GLCD'de çubuk çizdiriliyor.
60	x2=5;	60	delay_ms(1000);
61	y2=60;	61	x1=x1+10;
62	for (i=0;i<12;i++)	62	y1=y1-8;
63	{GLCD.FillRect(x1, 0, x2, y2, BLACK); // GLCD'de çubuk	63	x2=x2+10;
	çizdiriliyor.	64	y2=y2-4; }
64	delay(1000);	65	glcd_init(ON); // Ekran siliniyor.
65	x1=x1+10;	66	xeksen=5;yeksen=32;



66	y1=y1-8;	67	for (i=0;i<10;i++)
67	y2=y2-4; }	68	{x1=xeksen+i;
68	GLCD.ClearScreen();	69	y1=yeksen;
69	xeksen=5;yeksen=32;	70	glcd_pixel(x1,y1,ON); // Ekranda istenen pikseller
70	for (i=0;i<10;i++)		aktif yapılıyor.
71	{x1=xeksen+i;	71	delay_ms(50);}
72	y1=yeksen;	72	xeksen=x1;
73	GLCD.SetDot(x1,y1,BLACK); // Ekranda istenen pikseller	73	yeksen=y1;
	aktif yapılıyor.	74	for (i=0;i<6;i++)
74	delay(50);}	75	{x1=xeksen+i;y1=yeksen-i;
75	xeksen=x1;	76	glcd_pixel(x1,y1,ON); // Ekranda istenen pikseller
76	yeksen=y1;		aktif yapılıyor.
77	for (i=0;i<6;i++)	77	delay_ms(50);}
78	{x1=xeksen+i;y1=yeksen-i;	78	xeksen=x1;
79	GLCD.SetDot(x1,y1,BLACK); // Ekranda istenen pikseller	79	yeksen=y1;
	aktif yapılıyor.	80	for (i=0;i<12;i++)
80	delay(50);}	81	x1=xeksen+i;y1=yeksen+i;
81	xeksen=x1;	82	glcd_pixel(x1,y1,ON); // Ekranda istenen pikseller
82	yeksen=y1;		aktif yapılıyor.
83	for (i=0;i<12;i++)	83	delay_ms(50);}
84	{x1=xeksen+i;y1=yeksen+i;	84	xeksen=x1;
85	GLCD.SetDot(x1,y1,BLACK); // Ekranda istenen pikseller	85	yeksen=y1;
	aktif yapılıyor.	86	for (i=0;i<12;i++)
86	delay(50);}	87	{x1=xeksen+i;y1=yeksen-i;
87	xeksen=x1;	88	glcd_pixel(x1,y1,ON); // Ekranda istenen pikseller
88	yeksen=y1;		aktif yapılıyor.
89	for (i=0;i<12;i++)	89	delay_ms(50);}
90	{x1=xeksen+i;y1=yeksen-i;	90	xeksen=x1;
91	GLCD.SetDot(x1,y1,BLACK); // Ekranda istenen pikseller	91	yeksen=y1;
	aktif yapılıyor.	92	for (i=0;i<6;i++)
92	delay(50);}	93	{x1=xeksen+i;y1=yeksen+i;
93	xeksen=x1;	94	glcd_pixel(x1,y1,ON); // Ekranda istenen pikseller
94	yeksen=y1;		aktif yapılıyor.
95	for (i=0;i<6;i++)	95	delay_ms(50);}
96	{x1=xeksen+i;y1=yeksen+i;	96	xeksen=x1;
97	GLCD.SetDot(x1,y1,BLACK); // Ekranda istenen pikseller	97	yeksen=y1;
	aktif yapılıyor.	98	for (i=0;i<12;i++)
98	delay(50);}	99	{x1=xeksen+i;y1=yeksen;
99	xeksen=x1;	100	glcd_pixel(x1,y1,ON); // Ekranda istenen pikseller
100	yeksen=y1;		aktif yapılıyor.
101	for (i=0;i<12;i++)	101	delay_ms(50);}
102	{x1=xeksen+i;y1=yeksen;	102	glcd_init(ON); // Ekran siliniyor.
103	GLCD.SetDot(x1,y1,BLACK); // Ekranda istenen pikseller	103	x1=5;
	aktif yapılıyor.	104	y1=48;
104	delay(50);}.	105	x2=5;
105	GLCD.ClearScreen();	106	y2=60;
106	x1=5;	107	for(i=0;i<6;i++)

107	y1=48;	108	{glcd_line(x1, y1, x2, y2, ON); // GLCD'de çizgi çizdiriliyor
108	x2=5;	109	x1=x1+10;
109	y2=60;	110	y1=y1-8;
110	for(i=0;i<6;i++)	111	x2=x2+10;
111	{GLCD.DrawLine(x1, y1, x2, y2, BLACK); // GLCD'de çizgi çizdiriliyor.	112	delay_ms(500);}
112	x1=x1+10;	113	for(i=0;i<=6;i++)
113	y1=y1-8;	114	{glcd_line(x1, y1, x2, y2, ON);
114	x2=x2+10;	115	x1=x1+10;
115	delay(500);}	116	y1=y1+8;
116	for(i=0;i<=6;i++)	117	x2=x2+10;
117	{GLCD.DrawLine(x1, y1, x2, y2, BLACK);	118	delay_ms(500);}
118	x1=x1+10;	119	while(1);
119	y1=y1+8;	120	glcd_init(ON); // Ekran siliniyor.
120	x2=x2+10;	121	for(i=0;i<=30;i=i+5)
121	delay(500);}}	124	{glcd_circle(60, 30, i, NO, ON); // GLCD'de daire çizdiriliyor.
		125	delay_ms(1000);}
		126	glcd_init(ON); // Ekran siliniyor.
		127	x1=5;
		128	y1=5;
		129	x2=120;
		130	y2=63;
		131	for(i=0;i<6;i++)
		132	{glcd_rect(x1, y1, x2, y2, NO, ON); // GLCD'de dikdörtgen çizdiriliyor.
		133	y1=x1+=5;
		134	x2-=5;
		135	y2-=5;
		136	delay_ms(500);}}

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. GLCD'de her bir pikselin kontrol edilebilmesinin avantajı nedir?
2. GLCD'de dikdörtgen, daire vb. çizim komutlarının önceden hazırlanmış olmasının avantajları nelerdir?



Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

A) PARALEL İLETİŞİMLİ LCD UYGULAMASI

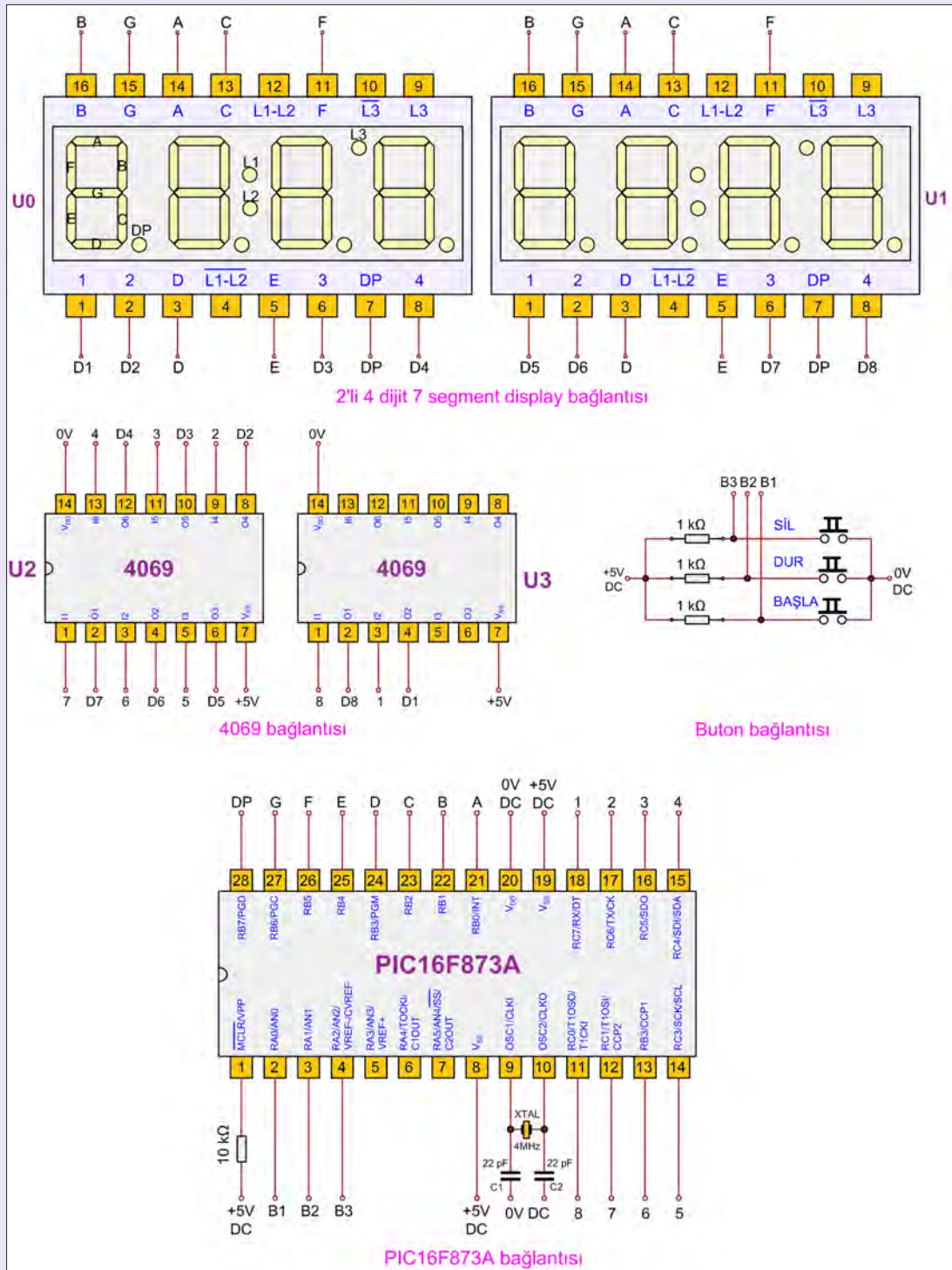
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

B) SERİ İLETİŞİMLİ LCD UYGULAMASI

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

C) GRAFİK LCD UYGULAMASI

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



Görsel 1.40: PIC16F873A entegresi ile kronometre devresi

Devre Hakkında: Kronometre devresinde iki ayrı tasarım sunulmaktadır. ARDUINO UNO ile yapılan tasarımda MAX7219 seri iletişimli 7 segment display kartı kullanılmaktadır. Kronometre verileri display kartına, ilgili displayde görüntülenmek üzere seri olarak gönderilir. PIC mikrodeneleyici ile yapılan tasarımda kronometre verileri paralel olarak display modülüne gönderilir. CD4069 entegresiyle multipleks tarama yöntemi kullanılarak ilgili displayde görüntülenir. Devreye enerji uygulandığında tüm displaylere 0 verisi yüklenir. BAŞLA butonuna basıldığında timer zamanlayıcısı çalışmaya başlar

ve her salisenin yüzde birinde kesme üretir. Mikrodenetleyici her kesmede sayıcıların değerlerini hesaplar ve sayıcılar içerisine kaydeder. Kesme olmadığı durumlarda sayıcılar içinde saklanan sayı değerleri display ekranda görüntülenir. DUR butonuna basıldığında timer zamanlayıcı çalışmayı durdurur ve kesme üretmez. Sayıcılar içinde saklanan sayı değerleri display ekranda görüntülenir. SİL butonuna basıldığında timer zamanlayıcı ve sayıcılar sıfırlanır (Görsel 1.39-1.40).

Malzemeler: Tablo 1.23'te iki devrenin malzeme listesi görülmektedir.

Tablo 1.23: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F873A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
CD4069	DEĞİL kapısı	2 adet
MAX7219 display modülü	Hazır PCB	1 adet
7 segment display	4 haneli ortak katot	2 adet
Buton	Mikro siviç	3 adet
Direnç	1 kΩ	3 adet
Direnç	10 kΩ	1 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.24'de iki devrenin kod yazılımları görülmektedir.

Tablo 1.24: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F873A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <SPI.h>	2	#include <16F873a.H>
3	#include <Adafruit_GFX.h>	3	#fuses XT, NOWDT, NOPROTECT, BROWNOUT, PUT, NOLVP
4	#include <LEDControl.h>	4	#use delay(clock=4000000)
5	#define basla A0	5	#use fast_io(b)
6	#define dur A1	6	#use fast_io(a)
7	#define sil A2	7	#use fast_io(c)
8	LEDControl yt = LEDControl(9, 8, 7, 1);	8	#define start pin_a0
9	//değişkenler	9	#define stop pin_a1
10	int ss=0, dk=0, sn=0, ss2=0, dk2=0, sn2=0 ;	10	#define reset pin_a2
11	word sl = 0;	11	#define dsp1 pin_c0
12	bool zamanlayici = false;	12	#define dsp10 pin_c1
13	char coden[]={ 0x7E, 0x30, 0x6D, 0x79, 0x33,	13	#define dsp100 pin_c2
	0x5B, 0x5F, 0x70, 0x7F, 0x73}; //KATOT	14	#define dsp1000 pin_c3



```

14 //alt programlar
15 ISR(TIMER1_COMPA_vect){
16 if(zamanlayici == true){sl++;
17 if(sl >= 100){sl = 0;
18 sn=sn+1;
19 if(sn>59){sn=0; dk=dk+1;}
20 if(dk>59){dk=0; ss=ss+1;}
21 if(ss>23){ss=0;}}}}
22 //genel ayarlar
23 void setup()
24 {
25 pinMode(basla, INPUT_PULLUP);
26 pinMode(dur, INPUT_PULLUP);
27 pinMode(sil, INPUT_PULLUP);
28 yt.shutdown(0, false);
29 yt.setIntensity(0, 5);
30 yt.clearDisplay(0);
31 noInterrupts();
32 TCCR1A = 0;
33 TCCR1B = 0;
34 TCNT1 = 0;
35 OCR1A = 60;
36 TCCR1B |= (1 << WGM12);
37 TCCR1B |= (1 << CS12) | (1 << CS10);
38 TIMSK1 |= (1 << OCIE1A);
39 interrupts();}
40 //ana program
41 void loop()
42 {
43 if(digitalRead (basla) == 0){
44 delay(100);
45 while (!digitalRead (basla));
46 zamanlayici = true;}
47 if(digitalRead (dur) == 0){
48 zamanlayici = false;
49 delay(100);}
50 if(digitalRead (sil) == 0){
51 ss=0, dk=0, sn=0, sl=0;
52 delay(100);}
53 yt.setRow(0,7,coden[ss/10]);
54 ss2=coden[ss%10];
55 bitSet (ss2,7);
56 yt.setRow(0,6,ss2);
57 yt.setRow(0,5,coden[dk/10]);
58 dk2=coden[dk%10];
59 bitSet (dk2,7);
60 yt.setRow(0,4,dk2);

```

```

15 #define dsp1s pin_c4
16 #define dsp10s pin_c5
17 #define dsp100s pin_c6
18 #define dsp1000s pin_c7
19 //değişkenler
20 char birler=0,onlar=0,yuzler=0,sayi=0, binler=0, onbin=0,
yuzbin=0, salise=0, saniye=0, dakika=0, saat=0, mil=0, mil2=0;
21 const int digit[10]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7c,
0x07, 0x7f, 0x6f};
22 //alt programlar
23 #int_timer0
24 void timer0_kesme () // Kesme fonksiyonu ismi
25 {disable_interrupts(INT_timer0);
26 salise++;
27 if (salise>=100)
28 {salise=0;
29 saniye++;
30 if (saniye>=60)
31 {saniye=0;
32 dakika++;
33 if (dakika>=60)
34 {dakika=0;
35 saat++;
36 if (saat>24)
37 {saat=0;}}}}
38 set_timer0(132);
39 enable_interrupts(INT_timer0);
40 enable_interrupts(GLOBAL);}
41 //genel ayarlar
42 void main()
43 {setup_timer_0(RTCC_INTERNAL|RTCC_DIV_64);
44 setup_timer_1(T1_DISABLED);
45 setup_timer_2(T2_DISABLED,0,1);
46 setup_comparator(NC_NC_NC_NC);
47 setup_vref(FALSE);
48 set_tris_a(0b00000111);
49 set_tris_b(0x00);
50 output_b(0x00);
51 output_a(0x00);
52 set_tris_c(0x00);
53 output_c(0x00);
54 disable_interrupts(INT_timer0);
55 enable_interrupts(GLOBAL);
56 //ana program
57 while(1)
58 {if (input(start)==0)
59 {delay_ms(5);

```

```

61 yt.setRow(0,3,coden[sn/10]);
62 sn2=coden[sn%10];
63 bitSet (sn2,7);
64 yt.setRow(0,2,sn2);
65 yt.setRow(0,1,coden[sl/10]);
66 yt.setRow(0,0,coden[sl%10]);
67 delay(50);}

```

```

60 while(!input(start));
61 set_timer0(132);
62 enable_interrupts(INT_timer0);}
63 if (input(stop)==0)
64 {delay_ms(5);
65 disable_interrupts(INT_timer0);
66 while(!input(stop));}
67 if (input(reset)==0)
68 {delay_ms(5);
69 disable_interrupts(INT_timer0);
70 salise=0, saniye=0, dakika=0;
71 while(!input(reset));}
72 birler=salise%10;
73 onlar=salise/10;
74 yuzler=saniye%10;
75 binler=saniye/10;
76 onbin=dakika%10;
77 yuzbin=dakika/10;
78 mil=saat%10;
79 mil2=saat/10;
80 output_b(digit[birler]);
81 output_high(dsp1);
82 delay_ms(5);
83 output_low(dsp1);
84 output_b(digit[onlar]);
85 output_high(dsp10);
86 delay_ms(5);
87 output_low(dsp10);
88 output_b(digit[yuzler]);
89 output_high(pin_b7);
89 output_high(dsp100);
90 delay_ms(5);
91 output_low(dsp100);
92 output_low(pin_b7);
93 output_b(digit[binler]);
94 output_high(dsp1000);
95 delay_ms(5);
96 output_low(dsp1000);
97 output_b(digit[onbin]);
98 output_high(pin_b7);
99 output_high(dsp1s);
100 delay_ms(5);
101 output_low(dsp1s);
102 output_b(digit[yuzbin]);
103 output_high(dsp10s);
104 delay_ms(5);
105 output_low(dsp10s);

```



		106	output_b(digit[mil]);
		107	output_high(pin_b7);
		108	output_high(dsp100s);
		109	delay_ms(5);
		110	output_low(dsp100s);
		111	output_b(digit[mil2]);
		112	output_high(dsp1000s);
		113	delay_ms(5);
		114	output_low(dsp1000s); }}

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Timer zamanlayıcı her saniye bir kesme üretirse devre çalışmasında nasıl bir değişiklik olur?
2. Programda kesmelerin aktif edilmesinin sebebi nedir?

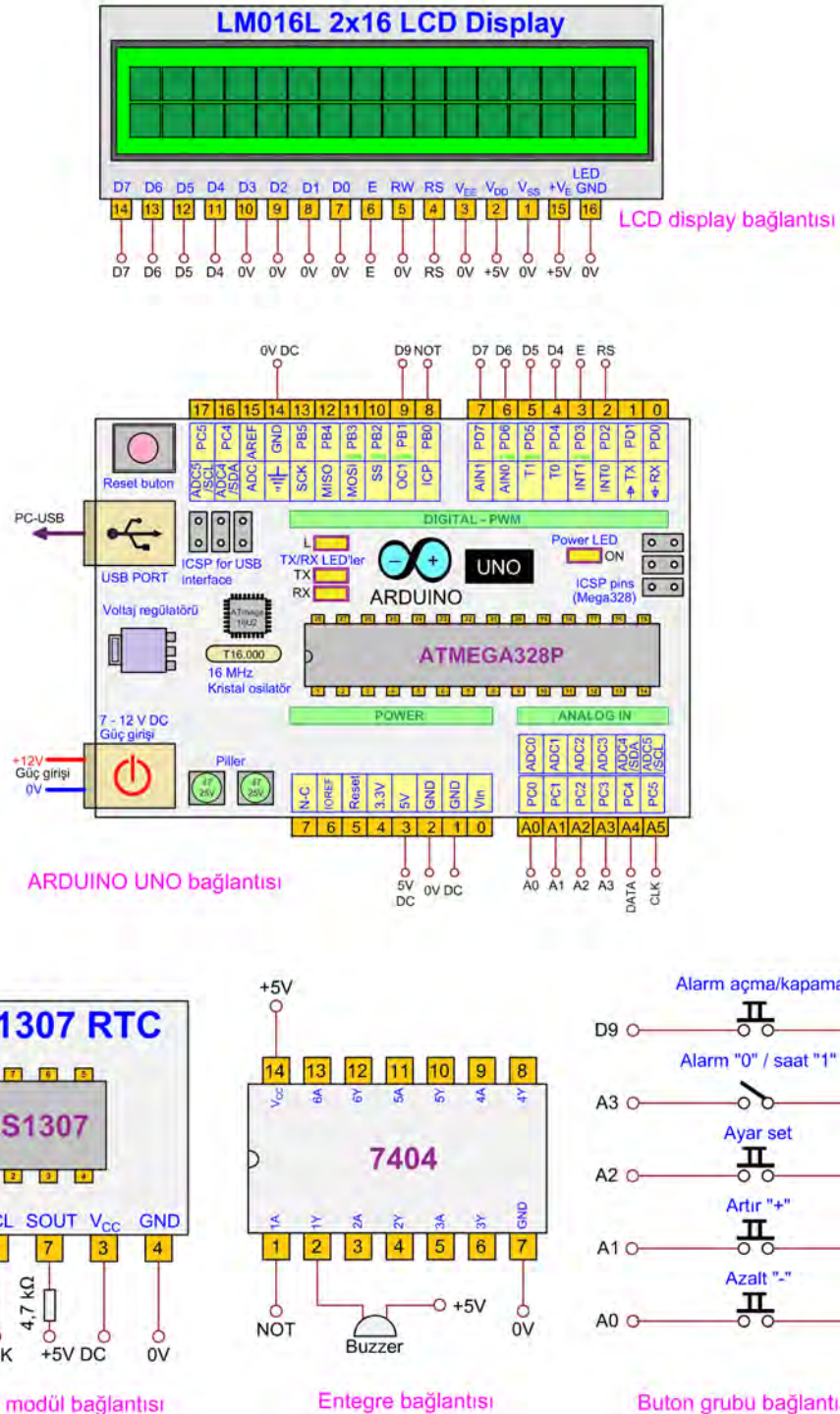
Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



1.10. Dijital Saat Devresi

Devre Şeması: Görsel 1.41’de ARDUINO UNO ile dijital saat devresi, Görsel 1.42’de PIC16F876A entegrasi ile dijital saat devresi görülmektedir.



Görsel 1.41: ARDUINO UNO ile dijital saat devresi

ayarlar menüsü aktif olur. ARTIR ve AZALT butonlarıyla LCD ekrandan alarm bilgileri ayarlanır. ALARM/SAAT anahtarı açık konumda (1) iken ALARMSET butonuna basılırsa saat ayarlar menüsü aktif olur. ARTIR ve AZALT butonlarıyla LCD ekrandan saat bilgileri ayarlanır. Dijital saat devresi ilk kez çalıştırıldığında ALARM AÇMA/KAPAMA butonuna basılırsa alarm test moduna geçer, buzzerden alarm sesi duyulur. Buzzeri durdurmak için ALARM AÇMA/KAPAMA butonuna tekrar basılır (Görsel 1.41-1.42).

Malzemeler: Tablo 1.25'te iki devrenin malzeme listesi görülmektedir.

Tablo 1.25: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodeneleyicili kart	1 adet
PIC16F876A	Mikrodeneleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
CD7404	DEĞİL kapısı	1 adet
LM016	2x16	1 adet
Buzzer	5V	1 adet
Buton	Mikro siviç	4 adet
Anahtar	On/off	1 adet
Direnç	4.7 k Ω	6 adet
Direnç	10 k Ω	3 adet
Kristal osilatör	4 MHz	2 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.26'da iki devrenin kod yazılımları görülmektedir.

Tablo 1.26: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F876A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <Time.h>	2	#include <16f876a.h>
3	#include <DS1307RTC.h>	3	#include <string.h>
4	#include <LiquidCrystal.h>	4	#include <math.h>
5	#include <Wire.h>	5	#include <dss1307.c>
6	LiquidCrystal lcd(2, 3, 4, 5, 6, 7); // LCD module connections (RS, E, D4, D5, D6, D7)	6	#fuses XT, NOWDT, NOPROTECT, NOBROWNOUT, NOLVP, NOPUT, NOWRT, NODEBUG, NOCPD
7	//değişkenler	7	#use delay(clock=4000000)
8	const int azalt = A0;	8	#define use_portb_lcd TRUE
9	const int artir = A1;	9	#include <lcd.c>
10	const int ayar = A2;	10	#use fast_io(a)
11	const int alarm_on = A3;	11	#use fast_io(b)



```

12 const int alarm_out = 8;
13 const int alarm_ac = 9;
14 char Saat[] = "   : : ";
15 char Takvim1[] = "   / /20 ";
16 char Takvim[] = "   / /20 ";
17 char Saat1[] = "   : : ";
18 tmElements_t sr;
19 int menu=0, setalarm=0, alarm_aktif=0, bekle=100;
20 char saat; char dak;
21 char yil ; char ay; char gun;
22 //alt programlar
23 void buzzer()
24 {digitalWrite(alarm_out, HIGH);
25 delay(100);
26 digitalWrite(alarm_out, LOW);}
27 void DS1307_oku()
28 {RTC.read(sr);
29 sr.Year=sr.Year-30;
30 Saat[11]= sr.Second %10 + 48;
31 Saat[10]= sr.Second / 10 + 48;
32 Saat[8]= sr.Minute %10 + 48;
33 Saat[7]= sr.Minute /10 + 48;
34 Saat[5]= sr.Hour %10 + 48;
35 Saat[4]= sr.Hour / 10 + 48;
36 Takvim1[12]= sr.Year % 10 + 48;
37 Takvim1[11]= sr.Year / 10 + 48;
38 Takvim1[7]= sr.Month % 10 + 48;
39 Takvim1[6]= sr.Month / 10 + 48;
40 Takvim1[4]= sr.Day % 10 + 48;
41 Takvim1[3]= sr.Day / 10 + 48;}
42 void LCD_saat_yaz()
43 {lcd.print(Saat);
44 lcd.print(" ");}
45 void LCD_takvim_yaz()
46 {lcd.setCursor(0, 1);
47 lcd.print(Takvim1);}
48 void Lblink(int x, int y)
49 {lcd.setCursor(0, 1);
50 Saat[x]= 32;
51 Saat[y]= 32;
52 LCD_saat_yaz();
53 delay(bekle);}
54 void Tblink(int x, int y)
55 {lcd.setCursor(0, 1);
56 Takvim1[x]= 32;
57 Takvim1[y]= 32;
58 LCD_takvim_yaz();

```

```

12 #use fast_io(c)
13 #define ALARM_OUT pin_C0
14 #define ALARM_AC pin_C1
15 #define ALARM_ON pin_C2
16 #define AYAR pin_C3
17 #define ARTIR pin_C4
18 #define AZALT pin_C5
19 //değişkenler
20 char Saat1[] = "   : : ";
21 char Takvim1[] = "   / /20 ";
22 char Takvim[] = "   / /20 ";
23 char Saat2[] = "   : : ";
24 int menu=0, setalarm=0, alarm_aktif=0, bekle=100;
25 char saat; char dak;
26 char yil ; char ay; char gun;
27 BYTE sec;
28 BYTE dow;
29 BYTE hr;
30 BYTE mth;
31 BYTE year;
32 BYTE min;
33 BYTE day;
34 //alt programlar
35 void saat_oku()
36 {ds1307_get_time(hr,min,sec),
37 ds1307_get_date(day,mth,year,dow),
38 Saat1[11] = sec %10 + 48;
39 Saat1[10] = sec / 10 + 48;
40 Saat1[8] = min %10 + 48;
41 Saat1[7] = min /10 + 48;
42 Saat1[5] = hr %10 + 48;
43 Saat1[4] = hr / 10 + 48;
44 Takvim1[12] = year % 10 + 48;
45 Takvim1[11] = year / 10 + 48;
46 Takvim1[7] = mth % 10 + 48;
47 Takvim1[6] = mth / 10 + 48;
48 Takvim1[4] = day % 10 + 48;
49 Takvim1[3] = day / 10 + 48;}
50 void buzzer()
51 {output_high(alarm_out);
52 delay_ms(100);
53 output_low(alarm_out);}
54 void LCD_saat_yaz()
55 {printf(LCD_PUTC,Saat1);
56 printf(LCD_PUTC," ");}
57 void LCD_takvim_yaz()
58 {lcd_putc("\n");

```

59	delay(bekle);}	59	printf(LCD_PUTC,Takvim1);}
60	void Yblink(int x, int y)	60	void Lblink(int x, int y){
61	{lcd.setCursor(0, 1);	61	lcd_putc("\n");
62	Saat1[x]= 32;	62	Saat1[x]= 32;
63	Saat1[y]= 32;	63	Saat1[y]= 32;
64	lcd.print(Saat1);	64	LCD_saat_yaz();
65	delay(bekle);}	65	delay_ms(bekle);}
66	void Sblink(int x, int y)	66	void Tblink(int x, int y)
67	{lcd.setCursor(0, 1);	67	{lcd_putc("\n");
68	Takvim[x]= 32;	68	Takvim1[x]= 32;
69	Takvim[y]= 32;	69	Takvim1[y]= 32;
70	lcd.print(Takvim);	70	LCD_takvim_yaz();
71	delay(bekle);}	71	delay_ms(bekle);}
72	//genel ayarlar	72	void Yblink(int x, int y)
73	void setup()	73	{lcd_putc("\n");
74	{lcd.begin(16, 2);	74	Saat2[x]= 32;
75	Wire.begin();	75	saat2[y]= 32;
76	pinMode(artir, INPUT_PULLUP);	76	printf(LCD_PUTC,Saat2);
77	pinMode(azalt, INPUT_PULLUP);	77	delay_ms(bekle);}
78	pinMode(ayar, INPUT_PULLUP);	78	void Sblink(int x, int y)
79	pinMode(alarm_on, INPUT_PULLUP);	79	{lcd_putc("\n");
80	pinMode(alarm_ac, INPUT_PULLUP);	80	Takvim[x]= 32;
81	pinMode(alarm_out, OUTPUT);}	81	Takvim[y]= 32;
82	//ana program	82	printf(LCD_PUTC,Takvim);
83	void loop()	83	delay_ms(bekle);}
84	{if(!digitalRead(alarm_ac))	84	//genel ayarlar
85	{alarm_aktif=!alarm_aktif;	85	void main()
86	lcd.clear();	86	{setup_timer_1(T1_DISABLED);
87	if (alarm_aktif==1)	87	setup_timer_2(T2_DISABLED,0,1);
88	{lcd.clear();	88	setup_adc_ports(NO_ANALOGS);
89	lcd.print("ALARM ACIK");}	89	setup_adc(ADC_OFF);
90	else	90	setup_CCP1(CCP_OFF);
91	{lcd.clear();	91	setup_CCP2(CCP_OFF);
92	lcd.print("ALARM KAPALI");}	92	set_tris_a(0x00);
93	while (!digitalRead(alarm_ac));}	93	output_a(0x00);
94	if (saat== sr.Hour && dak==sr.Minute && ay==sr.Month	94	set_tris_b(0x00);
	&& gun==sr.Day && yıl==sr.Year && alarm_aktif==1)	95	output_b(0x00);
95	{buzzer(); delay(100);	96	set_tris_c(0x0FE);
96	buzzer(); delay(150);}	97	output_c(0x00);
97	if(!digitalRead(alarm_on))	98	lcd_init();
98	{setalarm=1;}	99	lcd_putc("\f");
99	else	100	printf(LCD_PUTC, "\f SAAT ALARM UYG.");
100	{setalarm=0;}	101	delay_ms(1000);
101	if(!digitalRead(ayar))	102	//ana program
102	{menu++;	103	while (1)
103	if (menu>=6){menu=0;}	104	{if (!input(ALARM_AC))
104	buzzer();	105	{alarm_aktif=!alarm_aktif;



105	while(!digitalRead(ayar));	106	if (alarm_aktif==1)
106	digitalWrite(alarm_out, LOW);}	107	{lcd_putc("\f");
107	switch (menu)	108	printf(LCD_PUTC,"ALARM ACIK");}
108	{case 0:	109	else
109	lcd.clear();	110	{lcd_putc("\f");
110	DS1307_oku();	111	printf(LCD_PUTC,"ALARM KAPALI");}
111	if (setalarm==0)	112	while (!input(ALARM_AC));}
112	{saat=sr.Hour ; dak=sr.Minute;	113	if (saat== hr && dak==min && ay==mth && gun==day
113	yil=sr.Year ; ay=sr.Month; gun=sr.Day;}		&& yil==year && alarm_aktif==1)
114	lcd.setCursor(0, 0);	114	{buzzer();
115	LCD_saat_yaz();	115	delay_ms(100);
116	LCD_takvim_yaz();	116	buzzer();
117	delay(500);	117	delay_ms(150);}
118	break;	118	if(!input(ALARM_ON))
119	case 1:	119	{setalarm=1;}
120	lcd.clear();	120	else
121	while(digitalRead(ayar))	121	{setalarm=0;}
122	{if (setalarm==1)	122	if(input(AYAR)==0)
123	{Saat1[6]= saat / 10+48; Saat1[7]= saat %10+48;	123	{menu++;
	Saat1[9]= dak / 10+48; Saat1[10]= dak%10+48;	124	if (menu>=6){menu=0;}
124	lcd.print("SAAT ALARM AYARI");	125	buzzer();
125	lcd.setCursor(0, 1);	126	while(!input(AYAR));
126	lcd.print(Saat1);	127	output_low(alarm_out);}
127	delay(bekle);	128	switch (menu)
128	Yblink(6,7);	129	{case 0:
129	if(!digitalRead(artir))	130	lcd_putc("\f");
130	{buzzer();	131	saat_oku();
131	saat++;	132	if (setalarm==0)
132	if (saat>=24)	133	{saat=hr ; dak=min;
133	{saat=0;}	134	yil=year ; ay=mth; gun=day ;}
134	while(!digitalRead(artir));}	135	LCD_saat_yaz();
135	if(!digitalRead(azalt)&& (saat>0))	136	LCD_takvim_yaz();
136	{buzzer();	137	delay_ms(500);
137	saat--;	138	break;
138	if (saat==0)	139	case 1:
139	{saat=0;}	140	lcd_putc("\f");
140	while(!digitalRead(azalt));}	141	while(input(ayar))
141	else	142	{if (setalarm==1)
142	{DS1307_oku();	143	{Saat2[6]= saat / 10+48; Saat2[7]= saat %10+48;
143	lcd.setCursor(0, 0);		Saat2[9]= dak / 10+48; Saat2[10]= dak%10+48;
144	lcd.print("SAAT AYARI");	144	printf(LCD_PUTC, "SAAT ALARM AYARI");
145	lcd.setCursor(0,1);	145	lcd_putc("\n");
146	DS1307_oku();	146	printf(LCD_PUTC,Saat2);
147	LCD_saat_yaz();	147	delay_ms(bekle);
148	delay(bekle);	148	Yblink(6,7);
149	Lblink(4,5);	149	if(!input(artir))
150	if(!digitalRead(artir))	150	{buzzer();

```

151 {buzzer();
152 sr.Hour++;
153 if (sr.Hour>=24)
154 {sr.Hour=0;}
155 sr.Year=sr.Year+30;
156 RTC.write(sr);
157 while(!digitalRead(artir));}
158 if(!digitalRead(azalt)&& (sr.Hour>0))
159 {buzzer();
160 sr.Hour--;
161 if (sr.Hour==0)
162 {sr.Hour=0;}
163 sr.Year=sr.Year+30;
164 RTC.write(sr);
165 while(!digitalRead(azalt));}}
166 break;
167 case 2:
168 lcd.clear();
169 while(digitalRead(ayar))
170 {if (setalarm==1)
171 {Saat1[6]= saat / 10+48; Saat1[7]= saat %10+48;
172 Saat1[9]= dak / 10+48; Saat1[10]= dak%10+48;
173 lcd.print("DAK ALARM AYARI");
174 lcd.setCursor(0, 1);
175 lcd.print(Saat1);
176 delay(bekle);
177 Yblink(9,10);
178 if(!digitalRead(artir))
179 {buzzer();
180 dak++;
181 if (dak>59)
182 {dak=0;}
183 while(!digitalRead(artir));}
184 if(!digitalRead(azalt)&& (dak>0))
185 {buzzer();
186 dak--;
187 if (dak==0)
188 {dak=0;}
189 while(!digitalRead(azalt));}
190 else
191 {DS1307_oku();
192 lcd.setCursor(0, 0);
193 lcd.print("DAKIKAYARI");
194 lcd.setCursor(0, 1);
195 DS1307_oku();
196 LCD_saat_yaz();
197 delay(bekle);

```

```

151 saat++;
152 if (saat>=24)
153 {saat=0;}
154 while(!input(artir));}
155 if(!input(azalt)&& (saat>0))
156 {buzzer();
157 saat--;
158 if (saat==0)
159 {saat=0;}
160 while(!input(azalt));}
161 else
162 {Saat_oku();
163 lcd_putc("\f");
164 printf(LCD_PUTC, " SAAT AYARI ");
165 lcd_putc("\n");
166 LCD_saat_yaz();
167 delay_ms(bekle);
168 Lblink(4,5);
169 if(!input(artir))
170 {buzzer();
171 hr++;
172 if (hr>=24)
173 {hr=0;}
174 ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
175 while(!input(artir));}
176 if(!input(azalt)&& (hr>0))
177 {buzzer();
178 hr--;
179 if (hr==0)
180 {hr=0;}
181 ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
182 while(!input(azalt));}
183 case 2:
184 lcd_putc("\f");
185 while(input(ayar))
186 {if (setalarm==1)
187 {Saat2[6]= saat / 10+48; Saat2[7]= saat %10+48;
188 Saat2[9]= dak / 10+48; Saat2[10]= dak%10+48;
189 printf(LCD_PUTC, "DAK ALARM AYARI");
190 lcd_putc("\n");
191 printf(LCD_PUTC,Saat2);
192 delay_ms(bekle);
193 Yblink(9,10);
194 if(!input(artir))
195 {buzzer();
196 dak++;
197 if (dak>59)

```



197	Lblink(7,8);	197	{dak=0;}
198	if(!digitalRead(artir))	198	while(!input(artir));}
199	{buzzer();	199	if(!input(azalt)&& (dak>0))
200	sr.Minute++;	200	{buzzer();
201	if (sr.Minute>=59)	201	dak--;
202	{sr.Minute=0;}	202	if (dak==0)
203	sr.Year=sr.Year+30;	203	{dak=0;}
204	RTC.write(sr);	204	while(!input(azalt));}
205	while(!digitalRead(artir));}	205	else
206	if(!digitalRead(azalt)&& (sr.Minute>0)){	206	{Saat_oku();
207	buzzer();	207	lcd_putc("\f");
208	sr.Minute--;	208	printf(LCD_PUTC, " DAKIKA AYARI ");
209	if (sr.Minute==0)	209	lcd_putc("\n");
210	{sr.Minute=0;}	210	LCD_saat_yaz();
211	sr.Year=sr.Year+30;	211	delay_ms(bekle);
212	RTC.write(sr);	212	Lblink(7,8);
213	while(!digitalRead(azalt));}}	213	if(!input(artir))
214	break;	214	{buzzer();
215	case 3:	215	min++;
216	lcd.clear();	216	if (min>59)
217	while(digitalRead(ayar))	217	{min=0;}
218	{if (setalarm==1)	218	ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
219	{Takvim[12] = yil % 10 + 48; Takvim[11] = yil / 10 + 48;	219	while(!input(artir));}
	Takvim[7] = ay % 10 + 48; Takvim[6] = ay / 10 + 48;	220	if(!input(azalt)&& (min>0))
	Takvim[4] = gun % 10 + 48; Takvim[3] = gun / 10 + 48;	221	{buzzer();
220	lcd.print("YIL ALARM AYARI");	222	min--;
221	lcd.setCursor(0, 1);	223	if (min==0)
222	lcd.print(Takvim);	224	{min=0;}
223	delay(bekle);	225	ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
224	Sblink(12,11);	226	while(!input(azalt));}}
225	if(!digitalRead(artir))	227	case 3:
226	{buzzer();	228	lcd_putc("\f");
227	yil++;	229	while(input(ayar))
228	if (yil>99)	230	{if (setalarm==1)
229	{yil=0;}	231	{Takvim[12] = yil % 10 + 48; Takvim[11] = yil / 10 + 48;
230	while(!digitalRead(artir));}		Takvim[7] = ay % 10 + 48; Takvim[6] = ay / 10 + 48;
231	if(!digitalRead(azalt)&& (yil>0))		Takvim[4] = gun % 10 + 48; Takvim[3] = gun / 10 + 48;
232	{buzzer();	232	printf(LCD_PUTC, "YIL ALARM AYARI");
233	yil--;	233	lcd_putc("\n");
234	if (yil==0)	234	printf(LCD_PUTC,Takvim);
235	{yil=0;}	235	delay_ms(bekle);
236	while(!digitalRead(azalt));}}	236	Sblink(12,11);
237	else	237	if(!input(artir))
238	{DS1307_oku();	238	{buzzer();
239	lcd.setCursor(0, 0);	239	yil++;
240	lcd.print(" YIL AYARI ");	240	if (saat>99)
241	lcd.setCursor(2, 1);	241	{yil=0;}

242	DS1307_oku();	242	while(!input(artir));}
243	LCD_takvim_yaz();	243	if(!input(azalt)&& (yil>0))
244	delay(bekle);	244	{buzzer();
245	Tblink(11,12);	245	yil--;
246	if(!digitalRead(artir))	246	if (yil==0)
247	{buzzer();	247	{yil=0;}
248	sr.Year++;	248	while(!input(azalt));}
249	if (sr.Year>99)	249	else
250	{sr.Year=0;}	250	{Saat_oku();
251	sr.Year=sr.Year+30;	251	lcd_putc("\f");
252	RTC.write(sr);	252	printf(LCD_PUTC, " YIL AYARI ");
253	while(!digitalRead(artir));}	253	lcd_putc("\n");
254	if(!digitalRead(azalt)&& (sr.Year>0))	254	LCD_takvim_yaz();
255	{buzzer();	255	delay_ms(bekle);
256	sr.Year--;	256	Tblink(11,12); //.....
257	if (sr.Year==0)	257	if(!input(artir))
258	{sr.Year=0;}	258	{buzzer();
259	sr.Year=sr.Year+30;	259	year++;
260	RTC.write(sr);	260	if (year>99)
261	while(!digitalRead(azalt));}}	261	{year=0;}
262	break;	262	ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
263	case 4:	263	while(!input(artir));}
264	lcd.clear();	264	if(!input(azalt)&& (year>0))
265	while(digitalRead(ayar))	265	{buzzer();
266	{if (setalarm==1)	266	year--;
267	{Takvim[12] = yil % 10 + 48; Takvim[11] = yil / 10 + 48;	267	if (year==0)
	Takvim[7] = ay % 10 + 48; Takvim[6] = ay / 10 + 48;	268	{year=0;}
	Takvim[4] = gun % 10 + 48; Takvim[3] = gun / 10 + 48;	269	ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
268	lcd.print(" AY ALARM AYARI ");	270	while(!input(azalt));}}
269	lcd.setCursor(0, 1);	271	case 4:
270	lcd.print(Takvim);	272	lcd_putc("\f");
271	delay(bekle);	273	while(input(ayar))
272	Sblink(7,6);	274	{if (setalarm==1)
273	if(!digitalRead(artir))	275	{Takvim[12] = yil % 10 + 48; Takvim[11] = yil / 10 + 48;
274	{buzzer();		Takvim[7] = ay % 10 + 48; Takvim[6] = ay / 10 + 48;
275	ay++;		Takvim[4] = gun % 10 + 48; Takvim[3] = gun / 10 + 48;
276	if (ay>12)	276	printf(LCD_PUTC, "AY ALARM AYARI");
277	{ay=1;}	277	lcd_putc("\n");
278	while(!digitalRead(artir));}	278	printf(LCD_PUTC,Takvim);
279	if(!digitalRead(azalt)&& (ay>1))	279	delay_ms(bekle);
280	{buzzer();	280	Sblink(7,6);
281	ay--;	281	if(!input(artir))
282	if (ay==1)	282	{buzzer();
283	{ay=1;}	283	ay++;
284	while(!digitalRead(azalt));}	284	if (ay>12)
285	else	285	{ay=1;}
286	{DS1307_oku();	286	while(!input(artir));}



```

287 lcd.setCursor(0, 0);
288 lcd.print(" AY AYARI ");
289 lcd.setCursor(2, 1);
290 DS1307_oku();
291 LCD_takvim_yaz();
292 delay(bekle);
293 Tblink(6,7);
294 if(!digitalRead(artir))
295 {buzzer();
296 sr.Month++;
297 if (sr.Month>12)
298 {sr.Month=1;}
299 sr.Year=sr.Year+30;
300 RTC.write(sr);
301 while(!digitalRead(artir));}
302 if(!digitalRead(azalt)&& (sr.Month>1))
303 {buzzer();
304 sr.Month--;
305 if (sr.Month==1)
306 {sr.Month=1;}
307 sr.Year=sr.Year+30;
308 RTC.write(sr);
309 while(!digitalRead(azalt));}}
310 break;
311 case 5:
312 lcd.clear();
313 while(digitalRead(ayar))
314 {if (setalarm==1)
315 {Takvim[12] = yil % 10 + 48; Takvim[11] = yil / 10 + 48;
Takvim[7] = ay % 10 + 48; Takvim[6] = ay / 10 + 48;
Takvim[4] = gun % 10 + 48; Takvim[3] = gun / 10 + 48;
316 lcd.print(" GUN ALARM AYARI ");
317 lcd.setCursor(0, 1);
318 lcd.print(Takvim);
319 delay(bekle);
320 Sblink(4,3);
321 if(!digitalRead(artir))
322 {buzzer();
323 gun++;
324 if (gun>28 && ay==2 && yil%4!=0) {gun=1;}
325 if (gun>29 && ay==2 && yil%4==0) {gun=1;}
326 if ((gun>30) && (ay==4 || ay==6 || ay==9 || ay==11 ))
{gun=1;}
328 if (gun>31)
329 {gun=1;}
330 while(!digitalRead(artir));}
331 if(!digitalRead(azalt)&& (gun>1))

```

```

287 if(!input(azalt)&& (ay>1))
288 {buzzer();
289 ay--;
290 if (ay==1)
291 {ay=1;}
292 while(!input(azalt));}}
293 else
294 {Saat_oku();
295 lcd_putc("\f");
296 printf(LCD_PUTC, " AY AYARI ");
297 lcd_putc("\n");
298 LCD_takvim_yaz();
299 delay_ms(bekle);
300 Tblink(6,7);
301 if(!input(artir))
302 {buzzer();
303 mth++;
304 if (mth>12)
305 {mth=1;}
306 ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
307 while(!input(artir));}
308 if(!input(azalt)&& (mth>1))
309 {buzzer();
310 mth--;
311 if (mth==1)
312 {mth=1;}
313 ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
314 while(!input(azalt));}}
315 case 5:
316 lcd_putc("\f");
317 while(input(ayar))
318 {if (setalarm==1)
319 {Takvim[12] = yil % 10 + 48; Takvim[11] = yil / 10 + 48;
Takvim[7] = ay % 10 + 48; Takvim[6] = ay / 10 + 48;
Takvim[4] = gun % 10 + 48; Takvim[3] = gun / 10 + 48;
320 printf(LCD_PUTC, "GUN ALARM AYARI");
321 lcd_putc("\n");
322 printf(LCD_PUTC,Takvim);
323 delay_ms(bekle);
324 Sblink(4,3);
325 if(!input(artir))
326 {buzzer();
327 gun++;
328 if (gun>28 && ay==2 && yil%4!=0) {gun=1;}
329 if (gun>29 && ay==2 && yil%4==0) {gun=1;}
330 if ((gun>30) && (ay==4 || ay==6 || ay==9 || ay==11 )) {gun=1;}
331 if (gun>31)

```

```

332 {buzzer();
333 gun--;
334 if (gun==1)
335 {gun=1;}
336 while(!digitalRead(azalt));}
337 else
338 {DS1307_oku();
339 lcd.setCursor(0, 0);
340 lcd.print("GUN AYARI ");
341 lcd.setCursor(2, 1);
342 DS1307_oku();
343 LCD_takvim_yaz();
344 delay(bekle);
345 Tblink(3,4);
346 if(!digitalRead(artir))
347 {buzzer();
348 sr.Day++;
349 if (sr.Day>28 && sr.Month==2 && sr.Year%4!=0) {sr.Day=1;}
350 if (sr.Day>29 && sr.Month==2 && sr.Year%4==0) {sr.Day=1;}
351 if ((sr.Day>30) && (sr.Month==4 || sr.Month==6
|| sr.Month==9 || sr.Month==11)) {sr.Day=1;}
352 if (sr.Day>31)
353 {sr.Day=1;}
354 sr.Year=sr.Year+30;
355 RTC.write(sr);
356 while(!digitalRead(artir));}
357 if(!digitalRead(azalt)&& (sr.Day>1))
358 {buzzer();
359 sr.Day--;
360 if (sr.Day==1)
361 {sr.Day=1;}
362 sr.Year=sr.Year+30;
363 RTC.write(sr);
364 while(!digitalRead(azalt));}}
365 break;
366 default:
367 break;}}
368 {buzzer();

```

```

332 {gun=1;}
333 while(!input(artir));}
334 if(!input(azalt)&& (gun>1))
335 {buzzer();
336 gun--;
337 if (gun==1)
338 {gun=1;}
339 while(!input(azalt));}}
340 else
341 {Saat_oku();
342 lcd_putc("\f");
343 printf(LCD_PUTC, " GUN AYARI ");
344 lcd_putc("\n");
345 LCD_takvim_yaz();
346 delay_ms(bekle);
347 Tblink(3,4);
348 if(!input(artir))
349 {buzzer();
350 day++;
351 if (day>28 && mth==2 && year%4!=0) {day=1;}
352 if (day>29 && mth==2 && year%4==0) {day=1;}
353 if ((day>30) && (mth==4 || mth==6 || mth==9 || mth==11))
{day=1;}
354 if (day>31)
355 {day=1;}
356 ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
357 while(!input(artir));}
358 if(!input(azalt)&& (day>1))
359 {buzzer();
360 day--;
361 if (day==1)
362 {day=1;}
363 ds1307_set_date_time(day,mth,year,dow,hr,min,sec);
364 while(!input(azalt));}}
365 break;
366 default:
367 break;}}

```



İşlem Basamakları

1. Uygun tipli mikrodeneleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
6. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodeneleyiciye yükleyiniz. Devreyi çalıştırınız. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

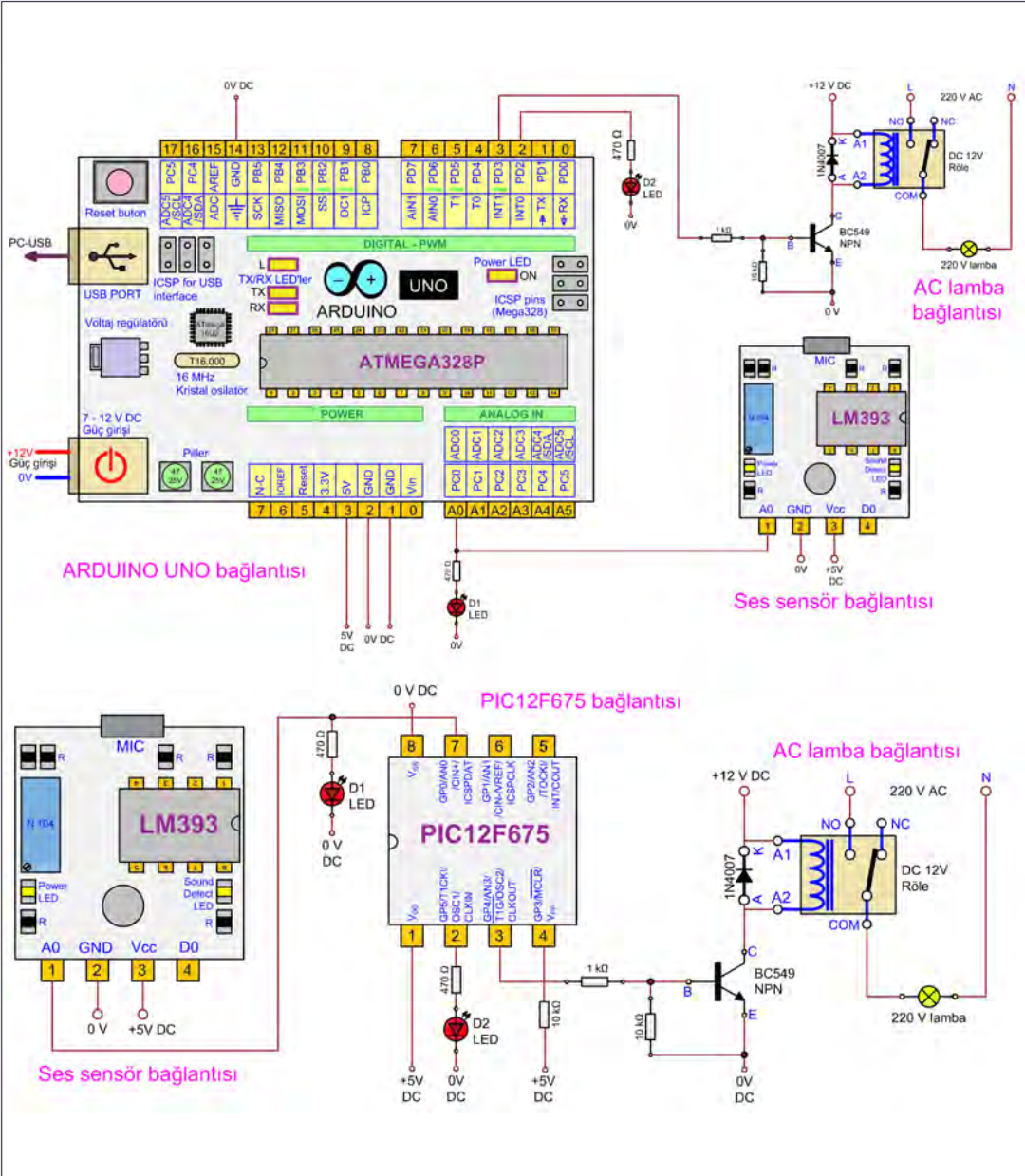
1. DS1307 RTC modülünün görevi nedir?
2. Programda onluk sayının ASCII kodunu üreten komut (Takvim[11]=yıl/10+48;) kullanılmasının sebebi nedir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

1.11. Alkışla Yanan Lamba Devresi

Devre Şeması: Görsel 1.43'te alkışla yanan lamba devresi görülmektedir.



Görsel 1.43: Alkışla yanan lamba devresi

Devre Hakkında: Alkışla yanan lamba devresi; ARDUINO ses sensör modülü, 12V DC röle ve röle sürücü devresi kullanılarak tasarlanmıştır. ARDUINO ses sensöründen gelen analog veri mikrodnetleyici tarafından işlenir. Analog veri genliğine göre röle iletime geçirilir veya durdurulur. Devreye enerji uygulandığında röle kapalı konumdadır ve lamba yanar. Timer kesmesi devreye alınır. Kesme sayıcısı süresi dolduğu, devrenin çalışmakta olduğu LED diyodun (sistem etiketli LED) yanıp sönmesi ile gösterilir. Ses sensörü çevreden gelen sesin şiddetine göre analog sinyal üretir. Alkış sesi algılandığında ses sensöründen yüksek genlikli analog sinyal üretilir. Mikrodnetleyici, sesin şiddetine ve devamlılığına göre işlem yapar. Gelen analog sinyal alkış olarak algılanırsa röleye enerji uygulanır veya rölenin enerjisi kesilir. Böylece alkış sesine göre lamba yanar veya söner (Görsel 1.43).



Malzemeler: Tablo 1.27’de iki devrenin malzeme listesi görülmektedir.

Tablo 1.27: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyici kart	1 adet
PIC12F675	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
Darbe sensörü	ARDUINO	1 adet
LED diyot	Kırmızı	2 adet
Direnç	470 Ω	2 adet
Direnç	10 k Ω	2 adet
Direnç	1 k Ω	1 adet
Röle	12V	1 adet
Transistör	BC549 (NPN)	1 adet
Lamba	220 V AC	1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.28’de iki devrenin kod yazılımları görülmektedir.

Tablo 1.28: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC12F675)
1	//değişkenler	1	//kütüphaneler ve tanımlamalar
2	const int role = 3;	2	#include <12F675.h>
3	const int sistem_calisiyor = 2;	3	#device adc=8
4	long alkis_var;	4	#fuses INTRC_IO, NOWDT, NOPUT, NOPROTECT, NOCPD, MCLR
5	int i=0;	5	#byte OSCCAL = 0x90
6	//alt programlar	6	#use delay(clock=4000000)
7	void setupTimer1()	7	#define GP0 PIN_A0
8	{noInterrupts();	8	#define role PIN_A4
9	TCCR1A= 0;	9	#define sistem_calisiyor PIN_A5
10	TCCR1B= 0;	10	//değişkenler
11	TCNT1= 0;	11	int i=0;
12	//0.5 Hz (16000000/((15624+1)*512))	12	int16 alkis_var;
13	OCR1A = 7812;	13	//alt programlar
14	TCCR1B = (1 << WGM12);	14	#int_timer0
15	TCCR1B = (1 << CS12) (1 << CS10);	15	void timer0_kesme ()
16	TIMSK1 = (1 << OCIE1A);	16	{set_timer0(60);
17	interrupts();}	17	i++;
18	ISR(TIMER1_COMPA_vect)	18	if (i==10)
19	{noInterrupts();		

```

20 digitalWrite(2, !digitalRead(2));
21 interrupts();}
22 //genel ayarlar
23 void setup()
24 {pinMode(role, OUTPUT);
25 pinMode(sistem_calisiyor, OUTPUT);
26 setupTimer1();}
27 //ana program
28 void loop()
29 {alkis_var= analogRead(A0);
30 while(alkis_var>128)
31 {alkis_var= analogRead(A0);
32 if (alkis_var > 128)
33 {digitalWrite(3, !digitalRead(3));
34 while(alkis_var>128)
35 {alkis_var= analogRead(A0);}}}}

```

```

19 output_high(sistem_calisiyor);
20 if (i==20)
21 {output_low(sistem_calisiyor);
22 i=0;}}
23 //genel ayarlar
24 void main()
25 {OSCCAL= 0x3f;
26 setup_adc(ADC_CLOCK_INTERNAL);
27 setup_adc_ports(sAN0|sAN1|sAN2|VSS_VDD);
28 setup_adc(adc_clock_div_32);
29 output_bit(sistem_calisiyor, 0);
30 output_bit(role,0);
31 delay_ms(2000);
32 setup_timer_0(RTCC_INTERNAL | RTCC_DIV_256);
33 set_timer0(60);
34 enable_interrupts(INT_timer0);
35 enable_interrupts(GLOBAL);
36 //ana program
37 while(true)
38 {set_adc_channel(0);
39 delay_us(100);
40 alkis_var = read_adc();
41 delay_ms(10);
42 while(alkis_var>128)
43 {alkis_var = read_adc();
44 if (alkis_var > 128)
45 {output_toggle(role);
46 while(alkis_var>128)
47 {set_adc_channel(0);
48 delay_us(100);
49 alkis_var = read_adc();}}}}

```



İşlem Basamakları

1. Uygun tipli mikrodenetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
6. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodenetleyiciye yükleyiniz. Devreyi çalıştırınız. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Lambayı yakacak veya söndürecek alkış sesinin şiddetini ayarlamak için program üzerinde yapılacak değişiklikler nelerdir?
2. Lambayı yakacak veya söndürecek alkış sesinin şiddetini ayarlamak için devre üzerinde yapılacak değişiklikler nelerdir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

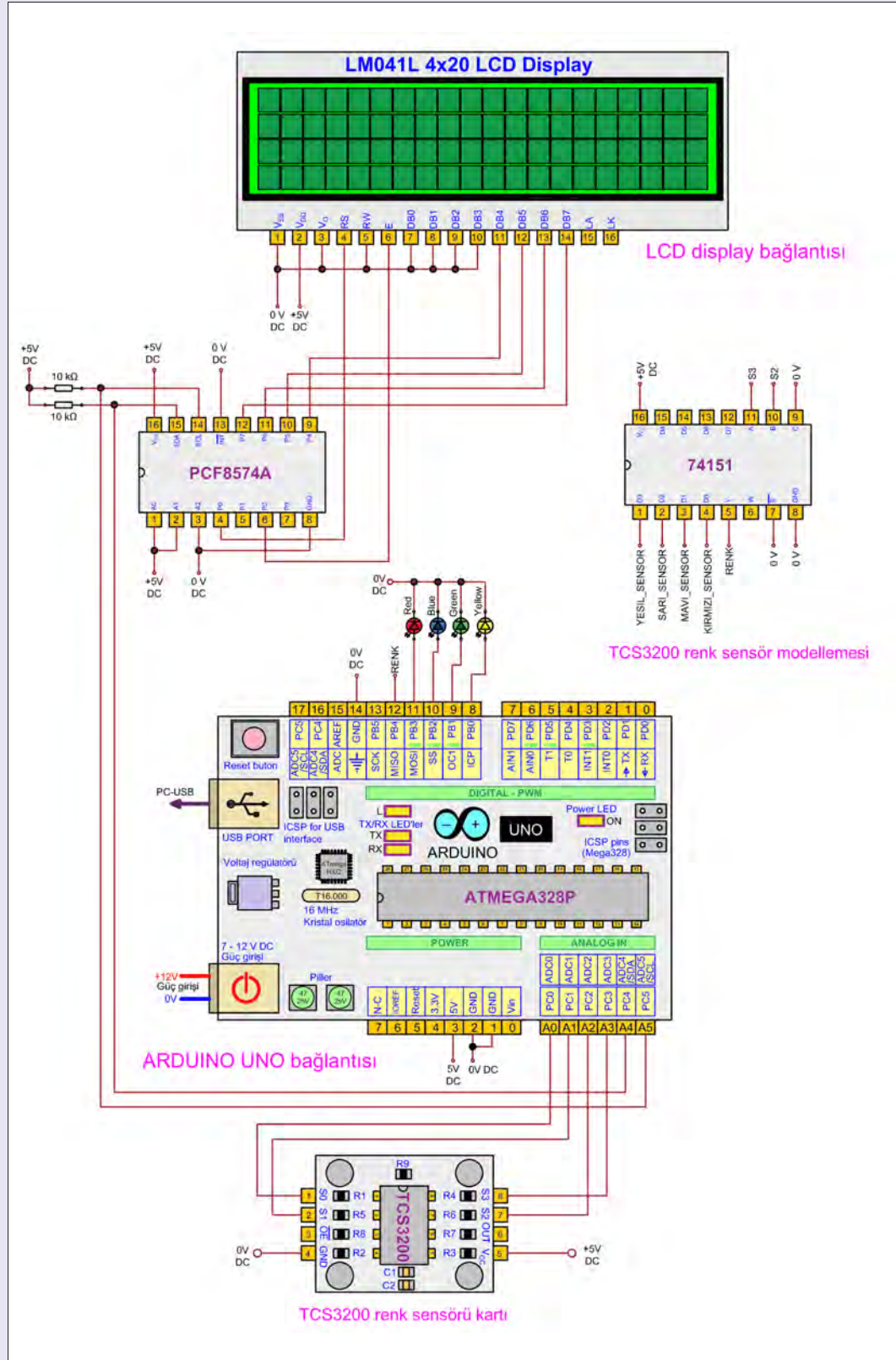
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



1.12. Renk Algılama Devresi



Devre Şeması: Görşel 1.44'te ARDUINO UNO ile renk algılama devresi, Görşel 1.45'te PIC16F873A ile renk algılama devresi görölmektedir.



Görşel 1.44: ARDUINO UNO ile renk algılama devresi

Malzemeler: Tablo 1.29’da iki devrenin malzeme listesi görülmektedir.

Tablo 1.29: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F873A	Mikrodenetleyici	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
LM041	4x20 LCD ekran	1 adet
PCF8574A	I2C seri-paralel kaymalı kaydedici	1 adet
TCS3200	Renk sensörü	1 adet
74151	1-8 multipleks	1 adet
LED diyot	Kırmızı, mavi, yeşil ve sarı	4 adet
Direnç	470 Ω	4 adet
Direnç	10 k Ω	3 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.30’da iki devrenin kod yazılımları görülmektedir.

Tablo 1.30: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F873A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <LiquidCrystal_I2C.h>	2	#include <16F873A.h>
3	LiquidCrystal_I2C lcd(0x23,16,4);	3	#fuses XT, NOWDT, PUT, NOPROTECT, BROWNOUT, NOLVP
4	#define S0 A0	4	#use delay(clock=4000000)
5	#define S1 A1	5	#define PCF_SDA PIN_C0
6	#define S2 A2	6	#define PCF_SCL PIN_C1
7	#define S3 A3	7	#use i2c(master, sda=PCF_SDA, scl=PCF_SCL)
8	#define renk 12	8	#define PCF8574_ID 3
9	#define k_LED 11	9	#define PCF_ADD 64
10	#define y_LED 10	10	#define RS P0
11	#define m_LED 9	11	#define RW P1
12	#define d_LED 8	12	#define E P2
13	//değişkenler	13	#define DB4 P4
14	int kırmızı_renk = 0;	14	#define DB5 P5
15	int yeşil_renk = 0;	15	#define DB6 P6
16	int mavi_renk = 0;	16	#define DB7 P7
17	int diger_renk = 0;	17	#define S0 PIN_A0
18	//alt programlar	18	#define S1 PIN_A1



```

19 void renk_oku()
20 {lcd.clear();
21 digitalWrite(S2, LOW);
22 digitalWrite(S3, LOW);
23 kirmizi_renk = pulseIn(renk, digitalRead(renk) == HIGH ?
    LOW : HIGH);
24 lcd.print("KIRMIZI RENK ");
25 lcd.print(kirmizi_renk);
26 delay(500);
27 digitalWrite(S2, LOW);
28 digitalWrite(S3, HIGH);
29 mavi_renk = pulseIn(renk, digitalRead(renk) == HIGH ?
    LOW : HIGH);
30 lcd.setCursor(0,1);
31 lcd.print("MAVI RENK ");
32 lcd.print(mavi_renk);
33 delay(500);
34 digitalWrite(S2, HIGH);
35 digitalWrite(S3, HIGH);
36 yesil_renk = pulseIn(renk, digitalRead(renk) == HIGH ?
    LOW : HIGH);
37 lcd.setCursor(-4,2);
38 lcd.print("YESIL RENK ");
39 lcd.print(yesil_renk);
40 delay(500);
41 digitalWrite(S2, HIGH);
42 digitalWrite(S3, LOW);
43 diger_renk = pulseIn(renk, digitalRead(renk) == HIGH ?
    LOW : HIGH);
44 lcd.setCursor(-4,3);
45 lcd.print("DIGER RENK ");
46 lcd.print(diger_renk);
47 delay(1000);}
48 //genel ayarlar
49 void setup()
50 {lcd.begin();
51 pinMode(S0, OUTPUT);
52 pinMode(S1, OUTPUT);
53 pinMode(S2, OUTPUT);
54 pinMode(S3, OUTPUT);
55 pinMode(12, INPUT);
56 pinMode(11, OUTPUT);
57 pinMode(10, OUTPUT);
58 pinMode(9, OUTPUT);
59 pinMode(8, OUTPUT);
60 lcd.clear();
61 lcd.setCursor(0,0);

```

```

19 #define S2 PIN_A2
20 #define S3 PIN_A3
21 #define renk PIN_B0
22 #define k_LED PIN_B1
23 #define y_LED PIN_B2
24 #define m_LED PIN_B3
25 #define d_LED PIN_B4
26 #include <yt_lcd.c>
27 //değişkenler
28 int16 kirmizi_renk = 0;
29 int16 yesil_renk = 0;
30 int16 mavi_renk = 0;
31 int16 diger_renk = 0;
32 //alt programlar
33 void sure_oku()
34 {while(input(renk));
35 while(!input(renk));
36 while(input(renk));
37 set_timer1(0);
38 while(!input(renk));}
39 void renk_oku()
40 {lcd_putc("\f");
41 output_low(S2);
42 output_low(S3);
43 sure_oku();
44 kirmizi_renk = get_timer1();
45 printf(lcd_putc, "KIRMIZI RENK %ld", kirmizi_renk);
46 delay_ms(500);
47 output_low(S2);
48 output_high(S3);
49 sure_oku();
50 mavi_renk = get_timer1();
51 printf(lcd_putc, "\nMAVI RENK %ld", mavi_renk);
52 delay_ms(500);
53 output_high(S2);
54 output_high(S3);
55 sure_oku();
56 yesil_renk = get_timer1();
57 lcd_gotoxy(-3,3);
58 printf(lcd_putc, "\YESIL RENK %ld", yesil_renk);
59 delay_ms(500);
60 output_high(S2);
61 output_low(S3);
62 sure_oku();
63 diger_renk = get_timer1();
64 lcd_gotoxy(-3,4);
65 printf(lcd_putc, "\DIGER RENK %ld", diger_renk);

```

62	lcd.print("RENK SENSORU");	66	delay_ms(1000);}
63	lcd.setCursor(0,1);	67	//genel ayarlar
64	lcd.print("UYGULAMASI");	68	void main ()
65	lcd.setCursor(-4,2);	69	{setup_psp(PSP_DISABLED);
66	lcd.print("ILERI MIKRO");	70	setup_timer_1(T1_INTERNAL T1_DIV_BY_4);
67	lcd.setCursor(-4,3);	71	setup_timer_2(T2_DISABLED,0,1);
68	lcd.print("UYGULAMA KITABI");	72	setup_adc_ports(NO_ANALOGS);
69	digitalWrite(S0, HIGH);	73	setup_adc(ADC_OFF);
70	digitalWrite(S1, HIGH);	74	set_tris_a(0x00);
71	delay(2000);}	75	set_tris_d(0x01);
72	//ana program	76	set_tris_c(0x00);
73	void loop()	77	lcd_init();
74	{lcd.clear();	78	lcd_send_byte(0, 0x0C);
75	renk_oku();	79	lcd_putc("\fRENK SENSORU");
76	if (kirmizi_renk < mavi_renk && kirmizi_renk <	80	lcd_putc("\nUYGULAMASI");
	yesil_renk && kirmizi_renk < diger_renk)	81	lcd_gotoxy(-3,3);
77	{lcd.clear();	82	lcd_putc("ILERI MIKRO");
78	lcd.setCursor(0,0);	83	lcd_gotoxy(-3,4);
79	lcd.print("KIRMIZI LED ");	84	lcd_putc("UYGULAMA KITABI");
80	lcd.setCursor(0,1);	85	output_high(S0);
81	lcd.print("AKTIF ");	86	output_high(S1);
82	digitalWrite(k_LED, HIGH);	87	delay_ms(500);
83	digitalWrite(y_LED, LOW);	88	//ana program
84	digitalWrite(m_LED, LOW);	89	while(1)
85	digitalWrite(d_LED, LOW);	89	{lcd_putc("\f");
86	delay(1000);}	90	renk_oku();
87	else if (mavi_renk < kirmizi_renk && mavi_renk <	91	if (kirmizi_renk < mavi_renk && kirmizi_renk <
	yesil_renk && mavi_renk < diger_renk)		yesil_renk && kirmizi_renk < diger_renk)
88	{lcd.clear();	92	{lcd_putc("\f");
89	lcd.setCursor(0,0);	93	lcd_putc("KIRMIZI LED ");
90	lcd.print("MAVI LED ");	94	lcd_putc("\n AKTIF ");
91	lcd.setCursor(0,1);	95	output_high(k_LED);
92	lcd.print("AKTIF ");	96	output_low(y_LED);
93	digitalWrite(m_LED, HIGH);	97	output_low(m_LED);
94	digitalWrite(k_LED, LOW);	98	output_low(d_LED);}
95	digitalWrite(y_LED, LOW);	99	else if (mavi_renk < kirmizi_renk && mavi_renk <
96	digitalWrite(d_LED, LOW);}		yesil_renk && mavi_renk < diger_renk)
97	else if (yesil_renk < kirmizi_renk && yesil_renk <	100	{lcd_putc("\f");
	mavi_renk && yesil_renk < diger_renk)	101	lcd_putc("MAVI LED ");
98	{lcd.clear();	102	lcd_putc("\nAKTIF ");
99	lcd.setCursor(0,0);	103	output_high(m_LED);
100	lcd.print("YESIL LED ");	104	output_low(k_LED);
101	lcd.setCursor(0,1);	105	output_low(y_LED);
102	lcd.print("AKTIF ");	106	output_low(d_LED);}
103	digitalWrite(y_LED, HIGH);	107	else if (yesil_renk < kirmizi_renk && yesil_renk <
104	digitalWrite(k_LED, LOW);		mavi_renk && yesil_renk < diger_renk)



105	digitalWrite(m_LED, LOW);	108	{lcd_putc("\f");
106	digitalWrite(d_LED, LOW);}	109	lcd_putc("YESİL LED ");
107	else	110	lcd_putc("\n AKTIF ");
108	{lcd.clear();	111	output_high(y_LED);
109	lcd.setCursor(0,0);	112	output_low(k_LED);
110	lcd.print("DİĞER LED ");	113	output_low(m_LED);
111	lcd.setCursor(0,1);	114	output_low(d_LED);}
112	lcd.print("AKTIF ");	115	else
113	digitalWrite(d_LED, HIGH);	116	{lcd_putc("\f");
114	digitalWrite(k_LED, LOW);	117	lcd_putc("DİĞER LED ");
115	digitalWrite(y_LED, LOW);	118	lcd_putc("\n AKTIF ");
116	digitalWrite(m_LED, LOW);}	119	output_high(d_LED);
117	delay(1000);}	120	output_low(k_LED);
		121	output_low(y_LED);
		122	output_low(m_LED);}
		123	delay_ms(1000);}

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

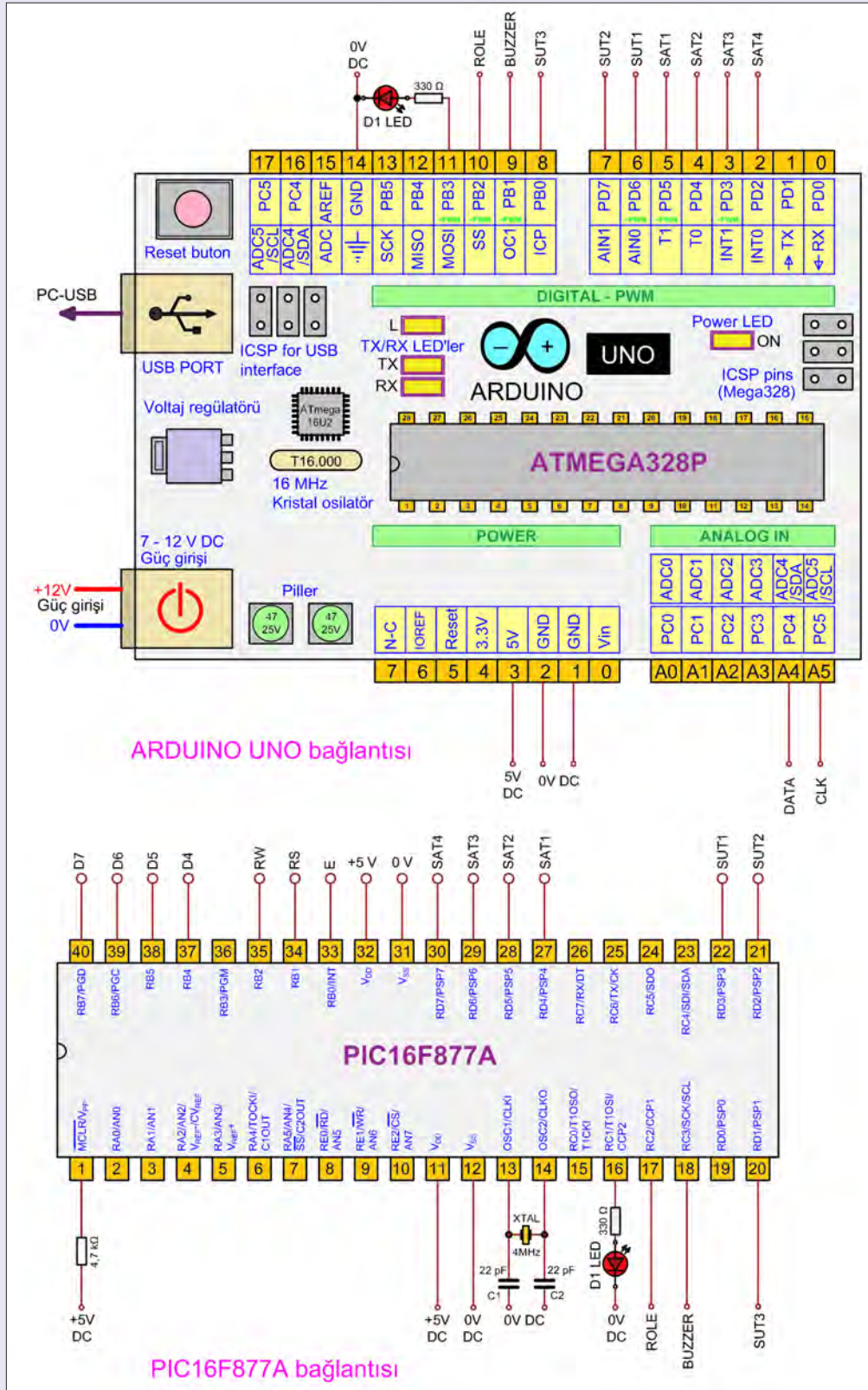
1. TCS3200 renk sensörü filtrelerinin mikrodnetleyici ile kontrol edilmesinin sebebi nedir?
2. Dijital sinyallerin genişliği ölçülürken timer sayıcılarının kullanılmasının sebebi nedir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

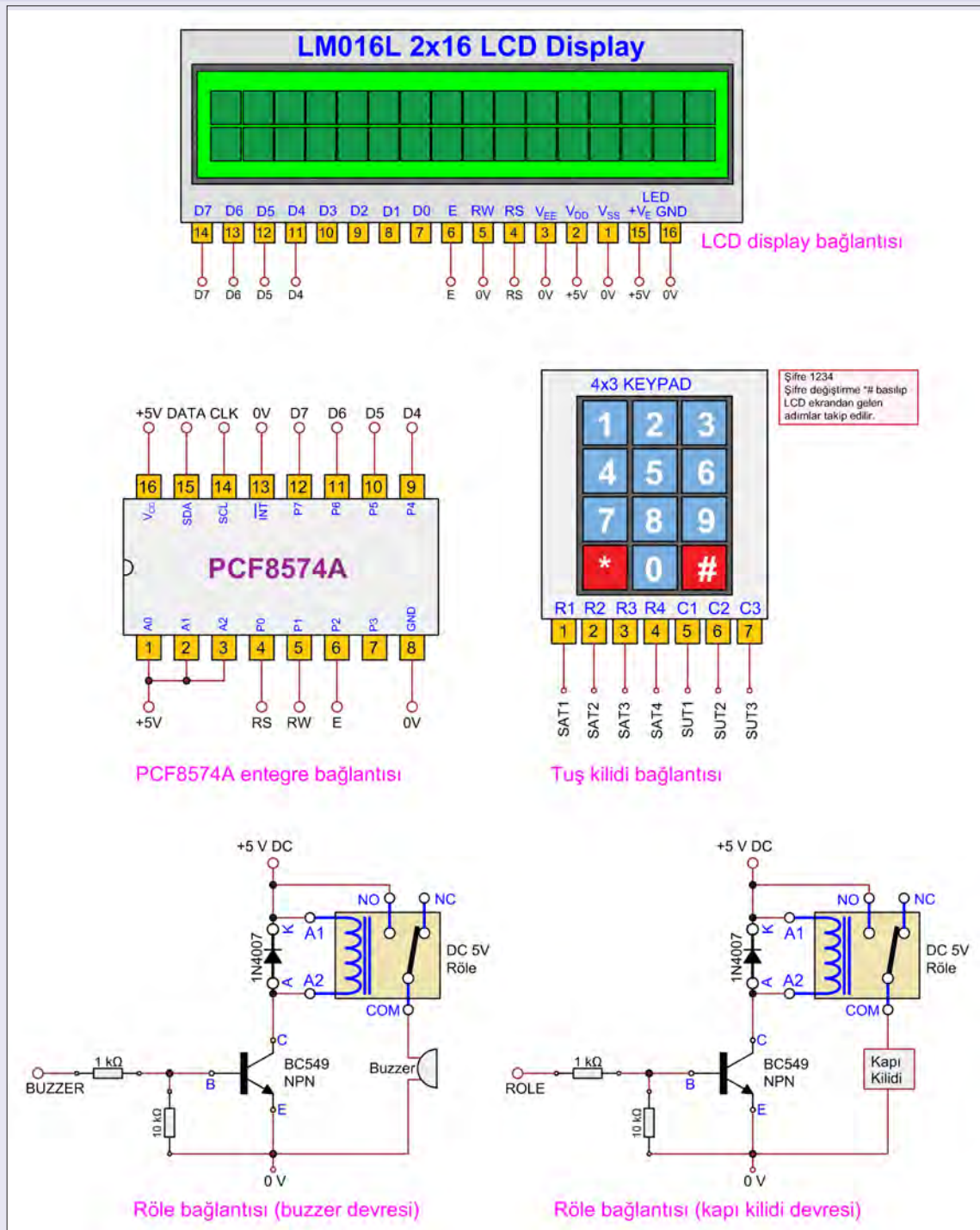
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

1.13. Şifreli Kilit Devresi

Devre Şeması: Görsel 1.46'da şifreli kapı kilit devresinin kontrol devre şeması, Görsel 1.47'de şifreli kapı kilit devresi görülmektedir.



Görsel 1.46: Şifreli kapı kilidi devresinin kontrol devre şeması



Görsel 1.47: Şifreli kapı kilidi devre bağlantısı

Devre Hakkında: Şifreli kilit devresi; 4x3 tuş takımı (keypad), LCD ekran, kapı kilidi, buzzer ve röleler kullanılarak tasarlanmıştır. Mikrodenetleyici, tuş takımından girilen şifreyi okur ve girilen şifre kapı kilidi şifresi ile aynıysa kapı kilidini açar. Devre ilk defa çalıştırıldığında EEPROM hafızası boş olduğundan şifre **1234**'tür. Şifre doğru yazıldığında kapı kilidi açılır ve buzzerden ikaz sesi duyulur. Şifre değişikliği yapılmak istenilirse ***#** (yıldız, kare) tuşlarına basılır. LCD ekranda görülen bilgilendirme yazısına göre şifre değişikliği yapılır. Şifre değişikliği yapıldıktan sonra şifre EEPROM hafızaya yazılır. Kapı kilidi enerjisi kesilip gelse dahi EEPROM hafızadaki şifre kullanılır (Görsel 1.46-1.47).

Malzemeler: Tablo 1.31’de iki devrenin malzeme listesi görölmektedir.

Tablo 1.31: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodnetleyicili kart	1 adet
PIC16F877A	Mikrodnetleyici entegre	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
LM016	4x16 LCD ekran	1 adet
PCF8574A	I2C seri-paralel kaymalı kaydedici	1 adet
Buzzer	5V	1 adet
Röle	5V DC	2 adet
LED diyot	Kırmızı	1 adet
Transistör	BC549 NPN	2 adet
Direnç	1 kΩ	2 adet
Direnç	330 Ω	4 adet
Direnç	10 kΩ	3 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.32’de iki devrenin kod yazılımları görölmektedir.

Tablo 1.32: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F877A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <Keypad.h>	2	#include <16F877A.h>
3	#include <EEPROM.h>	3	#fuses XT, NOWDT, PUT, NOPROTECT, BROWNOUT, NOLVP
4	#include <LiquidCrystal_I2C.h>	4	#use delay(clock=4000000)
5	LiquidCrystal_I2C lcd(0x3f,16,2);	5	#define use_portb_lcd TRUE
6	//değişkenler	6	#include <lcd.c>
7	const byte satir = 4;	7	#define sut1 pin_d3
8	const byte stun = 3;	8	#define sut2 pin_d2
9	const int buzzer =9;	9	#define sut3 pin_d1
10	const int ROLE =10;	10	#define sut4 pin_d0
11	const int LED =11;	11	#define sat1 pin_d4
12	char keys[satir][stun] = {	12	#define sat2 pin_d5
13	{'1','2','3'},	13	#define sat3 pin_d6
14	{'4','5','6'},	14	#define sat4 pin_d7
15	{'7','8','9'},	15	#define buzzer pin_c3
16	{' ','0','#'}	16	#define ROLE pin_c2



```

17 };
18 byte satirPins[satir] = {5, 4, 3, 2};
19 byte stunPins[stun] = {6, 7, 8};
20 Keypad keypad = Keypad(makeKeymap(keys), satirPins,
    stunPins, satir, stun);
21 char tus_k=0;
22 int tus_giris=0;
23 int sifre[4]={1,2,3,4};
24 int Ysifre[] = {0,0,0,0,0,0,0};
25 int bilgi=0, i=0;
26 int hata = 0, yeni, ytekrar, yildiz;
27 int gelen, sifre_degistir=0, degistir=0;
28 //alt programlar
29 void LED_yak ( )
30 {digitalWrite(LED,HIGH);
31 delay(250);
32 digitalWrite(LED,LOW);}
33 void ikaz(int x)
34 {for(i=0; i<x; i++)
35 {digitalWrite(buzzer,HIGH);
36 delay(100);
37 digitalWrite(buzzer,LOW);
38 delay(100);}}
39 //genel ayarlar
40 void setup()
41 {lcd.begin();
42 pinMode(buzzer,OUTPUT);
43 pinMode(ROLE,OUTPUT);
44 pinMode(LED,OUTPUT);
45 ytekrar=EEPROM.read(4);
46 if (ytekrar==15)
47 {for(i=0;i<=3;i++)
48 {sifre [i]=EEPROM.read(i);}}
49 else
50 {for(i=0;i<=3;i++)
51 {sifre[i]=i+1;}}
52 lcd.clear();
53 lcd.print("SIFRELI KILIT");
54 lcd.setCursor(1,1);
55 lcd.print("İLERİ MİKRO");
56 delay(1500);
57 lcd.clear();
58 lcd.print(" SIFREYİ GIRINIZ");}
59 //ana program

```

```

17 #define LED pin_c1
18 //değişkenler
19 char tus_k=0;
20 int tus_giris;
21 int sifre[]={1,2,3,4};
22 int Ysifre[] = {0,0,0,0,0,0,0};
23 int bilgi=0, i=0;
24 int hata = 0, yeni, ytekrar, yildiz;
25 int gelen, sifre_degistir=0, degistir=0;
26 //alt programlar
27 char keypad_oku()
28 {output_d(0x00);
29 tus_k = "";
30 output_high(sat1);
31 if (input(sut1))
32 { delay_ms(500); tus_k=1;tus_giris=1; }
33 if (input(sut2))
34 { delay_ms(500); tus_k=2;tus_giris=1; }
35 if (input(sut3))
36 { delay_ms(500); tus_k=3;tus_giris=1; }
37 output_low(sat1);
38 output_high(sat2);
39 if (input(sut1))
40 { delay_ms(500); tus_k=4;tus_giris=1; }
41 if (input(sut2))
42 { delay_ms(500); tus_k=5;tus_giris=1; }
43 if (input(sut3))
44 {delay_ms(500); tus_k=6;tus_giris=1; }
45 output_low(sat2);
46 output_high(sat3);
47 if (input(sut1))
48 { delay_ms(500); tus_k=7;tus_giris=1; }
49 if (input(sut2))
50 { delay_ms(500); tus_k=8;tus_giris=1; }
51 if (input(sut3))
52 { delay_ms(500); tus_k=9;tus_giris=1; }
53 output_low(sat3);
54 output_high(sat4);
55 if (input(sut1))
56 { delay_ms(500); tus_k=0xA;tus_giris=1; }
57 if (input(sut2))
58 { delay_ms(500); tus_k=0;tus_giris=1; }
59 if (input(sut3))
60 { delay_ms(500); tus_k=0xB;tus_giris=1; }
61 output_low(sat4);
62 return tus_k; }

```

```

60 void loop()
61 {char tus_k = keypad.getKey();
62 if (tus_k)
63 {tus_giris=1;
64 gelen=tus_k-'0';
65 if(gelen==6)
66 {LED_yak();
67 ikaz(2);
68 yildiz=1;
69 bilgi=0;
70 tus_giris=0;}
71 if(gelen==13)
72 {LED_yak();
73 ikaz(2);
74 tus_giris=0;
75 if (yildiz==1) {sifre_degistir=1;}}
76 if(sifre_degistir)
77 {if (tus_giris==1)
78 {LED_yak();
79 ikaz(1);
80 tus_giris=0;}
81 degistir=0;
82 lcd.clear();
83 lcd.print(" SIFRE DEGISTIR");
84 delay(1500);
85 lcd.clear();
86 lcd.print("ESKI SIFRE GIRNIZ");
87 lcd.setCursor(7,1);
88 while(sifre_degistir)
89 {char tus_k = keypad.getKey();
90 if (tus_k)
91 {tus_giris=1;
92 gelen=tus_k-'0';
93 if (tus_giris==1)
94 {LED_yak();
95 ikaz(1);
96 tus_giris=0;
97 lcd.print("*");
98 if (sifre[degistir] != gelen)
99 {hata=1;
100 degistir++;}
101 if (sifre[degistir] == gelen)
102 {degistir++;}
103 if (degistir>3)
104 {if (hata==0)
105 {lcd.clear();

```

```

63 void LED_yak ( )
64 {output_high(LED);
65 delay_ms(250);
66 output_low(LED);}
67 void ikaz(int x)
68 {for(i=0; i<x; i++)
69 {output_high(buzzer);
70 delay_ms(100);
71 output_low(buzzer);}}
72 //genel ayarlar
73 void main ( )
74 {setup_psp(PSP_DISABLED);
75 setup_timer_1(T1_DISABLED);
76 setup_timer_2(T2_DISABLED,0,1);
77 setup_adc_ports(NO_ANALOGS);
78 setup_adc(ADC_OFF);
79 set_tris_d(0x0F);
80 ytekrar=read_eeprom(4);
81 if (ytekrar==15)
82 {for(i=0;i<=3;i++)
83 {sifre [i]=read_eeprom(i);}}
84 else
85 {for(i=0;i<=3;i++)
86 {sifre[i]=i+1;}}
87 lcd_init();
88 printf(lcd_putc,"\ SIFRELI KILIT");
89 printf(lcd_putc,"\n ILERI MIKRO");
90 delay_ms(1500);
91 printf(lcd_putc,"\f SIFREYI GIRINIZ");
92 //ana program
93 while(1)
94 {gelen = keypad_oku();
95 if(gelen==0xA)
96 {if (tus_giris==1)
97 {LED_yak();
98 ikaz(2);
99 tus_giris=0;
100 yildiz=1;
101 bilgi=0;}}
102 if(gelen==0xB)
103 {if (tus_giris==1)
104 {LED_yak();
105 ikaz(2);
106 tus_giris=0;
107 if (yildiz==1) {sifre_degistir=1;}}
108 if(sifre_degistir)

```



106	lcd.print("YENİ SIRFE GIRNZ ");	109	{if (tus_giris==1)
107	lcd.setCursor(7,1);	110	{LED_yak();
108	yeni=0;	111	ikaz(1);
109	ytekrar=1;	112	tus_giris=0;}
110	while(ytekrar)	113	degistir=0;
111	{char tus_k = keypad.getKey();	114	printf(lcd_putc,"\\f SIFRE DEGISTIR");
112	if (tus_k)	115	delay_ms(1500);
113	{tus_giris=1;	116	printf(lcd_putc,"\\f ESKI SIFRE GIRNZ\\n ");
114	gelen=tus_k-'0';	117	printf(lcd_putc," ");
115	if (tus_giris==1)	118	while(sifre_degistir)
116	{LED_yak();	119	{gelen = keypad_oku();
117	ikaz(1);	120	if (tus_giris==1)
118	tus_giris=0;	121	{LED_yak();
119	Ysifre[yeni]=gelen;	122	ikaz(1);
120	yeni++;	123	tus_giris=0;
121	lcd.print("*");	124	printf(lcd_putc,"*");
122	if (yeni==4)	125	if (sifre[degistir] != gelen)
123	{lcd.clear();	126	{hata=1;
124	lcd.print("TEKRAR SIRFE GRN");	127	degistir++;}
125	lcd.setCursor(7,1);}	128	if (sifre[degistir] == gelen)
126	if (yeni==8)	129	{degistir++;}
127	{if ((Ysifre[0]==Ysifre[4])&& (Ysifre[1]==Ysifre[5])&&	130	if (degistir>3)
128	(Ysifre[2]==Ysifre[6])&& (Ysifre[3]==Ysifre[7]))	131	{if (hata==0)
129	{sifre[0]=Ysifre[0] ; sifre[1]=Ysifre[1] ; sifre[2]=Ysifre[2];	132	{printf(lcd_putc,"\\f YENİ SIRFE GIRNZ\\n");
130	sifre[3]=Ysifre[3];	133	printf(lcd_putc," ");
131	for(i=0;i<=3;i++) //şifreyi EEPROM'a yaz.	134	yeni=0;
132	{EEPROM.update(i,sifre[i]);}	135	ytekrar=1;
133	EEPROM.update(4,15);	136	while(ytekrar)
134	lcd.clear();	137	{gelen = keypad_oku();
135	lcd.print(" SIRFE DEGISTI");	138	if (tus_giris==1)
136	delay(1000);	139	{LED_yak();
137	ytekrar=0;	140	ikaz(1);
138	sifre_degistir=0;	141	tus_giris=0;
139	bilgi=0;	142	Ysifre[yeni]=gelen;
140	hata=0;	143	yeni++;
141	lcd.clear();	144	printf(lcd_putc,"*");
142	lcd.print(" SIFREYİ GIRINIZ");}	145	if (yeni==4)
143	else	146	{printf(lcd_putc,"\\f TEKRAR SIRFE GRN\\n");
144	{lcd.clear();	147	printf(lcd_putc," ");}
145	lcd.print(" SIFRE DEGİSMEDİ");	148	if (yeni==8)
146	delay(1000);	149	{if ((Ysifre[0]==Ysifre[4]) && (Ysifre[1]==Ysifre[5])
147	ytekrar=0;	150	&& (Ysifre[2]==Ysifre[6]) && (Ysifre[3]==Ysifre[7]))
148	sifre_degistir=0;	151	{sifre[0]=Ysifre[0] ; sifre[1]=Ysifre[1] ;
149	bilgi=0;	152	sifre[2]=Ysifre[2]; sifre[3]=Ysifre[3];
150	hata=0;	153	for(i=0;i<=3;i++) //şifreyi EEPROM'a yaz
151	lcd.clear();	154	{write_eeprom(i,sifre[i]);}
152	lcd.print("SIFREYİ GIRINIZ");}}}}}	155	printf(lcd_putc,"\\f SIRFE DEGISTI");
153		156	

151	else	154	delay_ms(1000);
152	{lcd.clear();	155	ytekrar=0;
153	lcd.print("HATALI SIFRE");	156	sifre_degistir=0;
154	lcd.setCursor(1,1);	157	bilgi=0;
155	lcd.print("ANA MENUYE DONUS");	158	hata=0;
156	delay(2000);	159	printf(lcd_putc,"\f SIFREYI GIRINIZ");}
157	lcd.clear();	160	else
158	lcd.print(" SIFRE GIRINIZ ");	161	{printf(lcd_putc,"\f SIRFE DEGISMEDI");
159	sifre_degistir=0;	162	delay_ms(1000);
160	bilgi=0;	163	ytekrar=0;
161	hata=0;}	164	sifre_degistir=0;
162	degistir=0;}}	165	bilgi=0;
163	bilgi++;}}	166	hata=0;
164	if(sifre_degistir==0)	167	printf(lcd_putc,"\f SIFREYI GIRINIZ"); }}}
165	{if (tus_giris==1)	168	else
166	{LED_yak();	169	{printf(lcd_putc,"\f YANLIS SIFRE \n ANA MENUYE DONUS");
167	ikaz(1);	170	delay_ms(2000);
168	yildiz=0;	171	printf(lcd_putc,"\f SIFRE GIRINIZ ");
169	tus_giris=0;	172	sifre_degistir=0;
170	lcd.setCursor(7+bilgi,1);	173	bilgi=0;
171	lcd.print("*");	174	hata=0;}
172	if (sifre[bilgi] != gelen)	175	degistir=0;}}
173	{hata=1;	176	bilgi++;}}
174	sifre[3]=Ysifre[3];	177	if(sifre_degistir==0)
175	for(i=0;i<=3;i++) //şifreyi EEPROM'a yaz.	178	{if (tus_giris==1)
176	{EEPROM.update(i,sifre[i]);}	179	{LED_yak();
177	EEPROM.update(4,15);	180	ikaz(1);
178	lcd.clear();	181	yildiz=0;
179	lcd.print(" SIRFE DEGISTI");	182	tus_giris=0;
180	delay(1000);	183	lcd_gotoxy(7+bilgi,2);
181	ytekrar=0;	184	printf(lcd_putc,"*");
182	sifre_degistir=0;	185	if (sifre[bilgi] != gelen)
183	bilgi=0;	186	{hata=1;
184	hata=0;	187	bilgi++;}
185	lcd.clear();	188	if (sifre[bilgi] == gelen)
186	lcd.print(" SIFREYI GIRINIZ");}	189	{bilgi++;}
187	else	190	if (bilgi>3)
188	{lcd.clear();	191	{if (hata==0)
189	lcd.print(" SIFRE DEGISMEDI");	192	{printf(lcd_putc,"\f SIFRE DOGRU");
190	delay(1000);	193	printf(lcd_putc,"\n KAPI ACILIYOR.");
191	ytekrar=0;	194	output_high(ROLE);
192	sifre_degistir=0;	195	delay_ms(2000);
193	bilgi=0;	196	output_low(ROLE);
194	hata=0;	197	printf(lcd_putc,"\f SIFREYI GIRINIZ"); }
195	lcd.clear();	198	else
196	lcd.print(" SIFREYI GIRINIZ");}	199	{printf(lcd_putc,"\f HATALI SIFRE \n GIRDINIZ");
197	bilgi=0;}}}}	200	lcd_gotoxy(15,2);



		201	delay_ms(1500);
		202	hata=0;
		203	printf(lcd_putc, "\f SIFREYİ GIRINIZ");}
		204	bilgi=0;}}}}

İşlem Basamakları

1. Uygun tipli mikrodeneleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodeneleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Şifreli kilit programında EEPROM hafızasının kullanılmasının sebebi nedir?
2. Şifreyi 4 haneli sayıdan 6 haneli sayıya çıkartmak için program üzerinde yapılacak değişiklikler nelerdir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

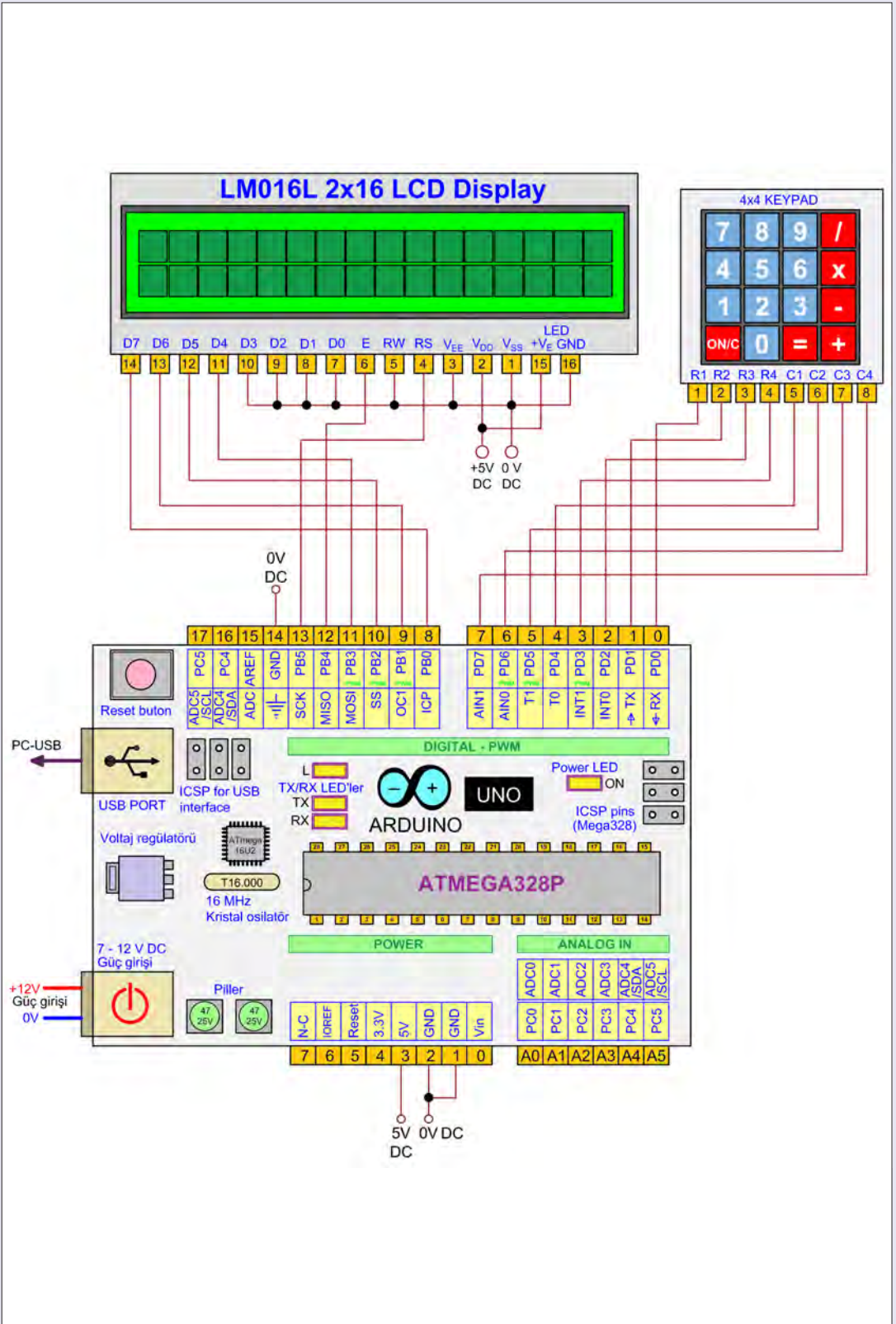
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



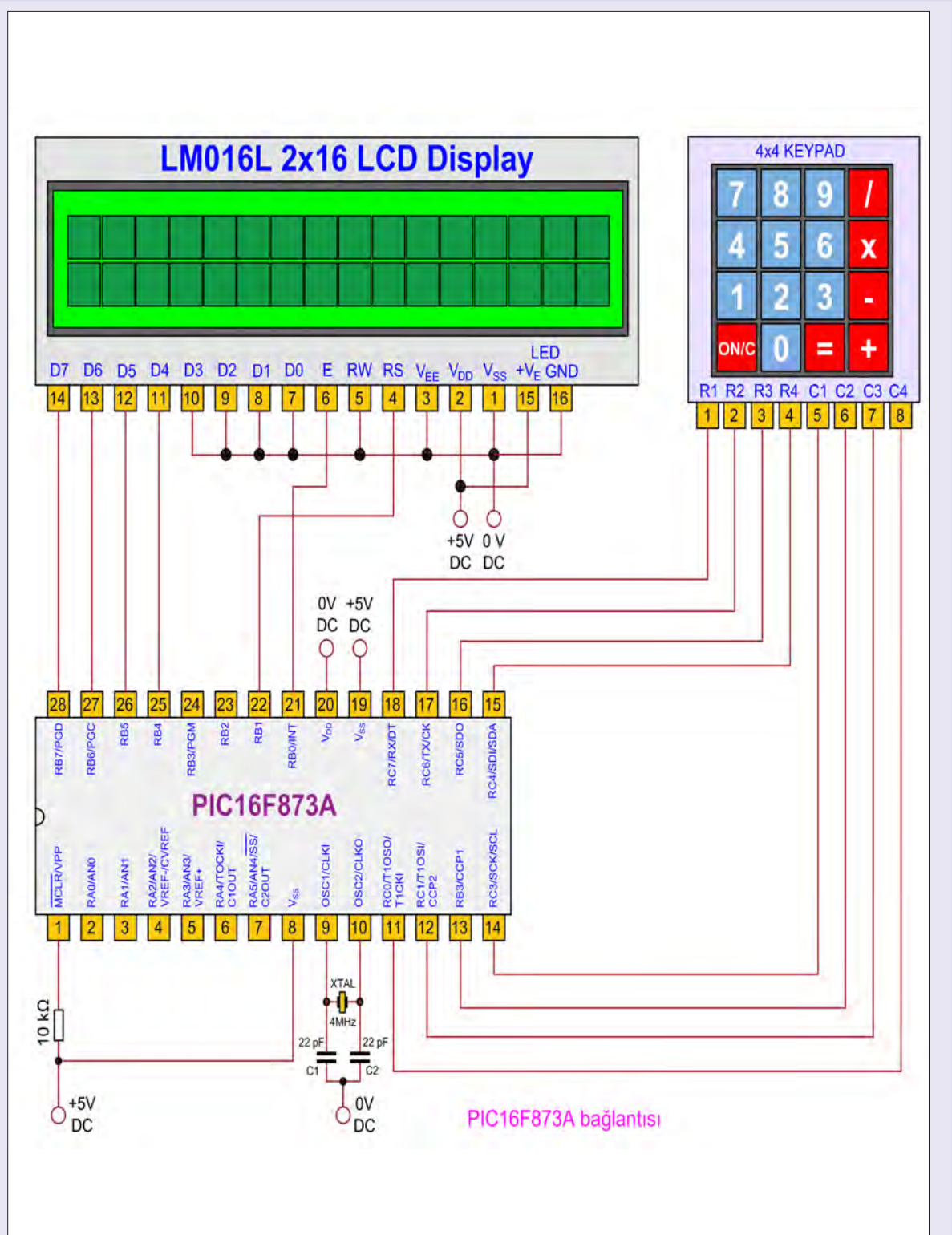
1.14. Dijital Hesap Makinesi Devresi



Devre Şeması: Görsel 1.48'de ARDUINO UNO ile hesap makinesi devresi, Görsel 1.49'da PIC16F873A entegrasi ile hesap makinesi devresi görülmektedir.



Görsel 1.48: ARDUINO UNO ile hesap makinesi devresi



Görsel 1.49: PIC16F873A entegresi ile hesap makinesi devresi

Devre Hakkında: Hesap makinesi devresi, 4x4 tuş takımı (keypad) ve LCD ekran kullanılarak tasarlanmıştır. Mikrodenetleyici, tuş takımından girilen veriler üzerinde dört işlem yapar ve işlem sonunu gösterir. Devreye enerji uygulandığında mikrodenetleyici, tuş takımından gelen sayı ve işlemlere göre matematiksel hesaplama yapar ve işlem sonucu LCD ekranda görüntülenir (Görsel 1.48-1.49).

Malzemeler: Tablo 1.33'te iki devrenin malzeme listesi görülmektedir.

Tablo 1.33: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F873A	Mikrodenetleyici entegre	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaketa	Yeterli ölçülerde	-
LM016	2x16 LCD ekran	1 adet
Direnç	10 kΩ	1 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.34'de iki devrenin kod yazılımları görülmektedir.

Tablo 1.34: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F873A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <LiquidCrystal.h>	2	#include <PIC16F873A.h>
3	#include <Keypad.h>	3	#fuses XT, NOWDT, PUT, NOPROTECT, BROWNOUT, NOLVP
4	//değişkenler	4	#use delay(clock=4000000)
5	const byte satir = 4;	5	#define use_portb_lcd TRUE
6	const byte stun = 4;	6	#include <lcd.c>
7	char tus_k, tus_sifir, bekleyen_islem=0, bolum_var=0;	7	#define round(x) (signed int32)(x < 0 ? (x - 0.5) : (x + 0.5))
8	float hesaplanan_b = 0.0;	8	#define sut1 pin_c3
9	long hesaplanan = 0;	9	#define sut2 pin_c2
10	boolean birinci_sayi = true, silme_aktif=false, sayi_yok;	10	#define sut3 pin_c1
11	String sayi0 = "", sayi1="";	11	#define sut4 pin_c0
12	int sayi2=0;	12	#define sat1 pin_c7
13	char islem = ' ';	13	#define sat2 pin_c6
14	char keys[satir][stun] = {	14	#define sat3 pin_c5
15	{'7','8','9','/'},	15	#define sat4 pin_c4
16	{'4','5','6','*'},	16	//değişkenler
17	{'1','2','3','-'},	17	char tus_k, tus_sifir, bekleyen_islem=0;
18	{'C','0','=','+'};	18	int16 sayi0, sayi1,sayi2;
19	byte satirPins[satir] = { 0, 1, 2, 3 };	19	float hesaplanan=0.0;
20	byte stunPins[stun] = { 4, 5, 6, 7 };	20	char islem=0, tus_giris=0;
21	Keypad keypad = Keypad(makeKeymap(keys),	21	int8 bolum_var=0;
22	satirPins, stunPins, satir, stun);	22	BOOLEAN sonuc, silme = true, sayi_yok=false;
22	const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;	23	//alt programlar



```

23 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
24 //alt programlar
25 void ekran_temizle()
26 {lcd.clear();
27 lcd.print("0");
28 lcd.setCursor(0, 0);
29 sayi0 = "";
30 sayi1 = "";
31 hesaplanan = 0;
32 islem = '0';
33 bekleyen_islem=0;
34 void hesapla()
35 {if (islem == '+')
36 {hesaplanan = sayi0.toInt() + sayi1.toInt();
37 bekleyen_islem=0;
38 if (islem == '-')
39 {hesaplanan = sayi0.toInt() - sayi1.toInt();
40 bekleyen_islem=0;
41 if (islem == '*')
42 {hesaplanan = sayi0.toInt() * sayi1.toInt();
43 bekleyen_islem=0;
44 if (islem == '/')
45 {hesaplanan_b = sayi0.toFloat() / sayi1.toFloat();
46 bekleyen_islem=0;
47 bolum_var=1;}}
48 //genel ayarlar
49 void setup()
50 {lcd.begin(16,2);
51 lcd.setCursor(0, 0);
52 lcd.print("HESAP MAKİNESİ");
53 lcd.setCursor(0, 1);
54 lcd.print("UYGULAMASI");
55 delay(1000);
56 lcd.clear();
57 lcd.print("0");
58 lcd.setCursor(0, 0);
59 sayi_yok=false;
60 //ana program
61 void loop()
62 {char tus_k = keypad.getKey();
63 if (!tus_k==NO_KEY)
64 {if (silme_aktif == true)
65 {ekran_temizle();
66 silme_aktif = false;
67 sayi_yok=false;
68 if (sayi_yok==false && tus_k == '-' )

```

```

24 char keypad_oku()
25 {output_c(0x00);
26 tus_k = "";
27 output_high(sat1);
28 if (input(sut1))
29 { delay_ms(500); tus_k='7';tus_giris=1; }
30 if (input(sut2))
31 { delay_ms(500); tus_k='8';tus_giris=1; }
32 if (input(sut3))
33 { delay_ms(500); tus_k='9';tus_giris=1; }
34 if (input(sut4))
35 { delay_ms(500); tus_k='/' ;tus_giris=1; }
36 output_low(sat1);
37 output_high(sat2);
38 if (input(sut1))
39 { delay_ms(500); tus_k='4';tus_giris=1; }
40 if (input(sut2))
41 { delay_ms(500); tus_k='5';tus_giris=1; }
42 if (input(sut3))
43 { delay_ms(500); tus_k='6';tus_giris=1; }
44 if (input(sut4))
45 { delay_ms(500); tus_k='*';tus_giris=1; }
46 output_low(sat2);
47 output_high(sat3);
48 if (input(sut1))
49 { delay_ms(500); tus_k='1';tus_giris=1; }
50 if (input(sut2))
51 { delay_ms(500); tus_k='2';tus_giris=1; }
52 if (input(sut3))
53 { delay_ms(500); tus_k='3';tus_giris=1; }
54 if (input(sut4))
55 { delay_ms(500); tus_k='-';tus_giris=1; }
56 output_low(sat3);
57 output_high(sat4);
58 if (input(sut1))
59 { delay_ms(500); tus_k='C' ;tus_giris=1; }
60 if (input(sut2))
61 { delay_ms(500); tus_k='0';tus_giris=1; }
62 if (input(sut3))
63 { delay_ms(500); tus_k='=';tus_giris=1; }
64 if (input(sut4))
65 { delay_ms(500); tus_k='+';tus_giris=1; }
66 output_low(sat4);
67 return tus_k; }
68 void hesapla()
69 {if (islem=='+' )

```

69	{if (birinci_sayi == true)	70	sayi0 = sayi1+sayi2;
70	sayi0 = sayi0 + tus_k;	71	if (islem=='-')
71	lcd.print(tus_k);}	72	sayi0 = sayi1-sayi2;
72	if (tus_k == '1' tus_k == '2' tus_k == '3' tus_k == '4'	73	if (islem=='*')
	tus_k == '5' tus_k == '6' tus_k == '7'	74	sayi0 = sayi1*sayi2;
	tus_k == '8' tus_k == '9' tus_k == '0') {	75	if (islem=='/')
73	if (birinci_sayi == true)	76	{hesaplanan = (float)sayi1 / (float)sayi2;
74	{sayi0 = sayi0 + tus_k;	77	bolum_var=1;}}
75	lcd.print(tus_k);	78	void tus_oku()
76	sayi_yok=true;}	79	{if(silme==true)
77	else	80	{lcd_send_byte(0,1);
78	{sayi1 = sayi1 + tus_k;	81	lcd_gotoxy(1,1);
79	lcd.print(tus_k);}}	82	silme=false;}
80	if (sayi_yok==true && (tus_k == '+' tus_k == '-'	83	if (tus_k=='C')
	tus_k == '*' tus_k == '/'))	84	{printf(lcd_putc, "\f0");
81	{if(bekleyen_islem)	85	sayi0=0; sayi1=0; sayi2=0; sonuc=false; silme=true;
82	{lcd.clear();		tus_sifir=0; islem=0;
83	lcd.setCursor(0, 0);	86	lcd_gotoxy(1,1);
84	if (islem == '+')	87	sayi_yok=false;}
85	{sayi2= sayi0.toInt() + sayi1.toInt(); }	88	if (tus_k == '1')
86	if (islem == '-')	89	{sayi0 = (sayi0*10) + 1;
87	{sayi2= sayi0.toInt() - sayi1.toInt(); }	90	sayi_yok=true;}
88	if (islem == '*')	91	if (tus_k == '2')
89	{sayi2= sayi0.toInt() * sayi1.toInt(); }	92	{sayi0 = (sayi0*10) + 2;
90	if (islem == '/')	93	sayi_yok=true;}
91	{sayi2= sayi0.toInt() / sayi1.toInt(); }	94	if (tus_k == '3')
92	lcd.print(sayi2);	95	{sayi0 = (sayi0*10) + 3;
93	sayi1="";	96	sayi_yok=true;}
94	sayi0=String(sayi2);}	97	if (tus_k == '4')
95	if (tus_k == '+')	98	{sayi0 = (sayi0*10) + 4;
96	{ islem = '+';	99	sayi_yok=true;}
97	bekleyen_islem=1;}	100	if (tus_k == '5')
98	if (tus_k == '-')	101	{sayi0 = (sayi0*10) + 5;
99	{ islem = '-';	102	sayi_yok=true;}
100	bekleyen_islem=1;}	103	if (tus_k == '6')
101	if (tus_k == '*')	104	{sayi0 = (sayi0*10) + 6;
102	{ islem = '*';	105	sayi_yok=true;}
103	bekleyen_islem=1;}	106	if (tus_k == '7')
104	if (tus_k == '/')	107	{sayi0 = (sayi0*10) + 7;
105	{ islem = '/';	108	sayi_yok=true;}
106	bekleyen_islem=1;}	109	if (tus_k == '8')
107	lcd.print(islem);	110	{sayi0 = (sayi0*10) + 8;
108	if (birinci_sayi == true)	111	sayi_yok=true;}
109	{birinci_sayi = false;}}	112	if (tus_k == '9')
110	if (sayi_yok==true && tus_k == '=')	113	{sayi0 = (sayi0*10) + 9;
111	{hesapla();	114	sayi_yok=true;}



112	lcd.clear();	115	if (sayi_yok==true && tus_k == '=')
113	lcd.setCursor(0, 0);	116	{sayi2=sayi0;
114	lcd.print(sayi0);	117	sonuc = true;}
115	lcd.print(islem);	118	if (sayi_yok==true && (tus_k == '/' tus_k == '*'
116	lcd.print(sayi1);		tus_k == '-' tus_k == '+'))
117	lcd.setCursor(0, 1);	119	{if(bekleyen_islem)
118	lcd.print("=");	120	{sayi2 = sayi0;
119	if (bolum_var)	121	hesapla();
120	{lcd.print(hesaplanan_b);	122	tus_sifir=0;}
121	bolum_var=0;}	123	sayi1 = sayi0;
122	else	124	sayi0 =0;
123	{lcd.print(hesaplanan);}	125	if (tus_k == '+')
124	birinci_sayi = true;	126	{islem = '+';
125	silme_aktif=true;}	127	bekleyen_islem=1;}
126	if (tus_k == 'C')	128	if (tus_k == '-')
127	{ekran_temizle();	129	{ islem = '-';
128	sayi_yok=false;}}	130	bekleyen_islem=1;}
		131	if (tus_k == '*')
		132	{ islem = '*';
		133	bekleyen_islem=1;}
		134	if (tus_k == '/')
		135	{ islem = '/';
		136	bekleyen_islem=1;}
		137	silme=false;}}
		138	void goster()
		139	{if (sonuc==true)
		140	{printf(lcd_putc, "=");
		141	if (bolum_var)
		142	{printf(lcd_putc, "%f", hesaplanan);
		143	bolum_var=0;
		144	sayi0=round(hesaplanan);}
		145	else
		146	{printf(lcd_putc, "%ld", sayi0);}
		147	sonuc=false; silme=true; islem=0; tus_sifir=0;}}
		148	//genel ayarlar
		149	void main ()
		150	{setup_timer_1(T1_DISABLED);
		151	setup_timer_2(T2_DISABLED,0,1);
		152	setup_adc_ports(NO_ANALOGS);
		153	setup_adc(ADC_OFF);
		154	lcd_init();
		155	printf(lcd_putc, "\f HESAP MAKINESI");
		156	printf(lcd_putc, "\n UYGULAMASI");
		157	delay_ms(1000);
		158	printf(lcd_putc, "\f0");
		159	//ana program

		160	while(1)
		161	{tus_k = keypad_oku();
		162	if (tus_giris)
		163	{tus_oku();
		164	if (!islem==0)
		165	{printf(lcd_putc,"%f%ld",sayi1);
		166	printf(lcd_putc,"%c",islem);
		167	if(!sayi0==0 tus_sifir==1)
		168	{printf(lcd_putc,"%ld",sayi0);}}
		169	else
		170	{printf(lcd_putc,"%f%ld",sayi0);}}
		171	tus_giris=0;
		172	if (sonuc==true)
		173	hesapla();
		174	goster();}}

İşlem Basamakları

1. Uygun tipli mikrodeneleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodeneleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Programda sayı ve işlemleri ayırt etmek için kullanılan komut satır numarası nedir?
2. **hesapla()** alt programında yapılan işlem türleri hangileridir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

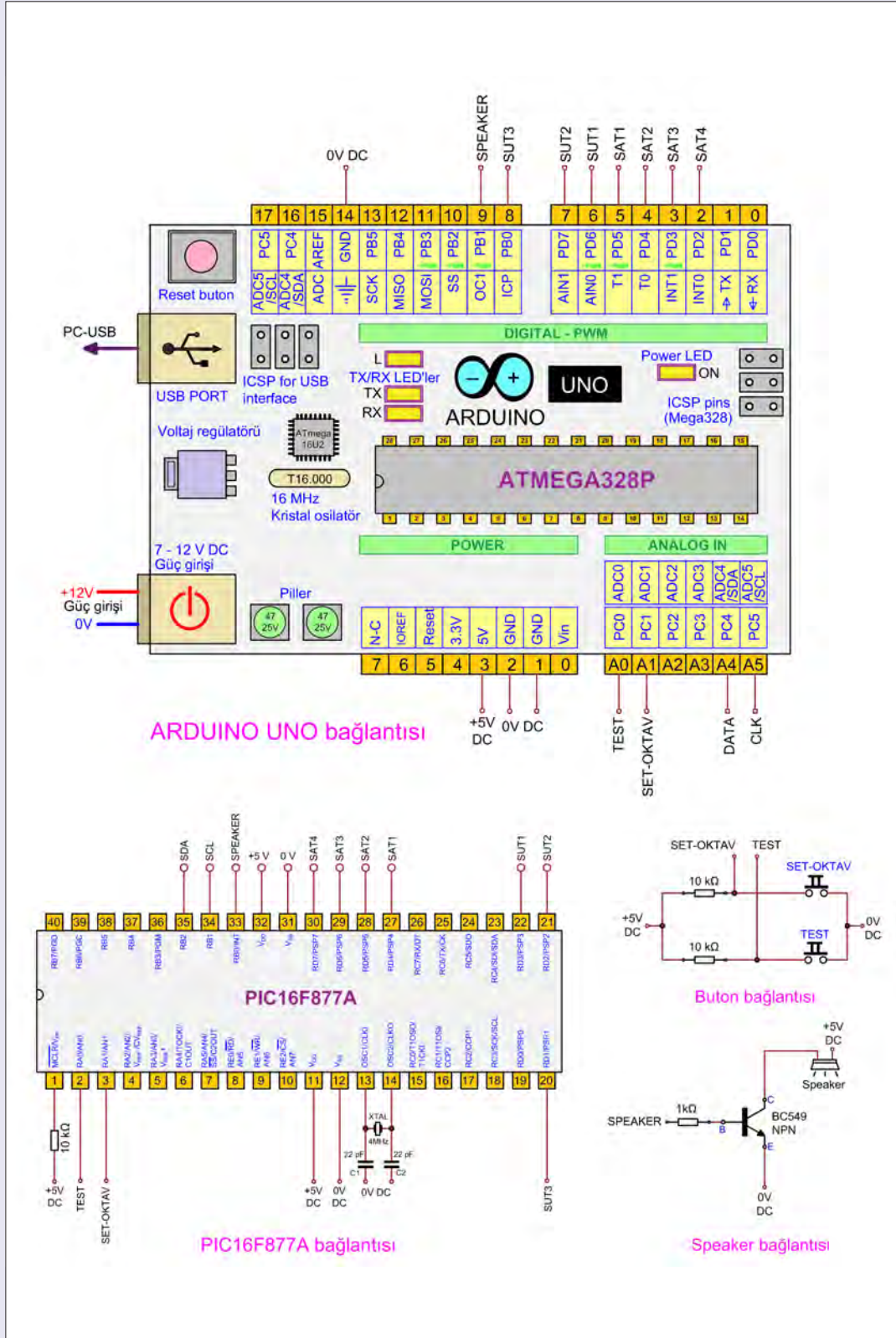
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



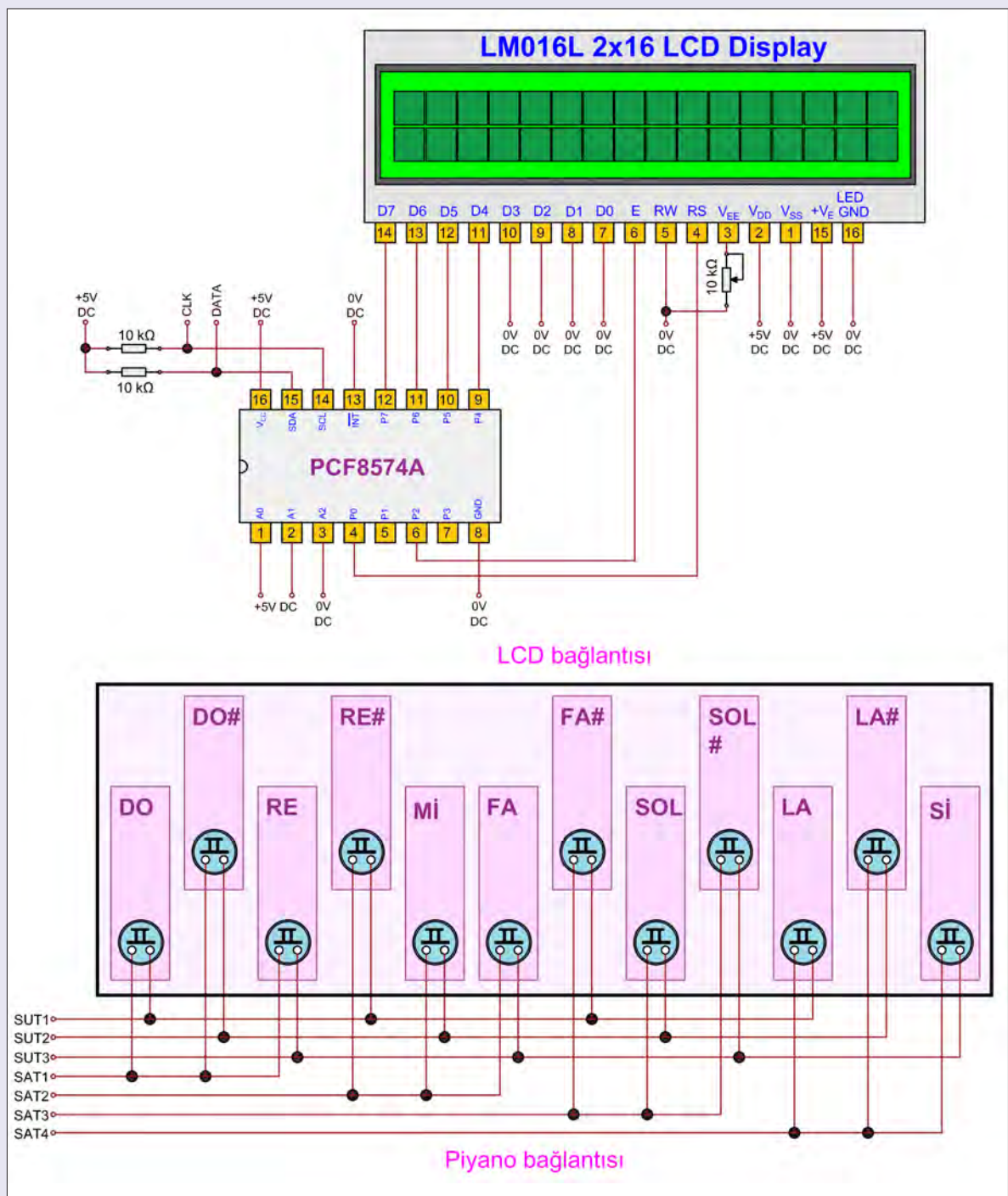
1.15. Elektronik Pişano Devresi



Devre Şeması: Görsel 1.50'de elektronik pişano kontrol devresi, Görsel 1.51'de elektronik pişano devre şeması görölmektedir.



Görsel 1.50: Elektronik pişano kontrol devresi



Görsel 1.51: Elektronik piyano devre şeması

Devre Hakkında: Elektronik piyano devresi; nota klavyesi, LCD ekran ve ses yükselteç devresi (bilgisayar ses kartı) kullanılarak tasarlanmıştır. Mikrodenetleyici, oktav seçimine göre klavyeden gelen tuşa ait nota frekansını üretir. Ses yükselteç devresinde yükseltilecek nota frekansı hoparlörden duyulur. Devreye enerji uygulandığında **TEST** butonuna basılırsa programa dâhil edilmiş müzik oynatılmaya başlar. **SET-OCTAV** butonuyla piyanonun oktav ayarı yapılır ve ayarın sonucu LCD ekranda görüntülenir. Oktav ayarı 0-7 tonda toplam sekiz kademelidir. Piyano klavyesindeki butona basıldığında butona karşılık gelen notanın oktav değerine göre üretilen ton (frekans) hoparlörden duyulur (Görsel 1.51).



Malzemeler: Tablo 1.35’de iki devrenin malzeme listesi görülmektedir.

Tablo 1.35: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F877A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
LM016	2x16 LCD ekran	1 adet
PCF8574A	I2C seri-paralel kaymalı kaydedici	1 adet
Buton	Mikro siviç	14 adet
Hoparlör	Ses yükselteçli	14 adet
Direnç	1 kΩ	1 adet
Direnç	10 kΩ	4 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 1.36’da iki devrenin kod yazılımları görülmektedir.

Tablo 1.36: Devrelerin Kodları

No.	Kodlar (ARDUINO)	No.	Kodlar (PIC16F877A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <Keypad.h>	2	#include <16F877A.h>
3	#include <LiquidCrystal_I2C.h>	3	#fuses XT, NOWDT, PUT, NOPROTECT, BROWNOUT, NOLVP
4	LiquidCrystal_I2C lcd(0x23,16,2);	4	#use delay(clock=4000000)
5	//değişkenler	5	#define PCF_SDA PIN_B2
6	const byte satir = 4;	6	#define PCF_SCL PIN_B1
7	const byte stun = 3;	7	#use i2c(master, sda=PCF_SDA, scl=PCF_SCL)
8	const int ses_out =9;	8	#define PCF8574_ID 3
9	const int test =A0;	9	#define PCF_ADD 64
10	const int octav =A1;	10	#define RS P0
11	char keys[satir][stun] = {	11	#define RW P1
12	{'1','2','3'},	12	#define E P2
13	{'4','5','6'},	13	#define DB4 P4
14	{'7','8','9'},	14	#define DB5 P5
15	{'.','<','>'};	15	#define DB6 P6
16	byte satirPins[satir] = {5, 4, 3, 2};	16	#define DB7 P7
17	byte stunPins[stun] = {6, 7, 8};	17	#define sut1 pin_d3
18	Keypad keypad = Keypad(makeKeymap(keys), satirPins, stunPins, satir, stun);	18	#define sut2 pin_d2
		19	#define sut3 pin_d1

```

19 long tones []={0,16,17,18,19,21,22,23,25,26,28,29,31};
20 long nota=0, gelen=0;
21 uint8_t oktav=0;
22 //genel ayarlar
23 void setup(){
24   lcd.begin();
25   pinMode(ses_out,OUTPUT);
26   pinMode(test,INPUT_PULLUP);
27   pinMode(octav,INPUT_PULLUP);
28   lcd.clear();
29   lcd.print("12 TUSLU");
30   lcd.setCursor(1,1);
31   lcd.print("7 OKTAV PIYANO");}
32 //ana program
33 void loop()
34 {if (!digitalRead(test))
35 {tone(ses_out,392); delay(300); tone(ses_out,440);
  delay(250); tone(ses_out,392); delay(500);
  tone(ses_out,349); delay(250); tone(ses_out,329);
  delay(250); tone(ses_out,293); delay(500);
  tone(ses_out,349); delay(250); tone(ses_out,392);
  delay(250); tone(ses_out,349); delay(500);
  tone(ses_out,329); delay(250); tone(ses_out,293);
  delay(250); tone(ses_out,261); delay(500);
  tone(ses_out,261); delay(300); tone(ses_out,293);
  delay(250); tone(ses_out,329); delay(250);
  tone(ses_out,293); delay(250); tone(ses_out,261);
  delay(500); tone(ses_out,293); delay(500);
  tone(ses_out,329); delay(500); tone(ses_out,293);
  delay(500); tone(ses_out,261); delay(500); noTone(ses_out); }
36 if (!digitalRead(octav))
37 {oktav++;
38 if(oktav>=8)
39 oktav=0;
40 lcd.clear();
41 lcd.print("PIYANO");
42 lcd.setCursor(8,0);
43 lcd.print(oktav);
44 lcd.setCursor(1,1);
45 lcd.print("OKTAV MODUNDA");
46 while (!digitalRead(octav));}
47 char gelen = keypad.getKey();
48 if (gelen)
49 {gelen=gelen-48;
50 nota=tones[gelen]<<(oktav);
51 lcd.clear();

20 #define sut4 pin_d0
21 #define sat1 pin_d4
22 #define sat2 pin_d5
23 #define sat3 pin_d6
24 #define sat4 pin_d7
25 #define ses_out pin_c3
26 #define test pin_a0
27 #define oktav pin_a1
28 #include <yt_lcd.c>
29 #include <TONES.c>
30 #include <math.h>
31 //değişkenler
32 long nota=0, gelen=0;
33 int oktav=0, tus=0,g_tus=0;
34 long tones []={0, 16, 17, 18, 19, 21, 22, 23, 25, 26, 28, 29, 31};
35 //alt programlar
36 char nota_oku()
37 {output_d(0x00);
38 nota = 0;
39 output_high(sat1);
40 if (input(sut1))
41 { delay_ms(1); tus=1;g_tus=1;} //NDO; }
42 if (input(sut2))
43 { delay_ms(1); tus=2;g_tus=1;} //NDO_DI; }
44 if (input(sut3))
45 { delay_ms(1); tus=3;g_tus=1;} //RE; }
46 output_low(sat1);
47 output_high(sat2);
48 if (input(sut1))
49 { delay_ms(1); tus=4;g_tus=1;} //RE_DI; }
50 if (input(sut2))
51 { delay_ms(1); tus=5;g_tus=1;} //MI; }
52 if (input(sut3))
53 { delay_ms(1); tus=6;g_tus=1;} //FA; }
54 output_low(sat2);
55 output_high(sat3);
56 if (input(sut1))
57 { delay_ms(1); tus=7;g_tus=1;} //FA_DI; }
58 if (input(sut2))
59 { delay_ms(1); tus=8;g_tus=1;} //SOL; }
60 if (input(sut3))
61 { delay_ms(1); tus=9;g_tus=1;} //SOL_DI; }
62 output_low(sat3);
63 output_high(sat4);
64 if (input(sut1))
65 { delay_ms(1); tus=10;g_tus=1;} //LA; }

```



```

52 lcd.print(oktav);
53 lcd.print(" OKTAV");
54 lcd.setCursor(1,1);
55 lcd.print("TON ");
56 lcd.print(nota);
57 lcd.print(" Hz");
58 tone(ses_out,nota);
59 delay(250);
60 noTone(ses_out);
61 gelen=0;}}

```

```

66 if (input(sut2))
67 { delay_ms(1);tus=11;g_tus=1;} //LA_DI; }
68 if (input(sut3))
69 { delay_ms(1); tus=12;g_tus=1;} //SI; }
70 output_low(sat4);
71 return tus; }
72 //genel ayarlar
73 void main ( )
74 {setup_psp(PSP_DISABLED);
75 setup_timer_1(T1_DISABLED);
76 setup_timer_2(T2_DISABLED,0,1);
77 setup_adc_ports(NO_ANALOGS);
78 setup_adc(ADC_OFF);
79 set_tris_d(0x0F);
80 set_tris_a(0x0F);
81 set_tris_c(0x00);
82 lcd_init();
83 lcd_send_byte(0, 0x0C);
84 lcd_putc("\f 12 TUSLU");
85 lcd_putc("\n 7 OKTAV PIYANO");
86 delay_ms(2000);
87 //ana program
88 while(1)
89 {if (!input(test))
90 {generate_tone(392,300);
generate_tone(440,250);
generate_tone(392,500);
generate_tone(349,250);
generate_tone(329,250);
generate_tone(293,500);
generate_tone(349,250);
generate_tone(392,250);
generate_tone(349,500);
generate_tone(329,250);
generate_tone(293,250);
generate_tone(261,500);
generate_tone(261,300);
generate_tone(293,250);
generate_tone(329,250);
generate_tone(293,250);
generate_tone(261,500);
generate_tone(293,500);
generate_tone(329,500);
generate_tone(293,500);

```

		<pre> generate_tone(261,500);} 91 if (!input(octav)) 92 {oktav++; 93 if(oktav>=8) 94 oktav=0; 95 printf(lcd_putc,"\f PIYANO %d",oktav); 96 lcd_putc("\n OKTAV MODUNDA"); 97 delay_ms(1000); 98 while (!input(octav));} 99 gelen = nota_oku(); 100 if (g_tus) 101 {nota=tones[gelen]<<(oktav); 102 printf(lcd_putc,"%f%d OKTAV ",oktav); 103 printf(lcd_putc,"\nTON %ld Hz",nota); 104 output_c(nota); 105 generate_tone(nota,250); 106 tus=0; 107 g_tus=0;}} </pre>
--	--	--

İşlem Basamakları

1. Uygun tipli mikrodeneleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonunu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodeneleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Elektronik piyano programında notaya ait ton nasıl elde edilir?
2. Oktava göre nota tonları kaç numaralı program satırında elde edilir?

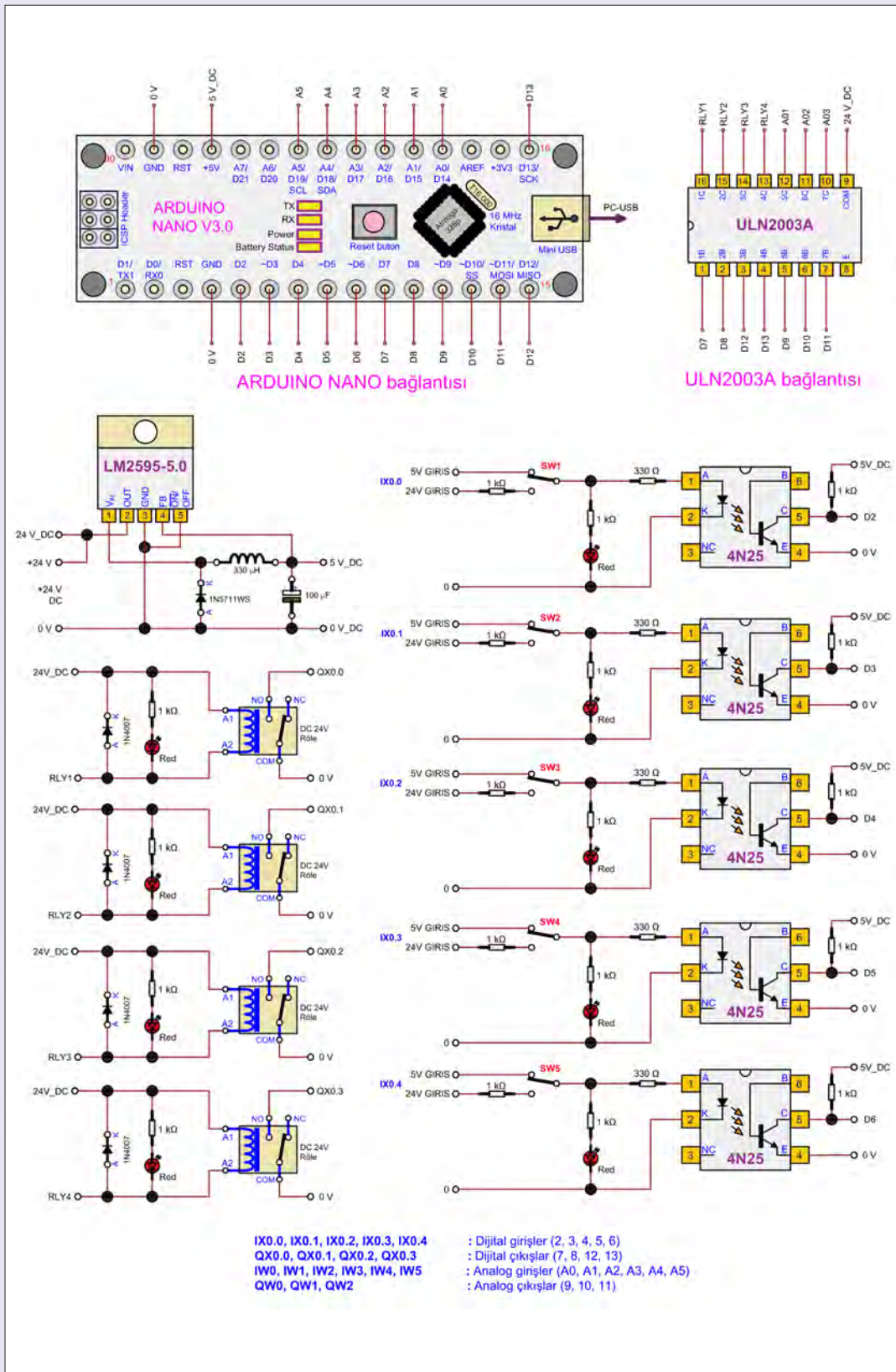
Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

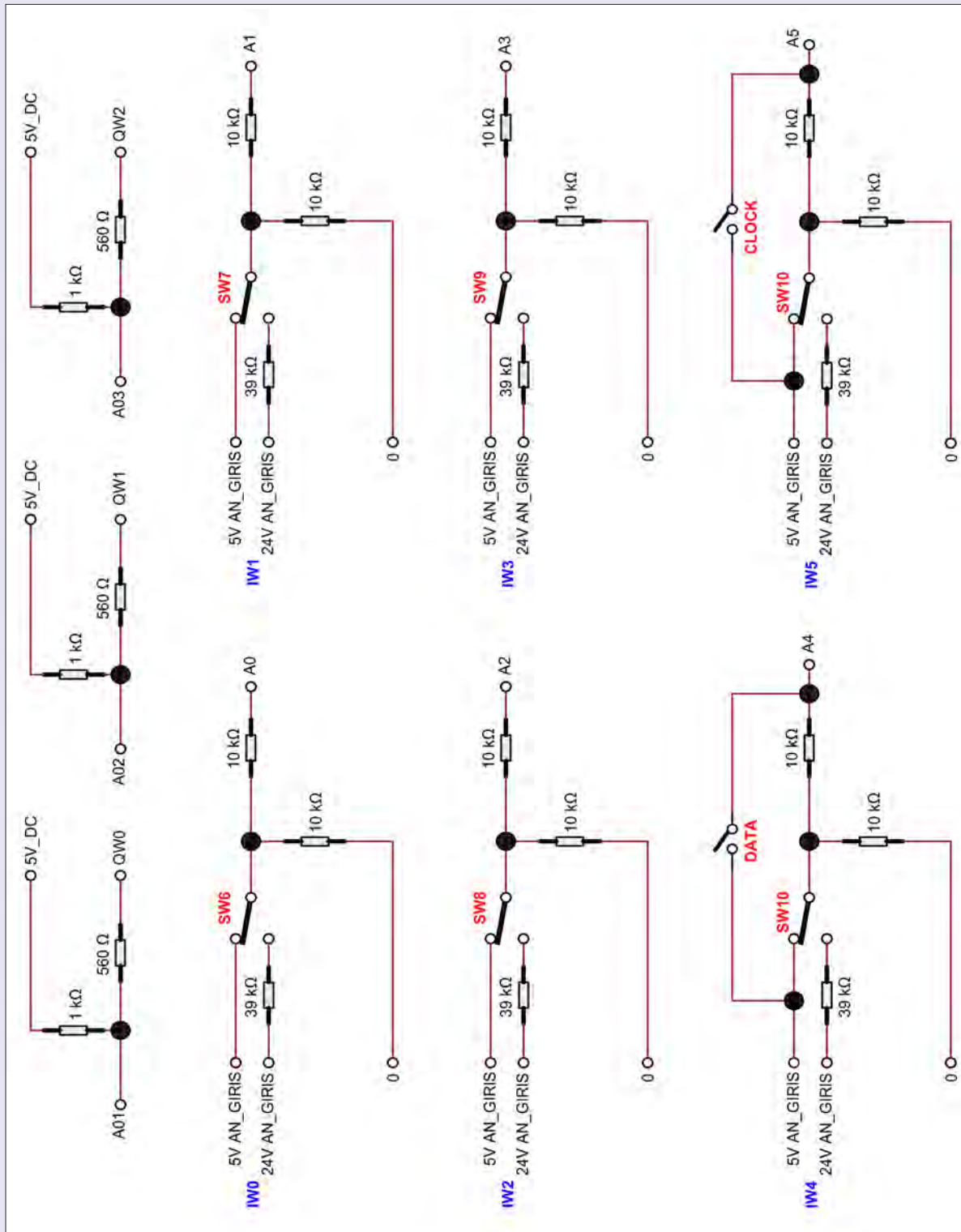


Devre Şeması: Görsel 1.52’de ARDUINO NANO ile PLC devre tasarımı görülmektedir.

1.16. Elektronik PLC Devresi



Görsel 1.52 (a): ARDUINO NANO ile PLC devresi



Görsel 1.52 (b): ARDUINO NANO ile PLC devresi

Devre Hakkında: ARDUINO NANO kart ve açık kaynak kodlu PLC programları kullanılarak ARDUINO NANO kartı PLC mantığıyla çalıştırmak mümkündür. Web üzerinden ARDUINO NANO kartı programlayan PLC programının deneme sürümü veya lisanlı sürümü indirilir, programın giriş ve çıkış port kullanımı durumuna göre ARDUINO NANO kart ile PLC tasarımı yapılır.



Malzemeler: Tablo 1.37’de iki devrenin malzeme listesi görülmektedir.

Tablo 1.37: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO NANO	Mikrodenetleyicili kart	1 adet
PLC programlayıcısı	Editör programı	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
ULN2003	Hat sürücüsü	1 adet
LM2595-5	Voltaj regülatörü	1 adet
4N25	Optokuplör	5 adet
Direnç	1 k Ω /0,25W	22 adet
Direnç	330 Ω /0,25W	5 adet
Direnç	560 Ω /0,25W	3 adet
Direnç	10 k Ω /0,25W	12 adet
Direnç	39 k Ω /0,25W	6 adet
Diyot	1N4007	5 adet
Diyot	1N5711WS	1 adet
Bobin	330 μ H	1 adet
Kondansatör	100 μ F elektrolitik	1 adet
LED diyot	10 mm kırmızı	9 adet
Röle	24 V DC	4 adet
Anahtar	On/off	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5V...24 V DC	1 adet
AC güç kaynağı		1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

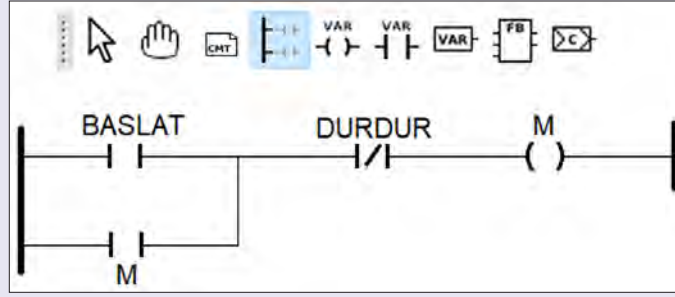
İstenenler

1. Verilen mikrodenetleyici ile PLC devresini breadboard veya plaket üzerine kurunuz.
2. Aşağıda verilen uygulamaları bu PLC üzerinde sırasıyla yaparak çalıştırınız.

1. Uygulama: ARDUINO PLC - AC Motorun Sürekli Çalıştırılması

PLC programında AC motor uygulaması şu adımlar doğrultusunda yapılır:

- Çalışma yapılan bilgisayarda masaüstünde veya diskte **AC_MOTOR** klasörü oluşturulur.
- PLC programlayıcısı çalıştırılır ve **Dosya** menüsünden **Yeni** komutu çalıştırılır.
- Açılan pencereden **AC_MOTOR** klasörü seçilir.
- Program araç çubuğundaki çizim araçları kullanılarak Görsel 1.53’te görülen ladder diyagramı çizilir.



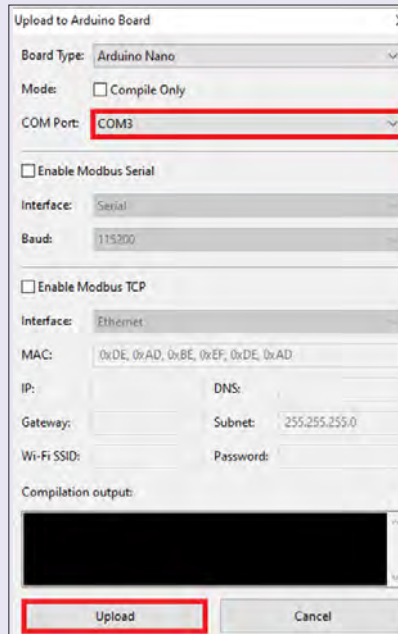
Görsel 1.53: AC motor ladder diyagramı

- Ladder diyagramında kullanılan giriş ve çıkış elemanlarının bağlı olduğu portlar ve port isimleri Görsel 1.54'te görüldüğü gibi düzenlenir.

Açıklama:		Sınıf filtre:	Herşey			
#	isim	Sınıf	tip	yer	Başlangıç değeri	belgeleme
1	BASLAT	Yerel	BOOL	%IX0.1		
2	DURDUR	Yerel	BOOL	%IX0.2		
3	M	Yerel	BOOL	%QX0.1		

Görsel 1.54: Giriş ve çıkış portları ve port isimlerinin düzenlenmesi

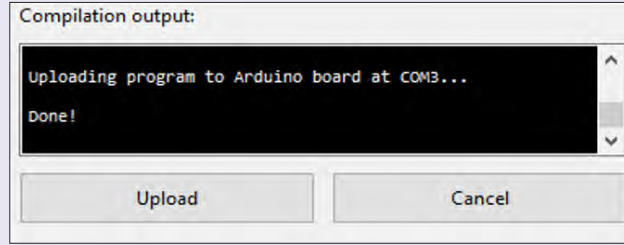
- Ladder diyagramın doğruluğunu test etmek için PLC programlayıcısında yükleme araç çubuğundaki **SIMULATION** (🔧📡🔄) düğmesine tıklanır. Program çalışıyorsa yükleme aşamasına geçilir. Program çalışmıyorsa ladder çizimi, giriş ve çıkış ayarları tekrar kontrol edilerek simülasyon yapılır.
- ARDUINO-PLC** kartı bilgisayara bağlanır ve ardından PLC programında yükleme araç çubuğundaki **ARDUINO** (🔧📡🔄) düğmesine tıklanır.
- Açılan port seçimi penceresinden **ARDUINO-PLC** kartın bulunduğu port seçilerek **Upload** butonuna tıklanır ve yükleme başlatılır (Görsel 1.55).



Görsel 1.55: ARDUINO-PLC karta programın yüklenmesi

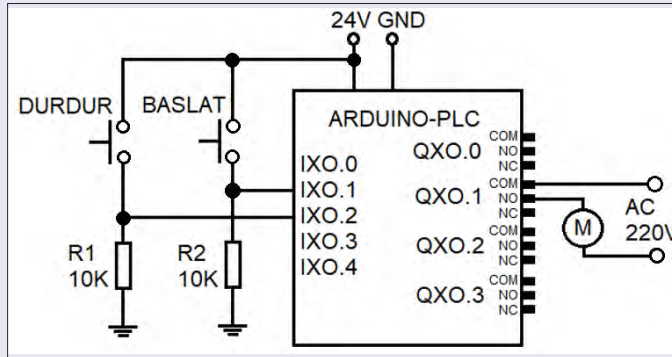


- Yükleme doğru yapılmışsa Görsel 1.56'da verilen uyarı görülecektir.



Görsel 1.56: ARDUINO-PLC karta yüklenen programın tamamlanması

- **ARDUINO-PLC** kartın bağlantıları Görsel 1.57'deki gibi yapılır ve program test edilir.



Görsel 1.57: ARDUINO-PLC devre bağlantısı

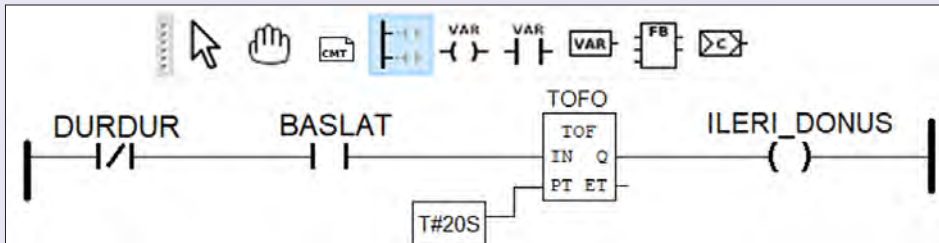
Sorular

1. Ladder diyagramında giriş ile çıkış port bağlantılarının yerleri değiştirdiğinde AC motor çalışır mı?
2. Uygulamadaki Ladder programı ile DC motor çalıştırılmak istenirse ARDUINO-PLC kartı üzerinde hangi değişikliklerin yapılması gerekir? Açıklayınız.

2. Uygulama: ARDUINO PLC-DC Motorun Sürekli Çalıştırılması

PLC programında DC motor uygulaması şu adımlar doğrultusunda yapılır:

- Çalışma yapılan alanda **DC_MOTOR** klasörü oluşturulur.
- PLC programlayıcısı çalıştırılır ve Dosya menüsünden **Yeni** komutu çalıştırılır.
- Açılan pencereden **DC_MOTOR** klasörü seçilir.
- Program araç çubuğundaki çizim araçları kullanılarak Görsel 1.58'de görülen ladder diyagramı çizilir.



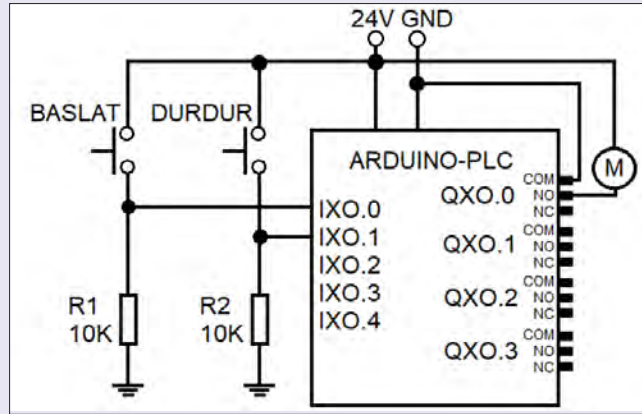
Görsel 1.58: DC motor sürekli çalıştırma ladder diyagramı

- Ladder diyagramında kullanılan giriş ve çıkış elemanlarının bağlı olduğu portlar ve port isimleri Görsel 1.59'da görüldüğü gibi düzenlenir.

Açıklama:	Sınıf filtre:	Herşey	+	-	↑	↓
#	isim	Sınıf	tip	yer	Başlangıç değeri	belgeleme
1	BASLAT	Yerel	BOOL	%IX0.0		
2	DURDUR	Yerel	BOOL	%IX0.1		
3	ILERI_DONUS	Yerel	BOOL	%QX0.0		
4	TOFO	Yerel	TOF			

Görsel 1.59: Giriş ve çıkış portları ve port isimlerinin düzenlenmesi

- Ladder diyagramın doğruluğunu test etmek için PLC programlayıcısında yükleme araç çubuğundaki **SIMULATION** () düğmesine tıklanır. Program çalışıyorsa yükleme aşamasına geçilir. Program çalışmıyorsa ladder çizimi, giriş ve çıkış ayarları tekrar kontrol edilerek simülasyon yapılır.
- ARDUINO-PLC** kartı bilgisayara bağlanır ve ardından PLC programında yükleme araç çubuğundaki **ARDUINO** () düğmesine tıklanır.
- Açılan port seçimi penceresinden **ARDUINO-PLC** kartın bulunduğu port seçilerek **Upload** butonuna tıklanır ve yükleme başlatılır.
- ARDUINO-PLC** kartın bağlantıları Görsel 1.60'taki gibi yapılır ve program test edilir.



Görsel 1.60: ARDUINO-PLC devre bağlantısı

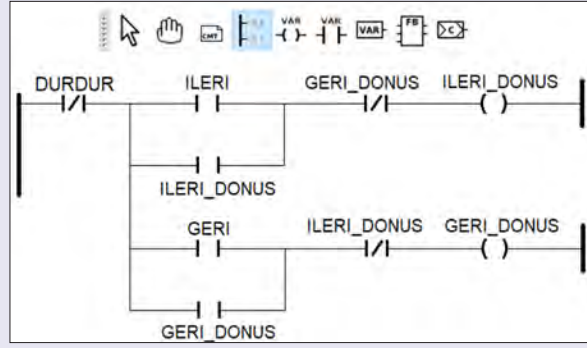
Sorular

- Aynı ladder diyagramı ile AC motor çalıştırılabilir mi?
- Uygulamadaki ladder programı ile AC motor çalıştırılmak istenirse ARDUINO UNO-PLC kartı üzerinde hangi değişikliklerin yapılması gerekir? Açıklayınız.

3. Uygulama: ARDUINO PLC-DC Motorun İleri-Geri Çalıştırılması

PLC programında DC motor uygulaması şu adımlar doğrultusunda yapılır:

- Çalışma yapılan alanda **DC_MOTOR_YON** klasörü oluşturulur.
- PLC programlayıcısı çalıştırılır ve **Dosya** menüsünden **Yeni** komutu çalıştırılır.
- Açılan pencereden **DC_MOTOR_YON** klasörü seçilir.
- Program araç çubuğundaki çizim araçları kullanılarak Görsel 1.61'de görülen ladder diyagramı çizilir.



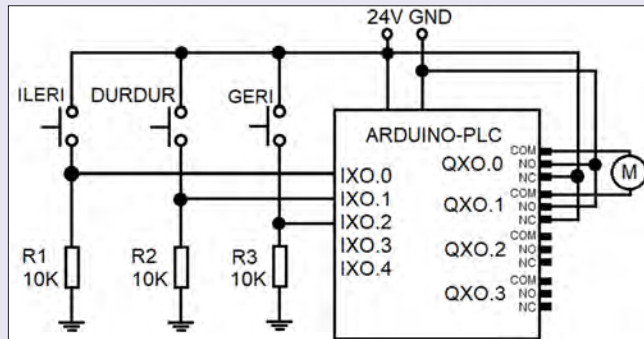
Görsel 1.61: DC motor yön değiştirme ladder diyagramı

- Ladder diyagramında kullanılan giriş ve çıkış elemanlarının bağlı olduğu portlar ve port isimleri Görsel 1.62’de görüldüğü gibi düzenlenir.

Açıklama:		Sınıf filtre:	Herşey			
#	isim	Sınıf	tip	yer	Başlangıç değeri	belgeleme
1	ILERI	Yerel	BOOL	%IX0.0		
2	DURDUR	Yerel	BOOL	%IX0.1		
3	GERI	Yerel	BOOL	%IX0.2		
4	ILERI_DONUS	Yerel	BOOL	%QX0.0		
5	GERI_DONUS	Yerel	BOOL	%QX0.1		

Görsel 1.62: Giriş ve çıkış portları ve port isimlerinin düzenlenmesi

- Ladder diyagramın doğruluğunu test etmek için PLC programlayıcısında yükleme araç çubuğundaki **SIMULATION** () düğmesine tıklanır. Program çalışıyorsa yükleme aşamasına geçilir. Program çalışmıyorsa ladder çizimi, giriş ve çıkış ayarları tekrar kontrol edilerek simülasyon yapılır.
- ARDUINO-PLC** kartı bilgisayara bağlanır ve ardından PLC programında yükleme araç çubuğundaki **ARDUINO** () düğmesine tıklanır.
- Açılan port seçimi penceresinden **ARDUINO-PLC** kartın bulunduğu port seçilerek **Upload** butonuna tıklanır ve yükleme başlatılır.
- ARDUINO-PLC** kartın bağlantıları Görsel 1.63’teki gibi yapılır ve program test edilir.



Görsel 1.63: ARDUINO-PLC devre bağlantısı

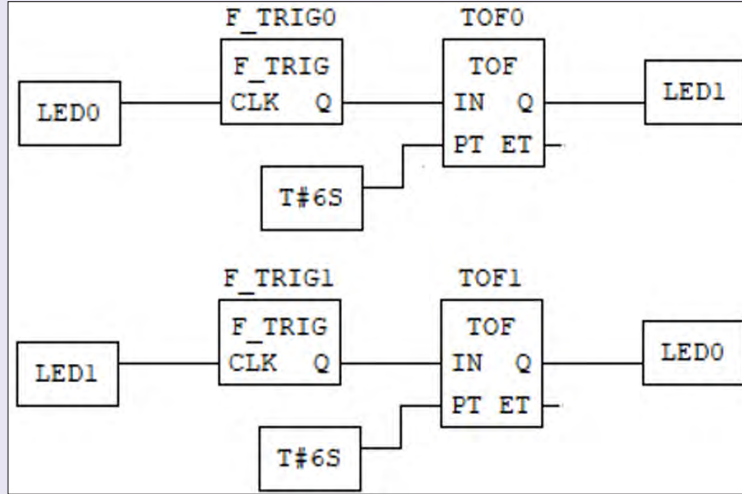
Sorular

- Aynı Ladder diyagramı ile AC motor ileri geri çalıştırılabilir mi?
- Uygulamadaki Ladder programı ile AC motor çalıştırılmak istenirse ARDUINO-PLC kartı üzerinde hangi değişikliklerin yapılması gerekir? Açıklayınız.

4. Uygulama: ARDUINO PLC-Flip Flop Uygulaması

PLC programında DC motor uygulaması şu adımlar doğrultusunda yapılır:

- Çalışma yapılan alanda **FLIPFLOP** klasörü oluşturulur.
- PLC programlayıcısı çalıştırılır ve **Dosya** menüsünden **Yeni** komutu çalıştırılır.
- Açılan pencereden **FLIPFLOP** klasörü seçilir.
- Program araç çubuğundaki çizim araçları kullanılarak Görsel 1.64'te görülen FBD diyagramı çizilir.



Görsel 1.64: Flip-flop FBD diyagramı

- FBD diyagramında kullanılan giriş ve çıkış elemanlarının bağlı olduğu portlar ve port isimleri Görsel 1.65'te görüldüğü gibi düzenlenir.

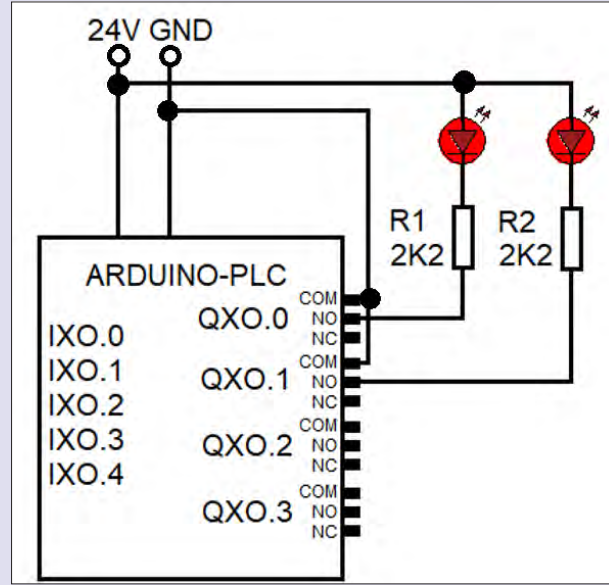
Açıklama:		Sınıf filtre:	Herşey			
#	isim	Sınıf	tip	yer	Başlangıç değeri	belgeleme
1	LED0	Yerel	BOOL	%QX0.0		
2	LED1	Yerel	BOOL	%QX0.1		
3	F_TRIG0	Yerel	F_TRIG			
4	TOF0	Yerel	TOF			
5	F_TRIG1	Yerel	F_TRIG			
6	TOF1	Yerel	TOF			

Görsel 1.65: Giriş ve çıkış portları ve port isimlerinin düzenlenmesi

- Ladder diyagramın doğruluğunu test etmek için PLC programlayıcısında yükleme araç çubuğundaki **SIMULATION** () düğmesine tıklanır. Program çalışıyorsa yükleme aşamasına geçilir. Program çalışmıyorsa ladder çizimi, giriş ve çıkış ayarları tekrar kontrol edilerek simülasyon yapılır.
- **ARDUINO-PLC** kartı bilgisayara bağlanır ve ardından PLC programında yükleme araç çubuğundaki **ARDUINO** () düğmesine tıklanır.
- Açılan port seçimi penceresinden **ARDUINO-PLC** kartın bulunduğu port seçilerek **Upload** butonuna tıklanır ve yükleme başlatılır.



- ARDUINO-PLC kartın bağlantıları Görsel 1.66'daki gibi yapılır ve program test edilir.



Görsel 1.66: ARDUINO UNO-PLC devre bağlantısı

Sorular

1. FBD diyagramda **F_TRIG0** ve **F_TRIG1**'in silinmesi durumunda PLC'de nasıl değişiklik olur?
2. FBD diyagramda kullanılan **TOF0** ve **TOF1**'in görevleri nedir? Açıklayınız.



SIRA SİZDE

PIC16F877 entegresini kullanarak Elektronik PLC devresi kurunuz. Bu devre üzerinde AC motorun sürekli, ileri-geri ve zaman ayarlı çalışması için gerekli devreleri kurarak devrenin çalışmalarını test ediniz. Sonuçları değerlendiriniz.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



1. ÖĞRENME BİRİMİ: UYGULAMA DEĞERLENDİRME TABLOSU

Aşağıda uygulama değerlendirme ölçütleri (Tablo 1.38) görülmektedir. Birinci öğrenme biriminde yapılan uygulama çalışmalarını bu ölçütlere göre değerlendiriniz. Her ölçütün değerlendirme sonucunu karşısındaki puanlama tablosunda işaretleyiniz (V) ve tablodan alınan toplam puanı hesaplayınız.

Tablo 1.38: Uygulama Değerlendirme Tablosu

Değerlendirme Ölçütleri		Puanlama Ölçütleri					
		Kötü (0)	Yeterli (1)	Orta (2)	İyi (3)	Çok İyi (4)	Mükemmel (5)
1.	Projede kullanılacak mikrodenetleyiciyi belirlendi.						
2.	Proje kodları uygun yazılım editöründe oluşturuldu.						
3.	Proje kodları derlendi ve kontrol edildi.						
4.	Proje kodları mikrodenetleyiciye yüklendi.						
5.	Proje devre tasarımı şema çizim programında çizildi.						
6.	Proje devresinin simülasyonunu yapıldı.						
7.	Projenin kurulacağı alan (breadboard, PCB vb.) belirlendi.						
8.	Projenin baskı devresinin tasarımı ya da breadboard kurulumu yapıldı.						
9.	Projedeki entegrelerin ve elemanların kullanılacak uçları kataloglar yardımıyla belirlendi.						
10.	Projede kullanılan elemanların sağlamlık kontrolleri yapıldı.						
11.	Proje malzemeleri devreye uygun şekilde yerleştirildi.						
12.	Projeye uygun kablolar belirlendi.						
13.	Proje kablo bağlantıları doğru yapıldı.						
14.	Projede, belirlenen parçalar kullanıldı.						
15.	Çalışmanın gerçekleştirilme aşamaları düzenli olarak kontrol edildi.						
16.	Projede oluşturulan devre istenilen şekilde çalıştı.						
17.	Projeye oluşturulan devrenin çalışması detaylı olarak incelendi.						
18.	Uygulamalar verilen sürede tamamlandı						
19.	Proje, zamanında teslim edildi.						
20.	Proje raporu eksiksiz teslim edildi.						
TOPLAM							
Değerlendirme sonucunda 100 puan üzerinden toplam puan almıştır.							
Öğrencinin Adı-Soyadı:							
Sınıfı-Numarası:							
İmza:							
Öğretmenin Adı-Soyadı:							
İmza:							

2. ÖĞRENME BİRİMİ

MİKRODENETLEYİCİ İLE ANALOG UYGULAMALAR

KONULAR

- 2.1. Dijital Termometre Devresi
- 2.2. Mesafe Ölçer Devresi
- 2.3. Gaz Dedektörü Devresi
- 2.4. Motor Hız Kontrol Devresi
- 2.5. Dijital Voltmetre Devresi

NELER ÖĞRENECEKSİNİZ?

- Farklı mikroişlemcileri kullanarak çeşitli analog devreleri breadboard üzerinde kurma ve çalıştırma
- Farklı mikroişlemcileri kullanarak çeşitli analog devreleri baskı devre plaketine yapma ve çalıştırma
- Farklı projeler içinden uygun projeleri seçme ve uygulama

HAZIRLIK ÇALIŞMALARI

- Bir cismin sıcaklığını ölçme yöntemlerini araştırınız.
- DC motorun hızı çalışma sırasında değiştirilebilir mi? Fikirlerinizi sınıfta arkadaşlarınızla paylaşınız.

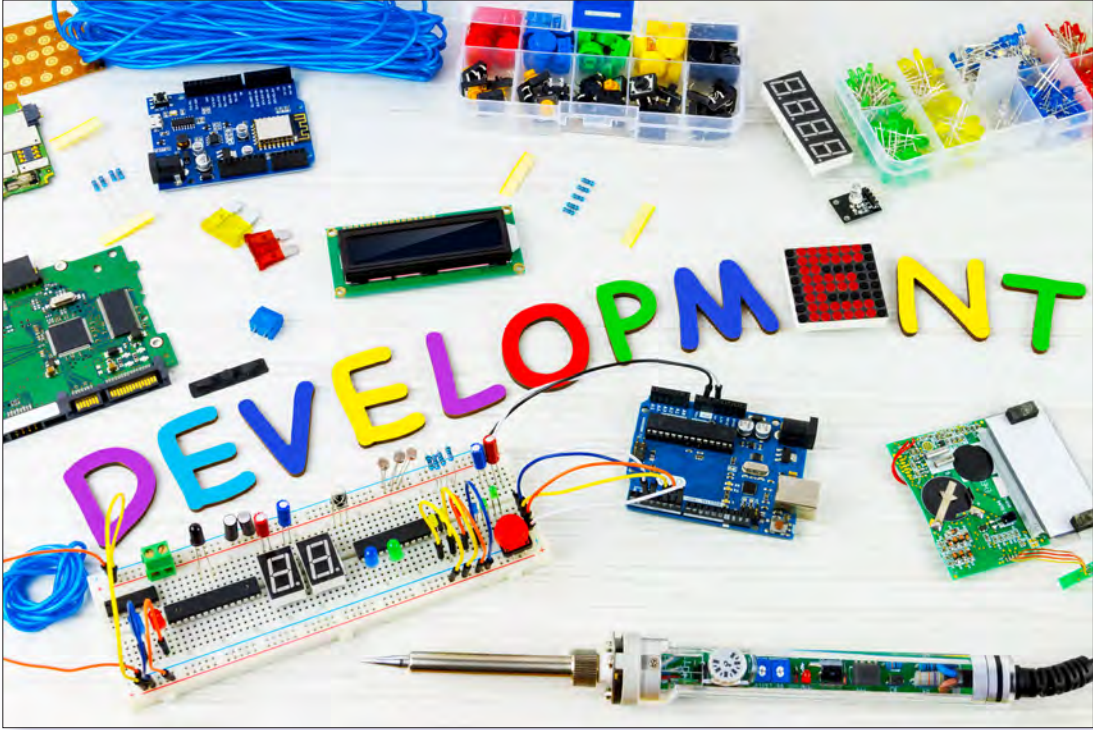
TEMEL KAVRAMLAR

ARDUINO, ARM32 mikroişlemciler, breadboard, el aletleri, elektronik devre, elektronik devre elemanları (direnc, display, sensör vb.), mikroişlemci yazılımları, ölçü aleti, PIC, plaket



2. MİKRODENETLEYİCİ İLE ANALOG UYGULAMALAR

Görsel 2.1’de ARDUINO UNO ile devre tasarımları, Görsel 2.2’de ARDUINO UNO kartına yazılım yüklenmesi ve devrenin çalıştırılması görülmektedir.



Görsel 2.1: ARDUINO UNO ile devre tasarımları



Görsel 2.2: ARDUINO UNO kartına yazılım yüklenmesi ve devrenin çalıştırılması

Bu öğrenme biriminde ARDUINO UNO ile PIC mikrodeneleyiciler kullanılarak çeşitli analog uygulamalar gerçekleştirilecektir. Bu doğrultuda aşağıda verilen uygulama yönergesini inceleyiniz ve çalışmaları yönergeye uygun gerçekleştiriniz.



UYGULAMA YÖNERGESİ

Bu öğrenme birimine ait beş farklı uygulama bulunmaktadır. Uygulamalar tablosunda üç farklı mikroişlemci modeline göre verilen projeler görülmektedir. Uygulamaları aşağıdaki yönerge doğrultusunda gerçekleştiriniz.

- İlgili uygulamada kullanılacak mikroişlemci modelini belirleyiniz.
- Uygulamayı gerçekleştirecek grubu (1 ila 6 kişi) oluşturunuz. Grup üyelerinin görevlerini belirleyiniz.
- Belirlenen mikroişlemci modeli için verilen uygulama devresinin simülasyonunu elektronik simülasyon yazılımı ile yapınız.
- Belirlenen mikroişlemci modeli için verilen uygulama devresini breadboard ya da baskı devre plaketi üzerine kurunuz. Bu doğrultuda malzeme listesi oluşturunuz ve malzemelerin sağlamlıklarını kontrol ediniz.
- Devrenin kodlarını ilgili editörde oluşturup derleyiniz ve mikroişlemciye yükleyiniz.
- Devrenin simülasyon ve programlama dosyalarını uygun isimde kaydediniz ve öğretmenimize teslim ediniz.
- İşlem sonunda devrenin raporunu öğretmeninizin yönergesi doğrultusunda (soruları da cevaplayarak) hazırlayıp öğretmenimize teslim ediniz.

UYGULAMALAR

No.	Uygulama Adı	PIC Serisi	ARDUINO Serisi
01	Dijital Termometre Devresi	16F628A	UNO kartı
02	Mesafe Ölçer Devresi	16F877A	UNO kartı
03	Gaz Dedektörü Devresi	16F876	UNO kartı
04	Motor Hız Kontrol Devresi	16F873A	UNO kartı
05	Dijital Voltmetre Devresi	16F873A	UNO kartı



DİKKAT!

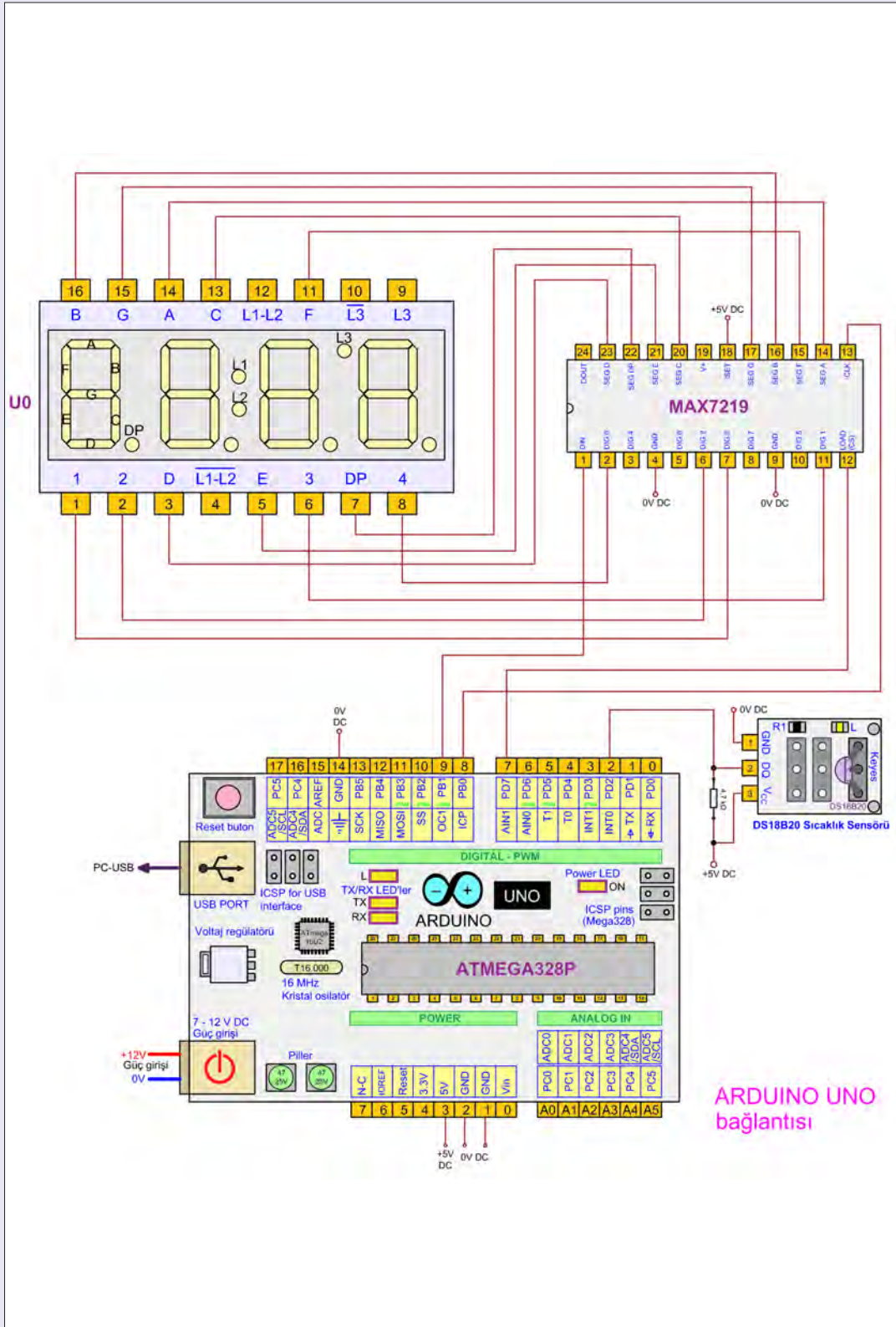
- Uygulama ile ilgili tüm dosyalara yanda verilen karekoddan ulaşabilirsiniz.
- Ekteki kısımdan devrelerde kullanılan elemanların ayak bağlantılarına, baskı devre şemalarına ve detaylı teknik bilgilerine ulaşabilirsiniz.



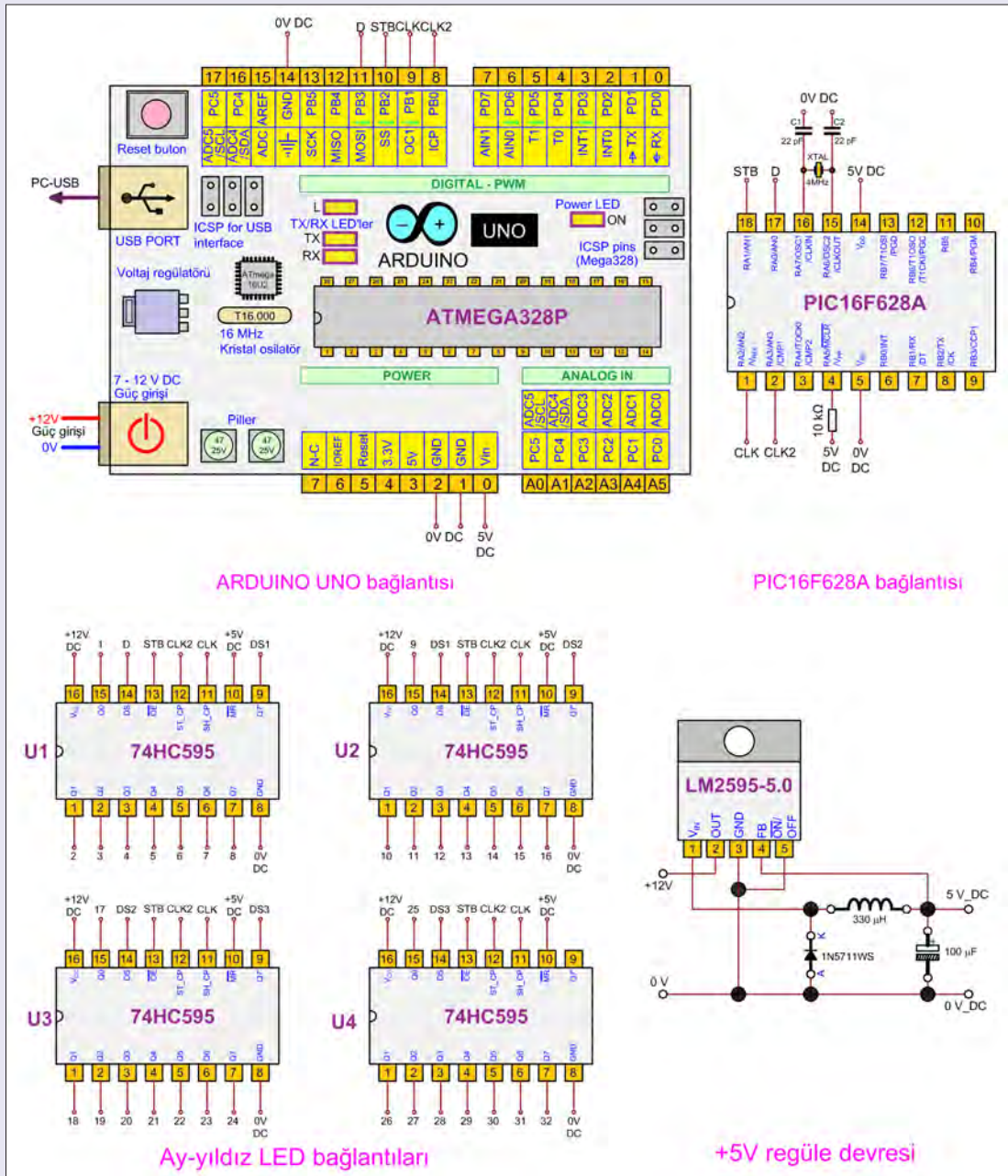
2.1. Dijital Termometre Devresi



Devre Şeması: Görsel 2.3'te ARDUINO UNO ile dijital termometre devresi, Görsel 2.4'te PIC16F628A entegresi ile dijital termometre bağlantı devresi görülmektedir.



Görsel 2.3: ARDUINO UNO ile dijital termometre devresi



Görsel 2.4: PIC16F628A entegresi ile dijital termometre devresi

Devre Hakkında: Dijital termometre devresi, DS18B20 sıcaklık sensörü kullanılarak tasarlanmıştır. DS18B20 sıcaklık sensörü, ölçtüğü sıcaklık verisini dijital seri veri formatında mikrodenetleyiciye gönderir. Mikrodenetleyici, gelen veri üzerinde hesaplama yaparak sıcaklık değerini ekranda gösterir. DS18B20 sıcaklık sensörü kütüphanesi, MAX7219 entegresi kütüphanesi ve diğer kütüphaneler programa dâhil edilir. DS18B20 sıcaklık sensörü iletişim portu ve ekran portları ayarları yapılır. Display ekranı temizlenir ve parlaklık ayarı yapılır. DS18B20 sensörüne okuma isteği gönderilir ve sıcaklık değeri okunur. Okunan dijital veri üzerinde hesaplama yapılır. Hesaplama sonucu elde edilen onluk ve birlik sayısal değerler 4 digit 7 segment display ekranında yazılım olarak kodlandıktan sonra gösterilir. Ekranda gösterme işlemi şu şekilde gerçekleşir: Okunan ısı değerinin onlar basamağı 3. dijitte, birler basamağı 2. dijitte, ° 1. dijitte, (C) harfi 0. dijitte gösterilir. Ölçülen sıcaklık değeri 44.3 ise 44'e, 44.6 ise 45'e yuvarlanır (Görsel 2.3-2.4).



Malzemeler: Tablo 2.1’de iki devrenin malzeme listesi görülmektedir.

Tablo 2.1: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F628A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaketa	Yeterli ölçülerde	-
MAX7219	LED sürücü	1 adet
4-Digit display	Ortak katot	1 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	+5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli miktarda	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 2.2’de iki devrenin kod yazılımları görülmektedir.

Tablo 2.2: Devrenin Kodları

No.	Kodlar (ARDUINO UNO)	No.	Kodlar (PIC16F628A)
1	//kütüphaneler ve tanımlamalar	1	//kütüphaneler ve tanımlamalar
2	#include <SPI.h>	2	#include <16F628A.H>
3	#include <Adafruit_GFX.h>	3	#fuses XT, NOWDT, NOPROTECT, BROWNOUT, PUT, NOLVP
4	#include <LedControl.h>	4	#use delay(clock=4000000)
5	#include <OneWire.h>	5	#include<1wire.c>
6	#include <DallasTemperature.h>	6	#include<ds1820.c>
7	#define ONE_WIRE_BUS 2	7	#define ONE_WIRE_PIN PIN_A3
8	//değişkenler	8	#define DATA PIN_B0
9	int derece, derece1;	9	#define LOAD PIN_B1
10	LedControl yt = LedControl(9, 8, 7, 1);	10	#define CLK PIN_B2
11	char coden[]={0x7E, 0x30, 0x6D, 0x79, 0x33, 0x5B, 0x5F, 0x70, 0x7F, 0x73}; //KATOT	11	#include<MAX7219.c>
12	OneWire oneWire(ONE_WIRE_BUS);	12	#use fast_io(b)
13	DallasTemperature sensors(&oneWire);	13	#use fast_io(a)
14	//alt programlar	14	//değişkenler
15	void ISI()	15	char dereceon;
16	{sensors.requestTemperatures();	16	const int digit[10]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7c, 0x07, 0x7f, 0x6f};
17	derece = sensors.getTempCByIndex(0);	17	float derece;
18	derece1= round(derece);	18	unsigned int derecebir;
19	yt.setRow(0,3,coden[derece1/10]);	19	int data1[4];
20	yt.setRow(0,2,coden[derece1%10]);	20	//alt programlar
21	yt.setRow(0,1,0x63);	21	void ekran()
22	yt.setRow(0,0,0x4E);}	22	{data1[0]=(digit[dereceon]);

23	//genel ayarlar	23	data1[1]=(digit[derecebir]);
24	void setup() {	24	data1[2]=(0b01100011);
25	pinMode(ONE_WIRE_BUS, INPUT_PULLUP);	25	data1[3]=(0b00111001);
26	yt.shutdown(0, false);	26	ekranda_goster(data1,0);
27	yt.setIntensity(0, 5);	27	//genel ayarlar
28	yt.clearDisplay(0);	28	void main()
29	sensors.requestTemperatures();	29	{setup_timer_1(T1_DISABLED);
30	derece = sensors.getTempCByIndex(0);	30	setup_CCP1(CCP_OFF);
31	delay(1000);}	31	set_tris_a(0b00001000);
32	//ana program	32	set_tris_b(0b00110000);
33	void loop() {	33	ekran_ayar(0);
34	ISI();	34	//ana program
35	delay(1000);}	35	while(1)
		36	{derece = ds1820_read();
		37	derece/=8;
		38	dereceon=derece/10;
		39	derecebir=(unsigned int)derece%10;
		40	ekran();}}

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Sayısal değerleri display ekran için kodlayan yazılım satırı, ARDUINO UNO ve PIC16F628A için kaç numaralı satırlarda yer almaktadır?
2. Kesirli (float) sayıları tam sayılara çeviren yazılım satırı, ARDUINO UNO ve PIC16F628A için kaç numaralı satırlarda yer almaktadır?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

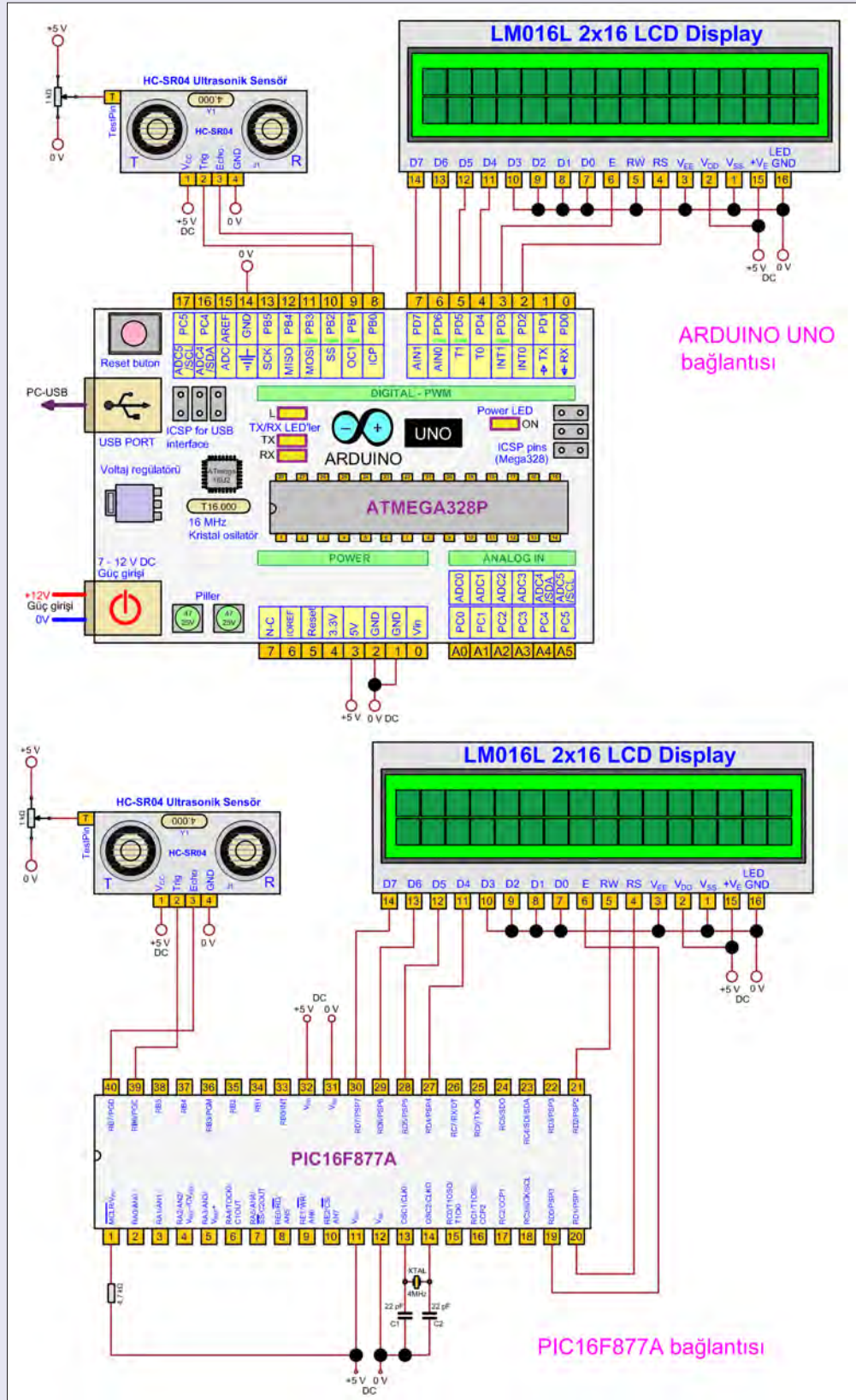
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



2.2. Mesafe Ölçer Devresi



Devre Şeması: Görsel 2.5'te mesafe ölçer devresi görülmektedir.



Görsel 2.5: Mesafe ölçer devresi

Devre Hakkında: Mesafe ölçer devre uygulaması HC-SR04 ultrasonik mesafe sensörü kullanılarak tasarlanmıştır. HC-SR04 ultrasonik mesafe sensörünün yapısında bir adet ultrasonik ses sinyali göndericisi (transmitter) ve bir adet ultrasonik ses sinyali alıcısı (receiver) bulunur. HC-SR04, 2-400 cm aralığındaki mesafeyi ölçmede kullanılır. Mikrodenetleyici, gönderilen ultrasonik ses dalgası ile yansıyıp gelen ultrasonik ses dalgası arasındaki süreyi ölçer ve gelen veri üzerinde hesaplama yaparak ölçülen mesafeyi ekranda gösterir. LCD ekran kütüphanesi ve diğer kütüphaneler programa dâhil edilir. LCD ekran iletişim portu ve HC-SR04 trig pini çıkış (OUTPUT), echo pini giriş (INPUT) olarak ayarlanır. Mikrodenetleyici tarafından HC-SR04 mesafe sensörünün trig pinine 20 µs tetikleme sinyali uygulanır. HC-SR04 mesafe sensörü tetikleme sinyalini algıladığında ultrasonik ses dalgası gönderir. Gönderilen ultrasonik ses dalgası cisimden yansıyıp geldiğinde HC-SR04 echo pini lojik 1 (5V) olur. Mikrodenetleyici, HC-SR04 echo pini lojik 1 olduğunda sayıcıyı çalıştırır ve echo pini lojik 0 (0V) olduğunda sayıcı değerini alır. Sesin havada yayılma hızı formülüyle geçen süre dikkate alınarak mesafe hesaplanır. Ölçülen mesafe LCD ekranına mm cinsinden yazdırılır (Görsel 2.5).

Malzemeler: Tablo 2.3'te iki devrenin malzeme listesi görülmektedir.

Tablo 2.3: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F877A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
HC-SR04	Ultrasonik sensör	1 adet
LCD display	LM016L 2x16	1 adet
Direnç	1 kΩ potansiyometre	1 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	+5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli ölçülerde	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 2.4'de iki devrenin kod yazılımları görülmektedir.

Tablo 2.4: Devrenin Kodları

No.	Kodlar (ARDUINO UNO)	No.	Kodlar (PIC16F877A)
1	//kütüphane	1	//kütüphaneler ve tanımlamalar
2	#include <LiquidCrystal.h>	2	#include <16F877A.h>
3	LiquidCrystal lcd(2, 3, 4, 5, 6, 7); // (rs, enable, d4, d5, d6, d7)	3	#FUSES XT, NOWDT, NOPUT, noPROTECT, NOBROWNOUT, NOLVP, NOCPD, NOWRT, NODEBUG
4	//değişkenler	4	#use delay(clock=4000000)
5	const int trigPin = 8;	5	#define use_portd_lcd TRUE
6	const int echoPin = 9;	6	#include <lcd.c>
7	long sure;	7	#define trig pin_B6
8	int mesafe;	8	#define echo pin_B7
9	//genel ayarlar	9	//değişkenler
10	void setup() {	10	int16 mesafe, sure;
11	lcd.begin(16,2);		



12	pinMode(trigPin, OUTPUT);	11	void main()
13	pinMode(echoPin, INPUT);}	12	{lcd_init();
14	//ana program	13	delay_ms(100);
15	void loop() {	14	printf(LCD_PUTC, "\f SONAR TEST \n İLERİ MIKRO"); delay_ms(1000);
16	digitalWrite(trigPin, LOW);	15	setup_timer_1(T1_INTERNAL T1_DIV_BY_8);
17	delayMicroseconds(2);	16	//ana program
18	digitalWrite(trigPin, HIGH);	17	while(true)
19	delayMicroseconds(10);	18	{output_high(trig);
20	digitalWrite(trigPin, LOW);	19	delay_us(20);
21	sure = pulseIn(echoPin, HIGH);	20	while(1)
22	mesafe= sure*0.34/2;	21	output_low(trig);
23	lcd.setCursor(0,0);	22	while(!input(ECHO))
24	lcd.print("OLCULEN MESAFE:");	23	set_timer1(0);
25	lcd.setCursor(0,1);	24	while(input(ECHO))
26	lcd.print(mesafe);	25	sure=get_timer1();
27	lcd.print(" mm");	26	mesafe=sure*1.35;
28	delay(10);}	27	printf(LCD_PUTC, "\f OLCULEN MESAFE \n %Lu mm", mesafe);
		28	delay_ms(1000);}}

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. LCD ekran port ayar satırı, ARDUINO UNO ve PIC16F877A için kaç numaralı yazılım satırlarında yer almaktadır?
2. Gönderilen ve alınan ultrasonik ses dalgaları arasındaki süreyi ölçen yazılım satırı, ARDUINO UNO ve PIC16f628A için kaç numaralı satırlarda yer almaktadır?

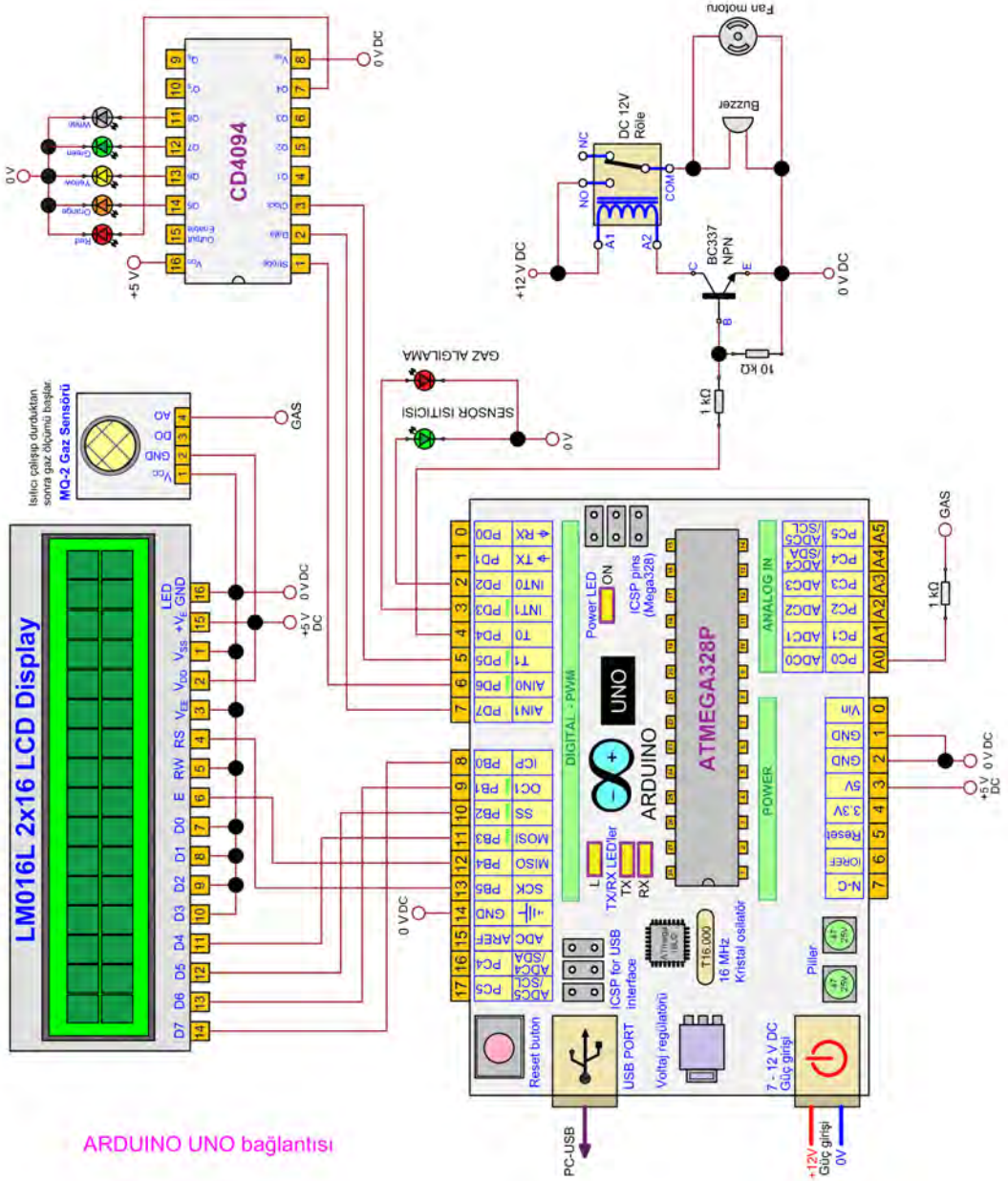
Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

2.3. Gaz Dedektörü Devresi



Devre Şeması: Görsel 2.6'da ARDUINO ile gaz dedektör devresi, Görsel 2.7'de PIC16F876A ile gaz dedektör devresi görülmektedir.



Görsel 2.6: ARDUINO UNO ile gaz dedektörü devresi

Devre Hakkında: MGaz dedektörü devre uygulaması, MQ-6 gaz sensörü kullanılarak tasarlanmıştır. MQ-6 gaz sensörü 300 ila 10.000 PPM konsantrasyonlarda yanıcı gazı (LPG/izobütan/propan) algılayan sensördür. MQ-6 gaz sensörü tarafından algılanan gaz konsantrasyonu arttıkça analog çıkış voltajı artar. Gaz konsantrasyonu azaldıkça çıkış voltajı da azalır. Mikrodenetleyici, MQ-6 gaz sensöründen gelen analog veri üzerinde hesaplama yaparak gaz durumunu ekranda gösterir. LCD ekran, SPI iletişim kütüphanesi, diğer kütüphaneler programa dâhil edilir. LCD ekran iletişim portu, MQ-6 A0 pini giriş, kullanılan diğer portlar çıkış olarak ayarlanır. MQ-6 gaz sensörünün ölçme işlemini yapabilmesi için yapısında bulunan ısıtıcısının 5 saniye ısıtılması beklenir (Sensör ısıtıcı ledi 5 kez yanıp söner.). Display ekranında “GAZ SENSORU ISINIYOR, LUTFEN BEKLEYİNİZ.” yazısı görülür ve analog gaz ölçümü yapılır. 10 bitlik bilgi 8 bitliğe dönüştürülür.

- Okunan veri “bilgi<40” ise ekranda “GAZ YOGUNLUGU 1. SEVIYE” yazısı görülür, beyaz led yanar.
- Okunan veri “bilgi>40 && bilgi<80” ise ekranda “GAZ YOGUNLUGU 2. SEVIYE” yazısı görülür, beyaz ve yeşil led yanar.
- Okunan veri “bilgi>80 && bilgi<120” ise ekranda “GAZ YOGUNLUGU 3. SEVIYE” yazısı görülür; beyaz, yeşil ve sarı led yanar.
- Okunan veri “bilgi>120 && bilgi<160” ise ekranda “GAZ YOGUNLUGU 4. SEVIYE” yazısı görülür; beyaz, yeşil, sarı ve turuncu led yanar.
- Okunan veri “bilgi>160 && bilgi<200” ise ekranda “GAZ YOGUNLUGU 5. SEVIYE” yazısı görülür; beyaz, yeşil, sarı, turuncu ve kırmızı led yanar.
- Okunan veri “bilgi>200” ise ekranda “GAZ YOGUNLUGU COK FAZLA ALARM DEVREDE” yazısı görülür; beyaz, yeşil, sarı, turuncu ve kırmızı led yanarken röle çeker, buzzer uyarı sesi verir ve fan motoru çalışarak gazı dağıtır.

Malzemeler: Tablo 2.5’te iki devrenin malzeme listesi görülmektedir.

Tablo 2.5: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyici kart	1 adet
PIC16F876	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
LCD display	LM016L 2x16	1 adet
MQ2	Gaz sensörü	1 adet
Röle	12 V DC	1 adet
Buzzer	12 V DC	1 adet
Led	5 mm çeşitli renklerde	7 adet
Fan motoru	12 V DC	1 adet
BC337	Transistör	1 adet
Direnç	10 kΩ, 1 kΩ	1’er adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Direnç	4.7 kΩ, 1 kΩ	1’er adet
CD4094	Shift register, 3 durum çıkışlı	1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	+5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli ölçülerde	Birer adet
Matkap ve uçları		-



Kod Yazılımları: Tablo 2.6’da iki devrenin kod yazılımları görülmektedir.

Tablo 2.6: Devrenin Kodları

No.	Kodlar (ARDUINO UNO)	No.	Kodlar (PIC16F876A)
1	//kütüphaneler	1	//kütüphaneler ve tanımlamalar
2	#include <LiquidCrystal.h>	2	#include <16f876A.h>
3	LiquidCrystal lcd(13, 12, 11, 10, 9, 8); // LCD bağlantısı:	3	#device ADC=8
	(rs, enable, d4, d5, d6, d7)	4	#fuses XT, NOWDT, NOPROTECT, NOBROWNOUT, NOLVP, NO
4	#include <SPI.h>		PUT, NOWRT, NODEBUG, NOCPD
5	//değişkenler	5	#use delay(clock=4000000)
6	const int ds = 7;	6	#define use_portb_lcd TRUE
7	const int latch = 6;	7	#include <lcd.c>
8	const int clk = 5;	8	#use fast_io(a)
9	const int alarm = 4;	9	#use fast_io(b)
10	const int gaz_algilama = 3;	10	#use fast_io(c)
11	const int isitici_p = 2;	11	#define ds pin_c0
12	const int gaz_sensor = A0;	12	#define latch pin_c1
13	unsigned int sayi1=0, sayi2=0, sayi3=0, sayi4=0,	13	#define clock pin_c2
	sayi5=0, sayialarm=0, isitici;	14	#define ALARM_OUT pin_c5
14	unsigned long int bilgi;	15	//değişkenler
15	//alt programlar	16	unsigned int8 sayi1=0, sayi2=0, sayi3=0, sayi4=0, sayi5=0, sa
16	void seviye_yaz(int seviye)		yialarm=0, isitici;
17	{int j=0;	17	unsigned long int bilgi;
18	for(j=0;j<8;j++)	18	//alt programlar
19	{shiftOut(ds, clk, LSBFIRST, seviye);}	19	void seviye_yaz(int seviye)
20	digitalWrite(latch,HIGH);	20	{int j=0;
21	delayMicroseconds(1);	21	for(j=0;j<8;j++)
22	digitalWrite(latch,LOW);}	22	{output_bit(ds,bit_test(seviye,j));
23	//genel ayarlar	23	output_high(clock);
24	void setup()	24	delay_us(1);
25	{ pinMode(ds, OUTPUT);	25	output_low(clock);}
26	pinMode(latch, OUTPUT);	26	output_high(latch);
27	pinMode(clk, OUTPUT);	27	delay_us(1);
28	pinMode(alarm, OUTPUT);	28	output_low(latch);}
29	pinMode(gaz_algilama, OUTPUT);	29	void buzzer()
30	pinMode(isitici_p, OUTPUT);	30	{output_high(ALARM_OUT);
31	lcd.begin(16, 2);	31	delay_ms(100);
32	lcd.clear();	32	output_low(ALARM_OUT);}
33	lcd.setCursor(0,0);	33	//genel ayarlar
34	lcd.print("GAZ SENSORU UYG.");	34	void main ()
35	lcd.setCursor(0,1);	35	{setup_timer_1(T1_DISABLED);
36	lcd.print("İLERİ MIKRO");	36	setup_timer_2(T2_DISABLED,0,1);
37	delay(1000);	37	setup CCP1(CCP_OFF);
38	lcd.clear();	38	setup CCP2(CCP_OFF);
39	lcd.setCursor(0,0);	39	set_tris_c(0x00);
40	lcd.print("GAZ SENSORU ISINIYOR");	40	set_tris_a(0x0F);
41	lcd.setCursor(0,1);	41	output_c(0x00);

```

42 lcd.print("LUTFEN BEKLEYİNİZ....");
43 for(isitici=0;isitici<5;isitici++)
44 {digitalWrite(isitici_p,HIGH);
45 delay(100);
46 digitalWrite(isitici_p,LOW);
47 delay(100);}}
48 //ana program
49 void loop()
50 {lcd.clear();
51 lcd.setCursor(0,0);
52 lcd.print("GAZ OLCUMU YAPILIYOR");
53 lcd.setCursor(0,1);
54 lcd.print("LUTFEN BEKLEYİNİZ....");
55 bilgi=analogRead(gaz_sensor);
56 bilgi=map(bilgi,0,1023,0,255);
57 if(bilgi<40)
58 {sayi1++;
59 if(sayi1>=3)
60 {lcd.clear();
61 lcd.setCursor(0,0);
62 lcd.print("GAZ YOGUNLUGU");
63 lcd.setCursor(0,1);
64 lcd.print("1.SEVIYE");
65 seviye_yaz(1);
66 sayi1=0;
67 digitalWrite(alarm,LOW);}
68 delay(1000);}
69 if(bilgi>40 && bilgi<80)
70 {digitalWrite(gaz_algilama,HIGH);
71 delay(100);
72 digitalWrite(gaz_algilama,LOW);
73 sayi2++;
74 if(sayi2>=3)
75 {lcd.clear();
76 lcd.setCursor(0,0);
77 lcd.print("GAZ YOGUNLUGU");
78 lcd.setCursor(0,1);
79 lcd.print("2.SEVIYE");
80 seviye_yaz(3);
81 sayi2=0;
82 digitalWrite(alarm,LOW);}
83 delay(1000);}
84 if(bilgi>80 && bilgi<120)
85 {digitalWrite(gaz_algilama,HIGH);
86 delay(100);
87 digitalWrite(gaz_algilama,LOW);
88 sayi3++;

```

```

42 setup_adc(adc_clock_div_32);
43 setup_adc_ports(AN0_AN1_AN3);
44 lcd_init();
45 lcd_putc("\f");
46 printf(LCD_PUTC, "\f GAZ SENSOR UYG. \n ILERI MIKRO");
47 delay_ms(1000);
48 lcd_putc("\f");
49 printf(LCD_PUTC, "\f GAZ SENSORU ISINIYOR \n
LUTFEN BEKLEYİNİZ....");
50 for(isitici=0;isitici<5;isitici++)
51 {output_high(pin_c6);
52 delay_ms(100);
53 output_low(pin_c6);
54 delay_ms(100);}
55 //ana program
56 while(1)
57 {printf(LCD_PUTC, "\f GAZ OLCUMU YAPILIYOR \n
LUTFEN BEKLEYİNİZ....");
58 set_adc_channel(3);
59 delay_us(20);
60 bilgi=read_adc();
61 if(bilgi<40)
62 {sayi1++;
63 if(sayi1>=3)
64 {printf(LCD_PUTC, "\f GAZ YOGUNLUGU \n 1.SEVIYE");
65 seviye_yaz(1);
66 sayi1=0;
67 output_low(ALARM_OUT);}
68 delay_ms(1000);}
69 if(bilgi>40 && bilgi<80)
70 {output_high(pin_c7);
71 delay_ms(100);
72 output_low(pin_c7);
73 sayi2++;
74 if(sayi2>=3)
75 {printf(LCD_PUTC, "\f GAZ YOGUNLUGU \n 2.SEVIYE");
76 seviye_yaz(3);
77 sayi2=0;
78 output_low(ALARM_OUT);}
79 delay_ms(1000);}
80 if(bilgi>80 && bilgi<120)
81 {output_high(pin_c7);
82 delay_ms(100);
83 output_low(pin_c7);
84 sayi3++;
85 if(sayi3>=3)
86 {printf(LCD_PUTC, "\f GAZ YOGUNLUGU \n 3.SEVIYE");

```



89	if(sayi3>=3)	87	seviye_yaz(7);
90	{lcd.clear();	88	sayi3=0;
91	lcd.setCursor(0,0);	89	output_low(ALARM_OUT);}
92	lcd.print("GAZ YOGUNLUGU");	90	delay_ms(1000);}
93	lcd.setCursor(0,1);	91	if(bilgi>120 && bilgi<160)
94	lcd.print("3.SEVIYE");	92	{output_high(pin_c7);
95	seviye_yaz(7);	93	delay_ms(100);
96	sayi3=0;	94	output_low(pin_c7);
97	digitalWrite(alarm,LOW);}	95	sayi4++;
98	delay(1000);}	96	if(sayi4>=3)
99	if(bilgi>120 && bilgi<160)	97	{printf(LCD_PUTC, "\\f GAZ YOGUNLUGU \\n 4.SEVIYE");
100	{digitalWrite(gaz_algilama,HIGH);	98	seviye_yaz(15);
101	delay(100);	99	sayi4=0;
102	digitalWrite(gaz_algilama,LOW);	100	output_low(ALARM_OUT);}
103	sayi4++;	101	delay_ms(1000);}
104	if(sayi4>=3)	102	if(bilgi>160 && bilgi<200)
105	{lcd.clear();	103	{output_high(pin_c7);
106	lcd.setCursor(0,0);	104	delay_ms(100);
107	lcd.print("GAZ YOGUNLUGU");	105	output_low(pin_c7);
108	lcd.setCursor(0,1);	106	sayi5++;
109	lcd.print("4.SEVIYE");	107	if(sayi5>=3)
110	seviye_yaz(15);	108	{printf(LCD_PUTC, "\\f GAZ YOGUNLUGU \\n 5.SEVIYE");
111	sayi4=0;	109	seviye_yaz(31);
112	digitalWrite(alarm,LOW);}	110	sayi5=0;
113	delay(1000);}	111	output_low(ALARM_OUT);}
114	if(bilgi>160 && bilgi<200)	112	delay_ms(1000);}
115	{digitalWrite(gaz_algilama,HIGH);	113	if(bilgi>200)
116	delay(100);	114	{output_high(pin_c7);
117	digitalWrite(gaz_algilama,LOW);	115	delay_ms(100);
118	sayi5++;	116	output_low(pin_c7);
119	if(sayi5>=3)	117	sayialarm++;
120	{lcd.clear();	108	if(sayialarm>=3)
121	lcd.setCursor(0,0);	109	{printf(LCD_PUTC, "\\f GAZ YOGUNLUGU COK FAZLA \\n ALARM
122	lcd.print("GAZ YOGUNLUGU");		DEVREDE");
123	lcd.setCursor(0,1);	120	seviye_yaz(31);
124	lcd.print("5.SEVIYE");	121	output_high(ALARM_OUT);
125	seviye_yaz(31);	122	sayialarm=0;}
126	sayi5=0;	123	delay_ms(200); }}
127	digitalWrite(alarm,LOW);}		
128	delay(1000);}		
129	if(bilgi>200)		
130	{digitalWrite(gaz_algilama,HIGH);		
131	delay(100);		
132	digitalWrite(gaz_algilama,LOW);		
133	sayialarm++;		

134	if(sayialarm>=3)		
135	{lcd.clear();		
136	lcd.setCursor(0,0);		
137	lcd.print("GAZ YOGUNLUGU COK ");		
138	lcd.setCursor(0,1);		
139	lcd.print("FAZLA ALARM DEVREDE");		
140	seviye_yaz(31);		
141	sayialarm=0;		
142	digitalWrite(alarm,HIGH);}		
143	delay(200);}}		

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. MQ-6 gaz sensörünün dijital pin yerine analog pini (A0) kullanılmasının sebebi nedir?
2. 10 bitlik dijital veriyi 8 bitlik dijital veriye dönüştüren yazılım satırı, ARDUINO UNO ve PIC-16F876A için kaç numaralı satırlarda yer almaktadır?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

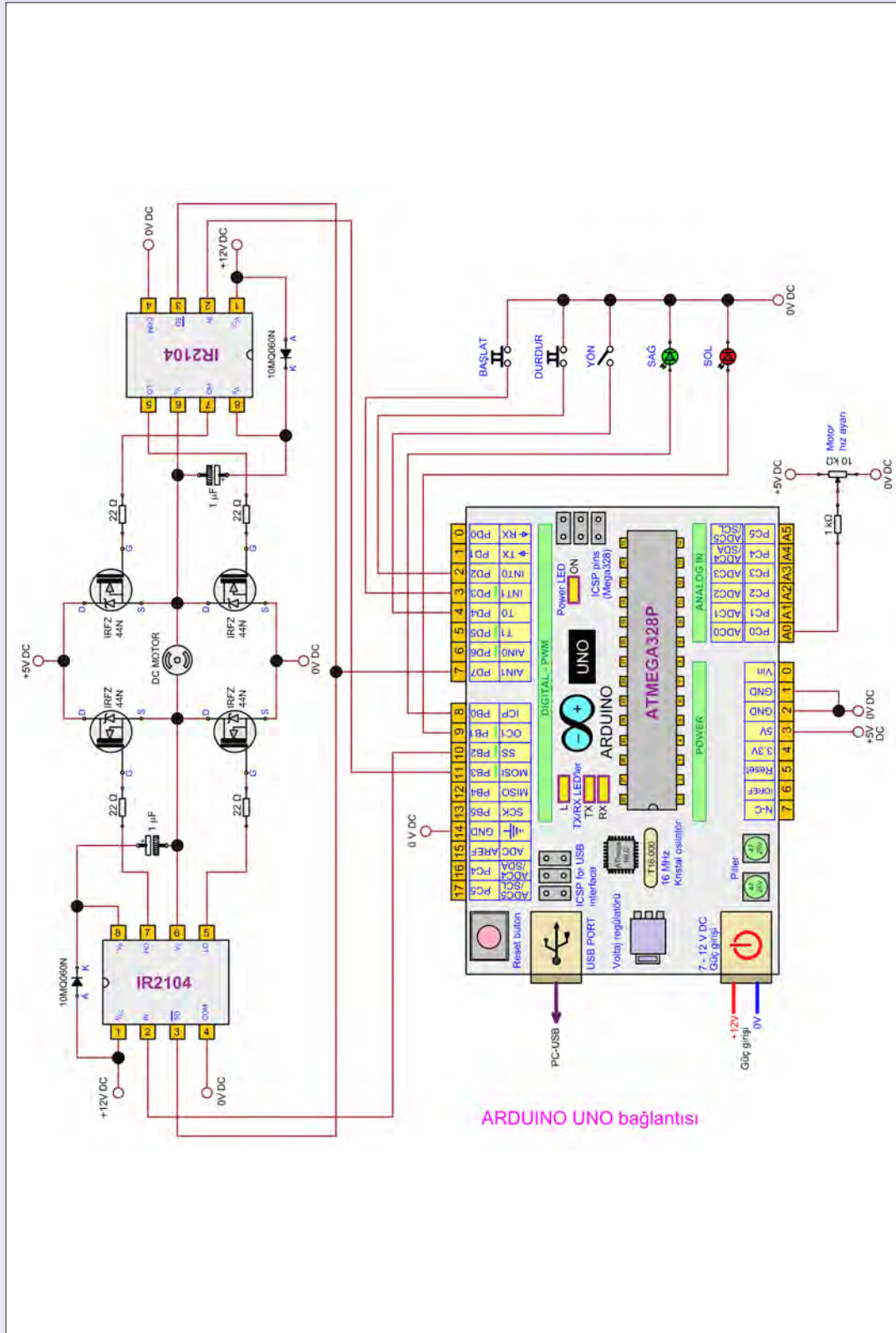
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



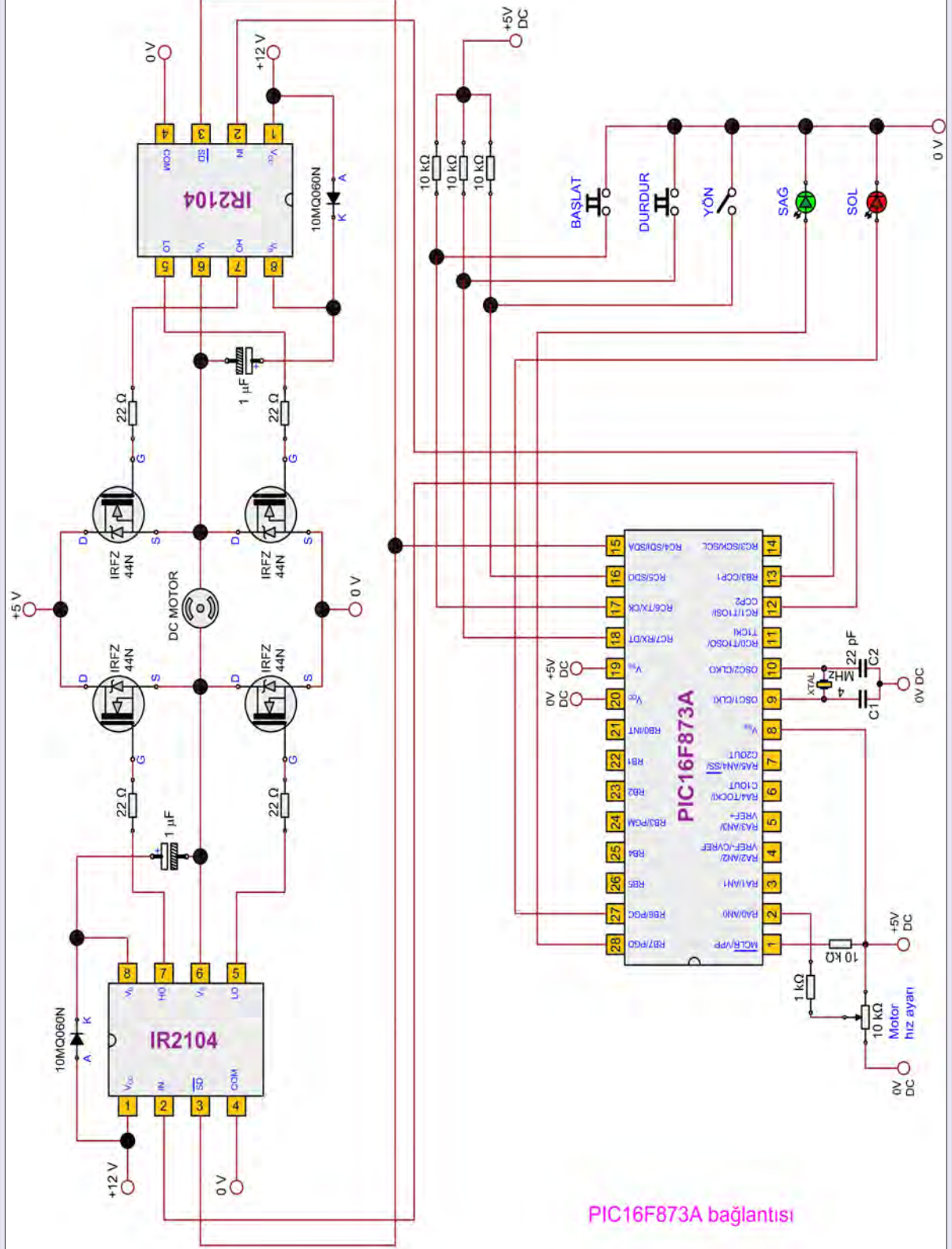
2.4. Motor Hız Kontrol Devresi



Devre Şeması: Görsel 2.8'de ARDUINO ile motor hız kontrol devresi, Görsel 2.9'da PIC16F873A entegresi ile motor hız kontrol devresi görülmektedir.



Görsel 2.8: ARDUINO UNO ile motor hız kontrol devresi



Görsel 2.9: PIC16F873A entegresi ile motor hız kontrol devresi



Devre Hakkında: Motor hız kontrol devre uygulaması IR2104 MOSFET sürücü entegresi ve IRFZ44 MOSFET'ler kullanılarak tasarlanmıştır. Mikrodenetleyici; yön anahtarı, butonların pozisyonu ve potansiyometreden gelen analog veriye göre PWM sinyali ile motor hız ve yön kontrolü yapmaktadır. Yön anahtarı ile DC motorun döneceği yön ayarı yapılır. Motor, yön anahtarı açık konumunda (lojik 1) ise sağa, kapalı konumunda (lojik 0) ise sola dönmeye ayarlanır. Devrede BAŞLAT butonuna basıldığında yön anahtarının konumuna ve potansiyometreden alınan analog veriye göre üretilen PWM sinyali ile DC motor sağa veya sola, ayarlanan hızda döndürülür. Yön belirten LED yanar. Devrede DURDUR butonuna basıldığında DC motor durur, LED söner.

Malzemeler: Tablo 2.7'de iki devrenin malzeme listesi görülmektedir.

Tablo 2.7: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F873A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaket	Yeterli ölçülerde	-
IR2104	MOSFET sürücü entegresi	2 adet
IRF244N	MOSFET	4 adet
DC motor	12 V DC	1 adet
Kondansatör	1 µF, kutuplu kondansatör	2 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
10MQ060N	Schottky doğrultucu	2 adet
Buton	Push-button	2 adet
Anahtar	On/off	1 adet
LED	5 mm	2 adet
Potansiyometre	1 kΩ, 10 kΩ	Birer adet
Direnç	22 Ω	4 adet
Direnç	10 kΩ	5 adet
Direnç	1 kΩ	1 adet
Kristal osilatör	4 MHz	1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	+5V...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli ölçülerde	Birer adet
Matkap ve uçları		-

Kod Yazılımları: Tablo 2.8'de iki devrenin kod yazılımları görülmektedir.

Tablo 2.8: Devrenin Kodları

No.	Kodlar (ARDUINO UNO)	No.	Kodlar (PIC16F873A)
1	//değişkenler	1	//kütüphaneler ve tanımlamalar
2	const int yön = 4;	2	#include <16F873A.h>
3	const int baslat = 3;	3	#device ADC=10
4	const int durdur = 2;	4	#FUSES NOWDT
5	const int hiz_oku = A0;	5	#FUSES NOBROWNOUT
6	const int saga_donus = 10;	6	#FUSES NOLVP

```

7 const int sola_donus = 11;
8 const int sag_led = 8;
9 const int sol_led = 9;
10 const int dur = 7;
11 long hiz_ayar;
12 int motor_calistir=0, sag=0, sol=0;
13 //genel ayarlar
14 void setup()
15 {pinMode(yon, INPUT_PULLUP);
16 pinMode(baslat, INPUT_PULLUP);
17 pinMode(durdur, INPUT_PULLUP);
18 pinMode(saga_donus, OUTPUT);
19 pinMode(sola_donus, OUTPUT);
20 pinMode(dur, OUTPUT);
21 pinMode(sag_led, OUTPUT);
22 pinMode(sol_led, OUTPUT);}
23 //ana program
24 void loop() {
25 if (digitalRead (yon)==1)
26 {sag=1;
27 sol=0;}
28 else
29 {sol=1;
30 sag=0;}
31 if (!digitalRead (baslat))
32 {motor_calistir=1;
33 while(!digitalRead (baslat));}
34 if (!digitalRead (durdur))
35 {motor_calistir=0;
36 digitalWrite(dur, LOW);
37 digitalWrite(sol_led, LOW);
38 digitalWrite(sag_led, LOW);
39 digitalWrite(saga_donus,LOW);
40 digitalWrite(sola_donus,LOW);
41 while(!digitalRead (durdur));}
42 if ((motor_calistir==1)&& (sag==1))
43 {hiz_ayar= analogRead(hiz_oku);
44 hiz_ayar=map(hiz_ayar,0,1023,0,255);
45 digitalWrite(dur, HIGH);
46 analogWrite(saga_donus,hiz_ayar);
47 digitalWrite(sol_led, LOW);
48 digitalWrite(sag_led, HIGH);}
49 if ((motor_calistir==1)&& (sol==1))
50 {hiz_ayar= analogRead(hiz_oku);
51 hiz_ayar=map(hiz_ayar,0,1023,0,255);
52 digitalWrite(dur, HIGH);

```

```

7 #use delay(crystal=4000000)
8 #use fast_io(a)
9 #use fast_io(b)
10 #use fast_io(c)
11 #define yon PIN_C5
12 #define baslat PIN_C6
13 #define durdur PIN_C7
14 #define hiz_oku PIN_A0
15 #define saga_donus PIN_C2
16 #define sola_donus PIN_C1
17 #define sag_led PIN_B7
18 #define sol_led PIN_B6
19 #define dur PIN_C4
20 long hiz_ayar;
21 int motor_calistir=0, sag=0, sol=0;
22 void main()
23 {setup_spi(SPI_SS_DISABLED);
24 setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
25 setup_timer_1(T1_DISABLED);
26 setup_timer_2(T2_DIV_BY_16,255,8);
27 set_tris_c(0xE0);
28 set_tris_b(0x00);
29 set_tris_a(0x01);
30 output_b(0x00);
31 output_c(0x00);
32 setup_adc(adc_clock_div_32);
33 setup_adc_ports(AN0_AN1_AN3);
34 delay_ms(100);
35 while(1)
36 {if (input(yon)==1)
37 {sag=1;
38 sol=0;}
39 else
40 {sol=1;
41 sag=0;}
42 if (input(baslat)==0)
43 { motor_calistir=1;
44 while(!input (baslat));}
45 if (!input (durdur))
46 { motor_calistir=0;
47 output_low(dur);
48 output_low(sol_led);
49 output_low(sag_led);
50 setup_CCP1(CCP_OFF);
51 setup_CCP2(CCP_OFF);
52 while(!input (durdur));}

```



53	analogWrite(sola_donus,hiz_ayar);	53	if ((motor_calistir==1)&& (sag==1))
54	digitalWrite(sol_led, HIGH);	54	{set_adc_channel(0);
55	digitalWrite(sag_led, LOW);}}	55	delay_us(20);
		56	hiz_ayar=read_adc();
		57	output_high(dur);
		58	setup_CCP2(CCP_OFF);
		59	setup_CCP1(CCP_PWM);
		60	set_pwm1_duty(hiz_ayar);
		61	output_low(sol_led);
		62	output_high(sag_led);
		63	delay_ms(100);}
		64	if ((motor_calistir==1)&& (sol==1))
		65	{ set_adc_channel(0);
		66	delay_us(20);
		67	hiz_ayar=read_adc();
		68	output_high(dur);
		69	setup_CCP1(CCP_OFF);
		70	setup_CCP2(CCP_PWM);
		71	set_pwm2_duty(hiz_ayar);
		72	output_high(sol_led);
		73	output_low(sag_led);
		74	delay_ms(100);}}}

İşlem Basamakları

1. Uygun tipli mikrodnetleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodnetleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Motor hız ayarının yapılması için dijital pin yerine analog pin kullanılmasının sebebi nedir?
2. Motor yön belirlemede buton yerine anahtar kullanılmasının sebebi nedir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

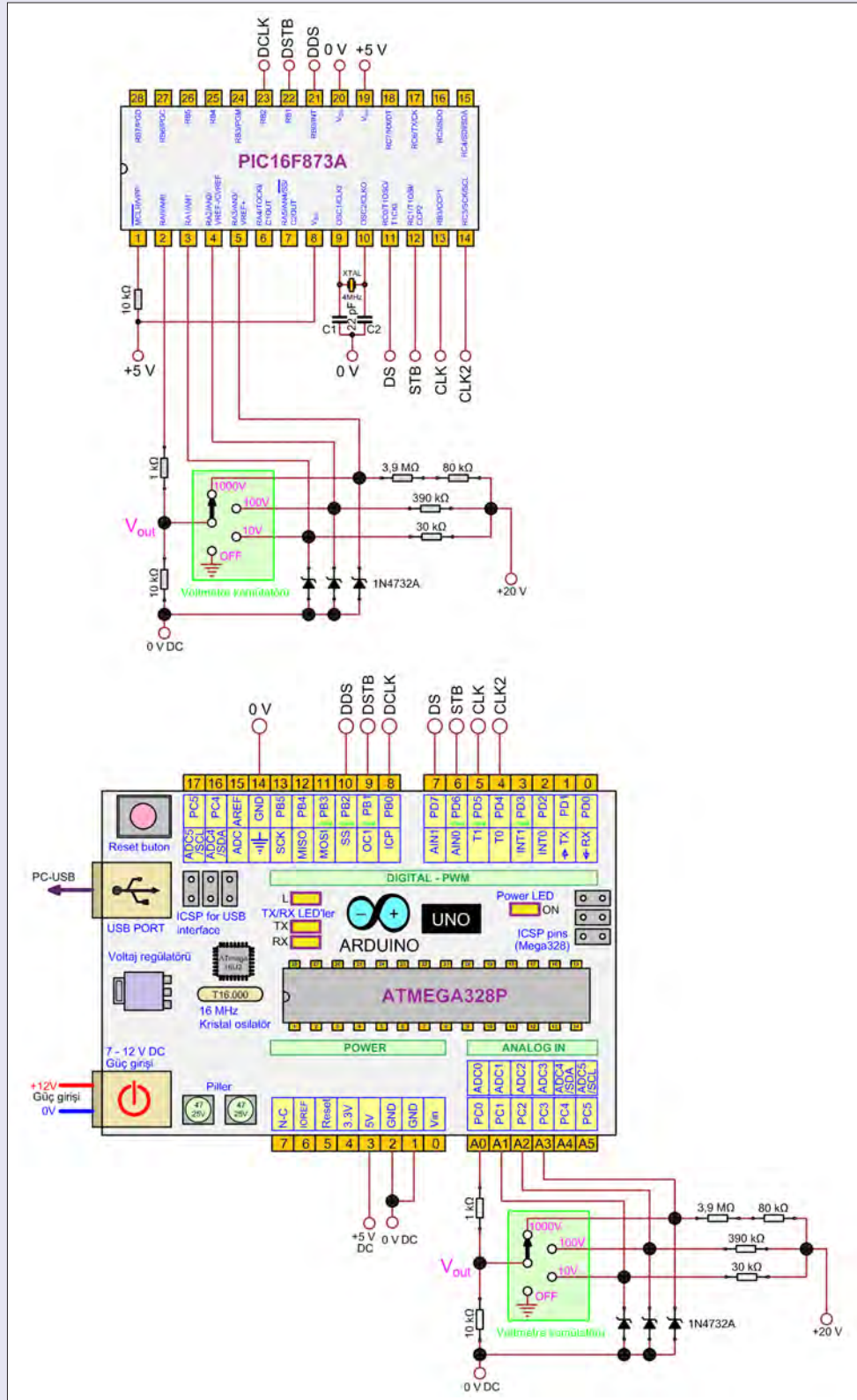
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



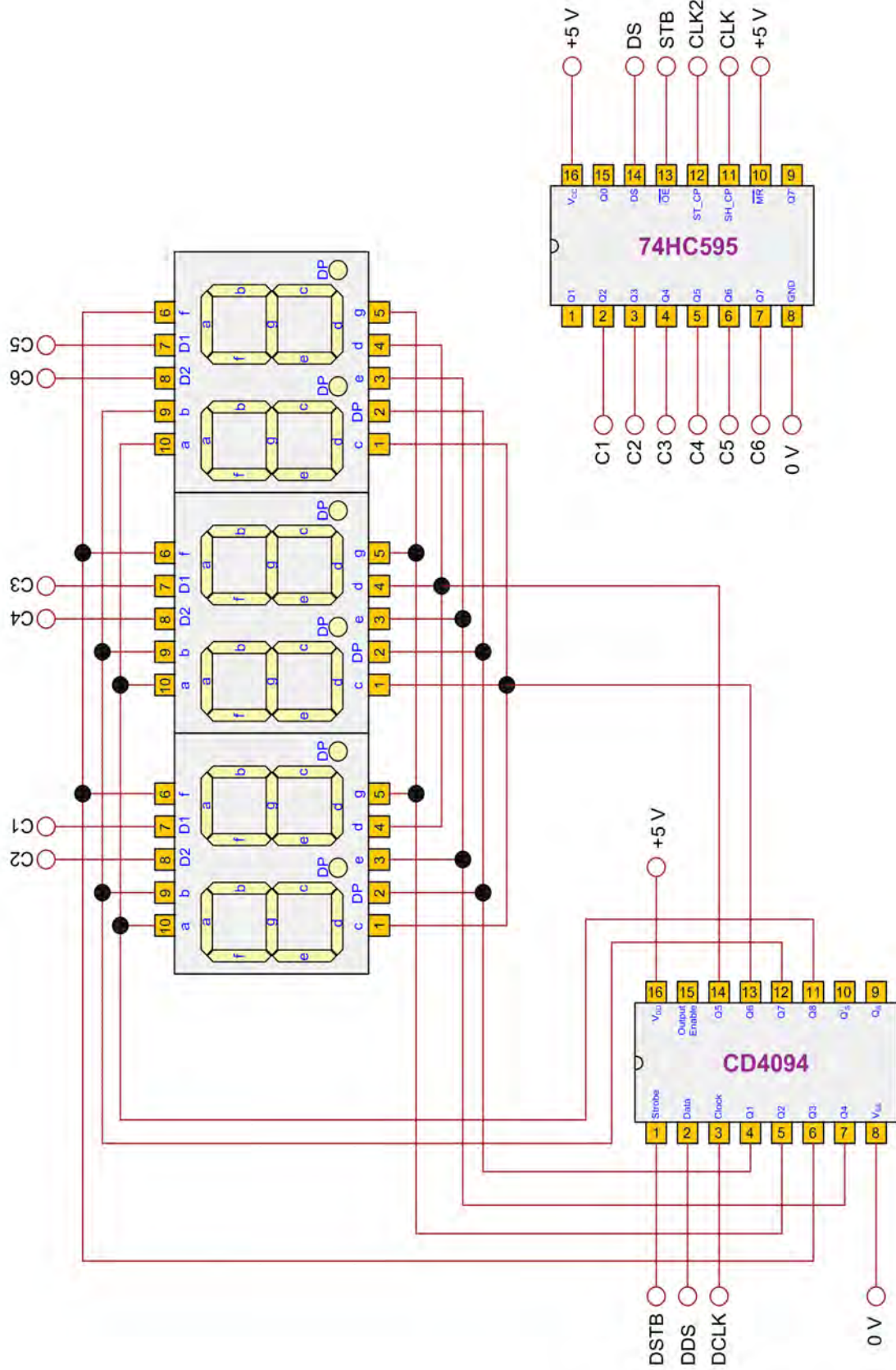
2.5. Dijital Voltmetre Devresi



Devre Şeması: Görşel 2.10'da dijital voltmetre kontrol devresi, Görşel 2.11'de dijital voltmetre devresi görölmektedir.



Görşel 2.10: Dijital voltmetre kontrol devresi



Görsel 2.11: Dijital voltmetre devresi

Devre Hakkında: Dijital voltmetre devresi 8 bitlik kaymalı kaydediciler (shift register) ve 6 digit ortak katot display kullanılarak tasarlanmıştır. Mikrodenetleyici, komütatör anahtar konuma göre gelen analog sinyali ölçer ve hesaplama yaparak display ekranda gösterir. Komütatör anahtarı ile ölçüm yapılacak aralık seçilir (0-10 V-100 V-1000 V). Komütatör anahtar konumuna göre işlemciye ayrıca ölçüm aralığı bilgisi gönderilir. Mikrodenetleyici, komütatör anahtardan gelen analog veriyi 10 bitlik dijital veriye dönüştürür ve ölçüm aralığı bilgisine göre hesaplama yapar. Hesaplanan veri kodlanarak display ekranda gösterilir (Görsel 2.10-2.11).

Malzemeler: Tablo 2.9’da iki devrenin malzeme listesi görülmektedir.

Tablo 2.9: Devrelerde Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
PIC16F873A	Mikrodenetleyici entegresi	1 adet
Breadboard	17 cm x 5,5 cm, 830 hücreli	1 adet
Bakır plaketa	Yeterli ölçülerde	-
Display	6 digit	2 adet
CD4094	Shift register	4 adet
74HC595	Shift register	1 adet
Direnç	3.9 kΩ, 80 kΩ, 390 kΩ, 30 kΩ,10 kΩ,1 kΩ	2 adet
1N4732A	Zener diyot	2 adet
Komitatör anahtar	4 kademe	2 adet
Direnç	10 kΩ	2 adet
Kristal osilatör	4 MHz	1 adet
Kondansatör	22 pF, mercimek tipi, kutupsuz	2 adet
Zil teli	0,5 veya 0,75 mm ²	Birer adet
DC güç kaynağı	+5V...24 V DC	4 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	5 adet
Baskı devre kalemi	S veya M tipli	1 adet
Asit ve tuz ruhu	Yeterli ölçülerde	-
Matkap ve uçları		

Kod Yazılımları: Tablo 2.10’da iki devrenin kod yazılımları görülmektedir.

Tablo 2.10: Devrenin Kodları

No.	Kodlar (ARDUINO UNO)	No.	Kodlar (PIC16F873A)
1	#include <SPI.h>	1	#include <16f873A.h>
2	#define ds 7	2	#device ADC=10
3	#define latch 6	3	#fuses HS, NOWDT, NOPROTECT, NOBROWNOUT, NOLVP, NOPUT, NOWRT, NODEBUG, NOCPD
4	#define clk1 5	4	#use delay (clock=20000000)
5	#define clk2 4	5	#use fast_io(c)
6	#define STB 9	6	#use fast_io(a)
7	#define CLKS 8	7	#use fast_io(b)
8	#define DATAS 10	8	#define ds pin_c0
9	#define volt_giris A0	9	#define latch pin_c1
10	#define K10 A1	10	#define clock pin_c2
11	#define K100 A2	11	#define clk2 pin_c3
12	#define K1000 A3		



```

13 //değişkenler
14 int j=0, k=0;
15 unsigned int digit[12]= {0x3F, 0x06, 0x5B, 0x4F,
16 0x66, 0x6D, 0x7C, 0x07, 0x7F, 0x6F, 0x46, 0x3E };
17 unsigned int binler, yuzler, onlar, birler, i, nokta, giden;
18 unsigned long int bilgi, voltaj2, sayi;
19 float voltaj, kademe;
20 //alt programlar
21 void kademe_sec ()
22 {kademe=analogRead(K10);
23 kademe=0.0048828125*kademe;
24 if (kademe<2.75)
25 {nokta=1;}
26 kademe=analogRead(K100);
27 kademe=0.0048828125*kademe;
28 if (kademe<2.75)
29 {nokta=2;}
30 kademe=analogRead(K1000);
31 kademe=0.0048828125*kademe;
32 if (kademe<2.75)
33 {nokta=3;}}
34 void disp_yaz(int ent_data)
35 {digitalWrite(latch,HIGH);
36 for(j=0;j<8;j++)
37 {digitalWrite(ds,!bitRead(ent_data,j));
38 digitalWrite(clk1,HIGH);
39 digitalWrite(clk2,LOW);
40 delayMicroseconds(1);
41 digitalWrite(clk1,LOW);
42 digitalWrite(clk2,HIGH);}
43 void disp_gonder(int dispsayi, int dnokta)
44 {giden=digit[dispsayi];
45 if (dnokta==1)
46 bitSet(giden,7);
47 shiftOut(DATAS, CLKS, LSBFIRST, giden);
48 digitalWrite(STB, HIGH);
49 delayMicroseconds(1);
50 digitalWrite(STB, LOW);}
51 //genel ayarlar
52 void setup()
53 {pinMode(ds, OUTPUT);
54 pinMode(latch, OUTPUT);
55 pinMode(clk1, OUTPUT);
56 pinMode(clk2, OUTPUT);
57 pinMode(DATAS, OUTPUT);
58 pinMode(CLKs, OUTPUT);

```

```

12 #define dds pin_b0
13 #define dlatch pin_b1
14 #define dclock pin_b2
15 int j=0;
16 const int8 digit[12]={0x3F, 0x06, 0x5B, 0x4F, 0x66,
17 0x6D, 0x7C, 0x07, 0x7F, 0x6F, 0x46, 0x3E};
18 unsigned int binler, yuzler, onlar, birler, i, nokta, giden;
19 unsigned long int bilgi, voltaj2, sayi;
20 float voltaj, kademe;
21 void kademe_sec ()
22 {set_adc_channel(1);
23 delay_us(20);
24 kademe=read_adc();
25 kademe=0.0048828125*kademe;
26 if (kademe<2.75)
27 {nokta=1;}
28 set_adc_channel(2);
29 delay_us(20);
30 kademe=read_adc();
31 kademe=0.0048828125*kademe;
32 if (kademe<2.75)
33 {nokta=2;}
34 set_adc_channel(3);
35 delay_us(20);
36 kademe=read_adc();
37 kademe=0.0048828125*kademe;
38 if (kademe<2.75)
39 {nokta=3;}}
40 void disp_gonder(int8 dispsayi,int8 dnokta)
41 {giden=digit[dispsayi];
42 if (dnokta==1)
43 bit_set(giden,7);
44 for(i=0;i<8;i++)
45 {output_bit(dds,bit_test(giden,i));
46 output_high(dclock);
47 delay_us(1);
48 output_low(dclock);}
49 output_high(dlatch);
50 delay_us(1);
51 output_low(dlatch);}
52 void disp_yaz(int8 ent_data)
53 {output_high(latch);
54 for(j=0;j<8;j++)
55 {output_bit(ds,!bit_test(ent_data,j));
56 output_high(clock);
57 output_low(clk2);
58 delay_us(1);

```

59	pinMode(STB, OUTPUT);}	59	output_low(clock);
60	//ana program	60	output_high(clk2);}
61	void loop()	61	output_low(latch);}
62	{kademe_sec();	62	void main ()
63	bilgi=analogRead(volt_giris);	63	{setup_timer_1(T1_DISABLED);
64	voltaj=0.0048828125*bilgi;	64	setup_timer_2(T2_DISABLED,0,1);
65	if (voltaj>2.5)	65	setup_CCP1(CCP_OFF);
66	{disp_gonder(0x0A,0);	66	setup_CCP2(CCP_OFF);
67	disp_yaz(0x20);	67	set_tris_c(0x00);
68	delay(5);	68	set_tris_a(0x0F);
69	disp_yaz(0x00);	69	set_tris_b(0x00);
70	disp_gonder(0x00,0);}	70	output_c(0x00);
71	voltaj2=voltaj*400;	71	output_b(0x00);
72	if (voltaj2<=0.5)	72	setup_adc(adc_clock_div_2);
73	nokta=0;	73	setup_adc_ports(ALL_ANALOG);
74	binler=voltaj2/1000;	74	while(1)
75	sayi=voltaj2%1000;	75	{kademe_sec();
76	yuzler=sayi/100;	76	set_adc_channel(0);
77	sayi=sayi%100;	77	delay_us(20);
78	onlar=sayi/10;	78	bilgi=read_adc();
79	sayi=sayi%10;	79	voltaj=0.0048828125*bilgi;
80	birler=sayi/1;	80	if (voltaj>2.5)
81	if (voltaj<=2.5)	81	{disp_gonder(0x0A,0);
82	{disp_gonder(0x0B,0);	82	disp_yaz(0x20);
83	disp_yaz(0x01);	83	delay_ms(5);
84	delay(5);	84	disp_yaz(0x00);
85	disp_yaz(0x00);	85	disp_gonder(0x00,0);}
86	disp_gonder(0x00,0);	86	voltaj2=voltaj*400;
87	disp_gonder(birler,0);	87	if (voltaj2<=0.5)
88	disp_yaz(0x02);	88	nokta=0;
89	delay(5);	89	binler=voltaj2/1000;
90	disp_yaz(0x00);	90	sayi=voltaj2%1000;
91	disp_gonder(0x00,0);	91	yuzler=sayi/100;
92	if (nokta==2)	92	sayi=sayi%100;
93	{disp_gonder(onlar,1);}	93	onlar=sayi/10;
94	else	94	sayi=sayi%10;
95	{disp_gonder(onlar,0);}	95	birler=sayi/1;
96	disp_yaz(0x04);	96	if (voltaj<=2.5)
97	delay(5);	97	{disp_gonder(0x0B,0);
98	disp_yaz(0x00);	98	disp_yaz(0x01);
99	disp_gonder(0x00,0);	99	delay_ms(5);
100	if (nokta==1)	100	disp_yaz(0x00);
101	{disp_gonder(yuzler,1);}	101	disp_gonder(0x00,0);
102	else	102	disp_gonder(birler,0);
103	{disp_gonder(yuzler,0);}	103	disp_yaz(0x02);
104	disp_yaz(0x08);	104	delay_ms(5);
105	delay(5);	104	disp_yaz(0x00);



106	disp_yaz(0x00);	105	disp_gonder(0x00,0);
107	disp_gonder(0x00,0);	106	if (nokta==2)
108	disp_gonder(binler,0);	107	{disp_gonder(onlar,1);}
109	disp_yaz(0x10);	108	else
110	delay(5);	109	{disp_gonder(onlar,0);}
111	disp_yaz(0x00);	110	disp_yaz(0x04);
112	disp_gonder(0x00,0);}}	111	delay_ms(5);
		112	disp_yaz(0x00);
		113	disp_gonder(0x00,0);
		114	if (nokta==1)
		115	{disp_gonder(yuzler,1);}
		116	else
		117	{disp_gonder(yuzler,0);}
		118	disp_yaz(0x08);
		119	delay_ms(5);
		120	disp_yaz(0x00);
		121	disp_gonder(0x00,0);
		122	disp_gonder(binler,0);
		123	disp_yaz(0x10);
		124	delay_ms(5);
		125	disp_yaz(0x00);
		126	disp_gonder(0x00,0);}}

İşlem Basamakları

1. Uygun tipli mikrodeneleyiciyi (PIC veya ARDUINO) seçiniz.
2. Devreyi bilgisayar ortamında kurarak devrenin simülasyonu yapınız.
3. Baskı devre yapılacaksa şemasını bilgisayar ortamında oluşturunuz.
4. Devreyi breadboard veya plaket üzerinde kurunuz.
5. Uygun yazılımı editör ortamında oluşturup derleyiniz ve mikrodeneleyiciye yükleyiniz. Devreyi çalıştırınız.
6. Yazılımları ve devreyi öğretmeninize teslim ediniz.

Sorular

1. Analog verinin 8 bit yerine 10 bit dijital veriye dönüştürülme sebebi nedir?
2. Komütatör anahtardan hem analog verinin hem de ölçüm aralığı verisinin alınma sebebi nedir?

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerin başındaki boşluğa ifade doğru ise “D”, yanlış ise “Y” yazınız.

1. () Mikrodenetleyici girişlerine yalnızca dijital veriler uygulanır.
2. () LCD ekranı yazılımda kolayca kullanmak için LCD kütüphanesinden faydalanılır.
3. () Mikrodenetleyicili sistemlerde çalışma hızı kullanılan kristalin frekansına bağlıdır.
4. () Mikrodenetleyicili sistemlerde LED diyot, giriş elemanı olarak kullanılır.
5. () Mikrodenetleyici, 12V röleyi iletime geçirmek için sürücü devresine ihtiyaç duyar.

B) Aşağıdaki cümlelerde boş bırakılan noktalı yerlere, doğru ifadeleri yazınız.

6. Sıcaklık ölçmek için sensörü kullanılır.
7. Mesafe ölçümü yapılırken ses hızı m/sn olarak kabul edilir.
8. Motor hızını ayarlamak için sinyaline ihtiyaç duyulur.
9. Analog veri, mikrodenetleyici ile 8 bitlik veya bitlik dijital veriye dönüştürülür.
10. Voltmetrede kademe ayarı anahtar ile yapılır.

C) Aşağıdaki çoktan seçmeli soruları okuyunuz ve doğru seçeneği işaretleyiniz.

11. LPG/izobütan/propan gaz ölçümünü yapan sensör aşağıdakilerden hangisidir?

- A) MQ-2 B) MQ-3 C) MQ-4
D) MQ-5 E) MQ-6

12. Mesafe ölçümünü yapan sensör aşağıdakilerden hangisidir?

- A) HC-SR03 B) HC-SR04 C) HC-SR05
D) HC-SR06 E) HC-SR07

13. Analog girişi okuyan ARDUINO komutu aşağıdakilerden hangisidir?

- A) analogRead B) analogWrite C) AnalogRead
D) Analogwrite E) analogread

14. x=5 ise x++ komutunda x'in alacağı değer aşağıdakilerden hangisidir?

- A) 5 B) 6 C) 7
D) 8 E) 9

15. PIC mikrodenetleyicide “B” portuna 12 sayısını gönderen komut aşağıdakilerden hangisidir?

- A) output_b(b12) B) output_b(h12) C) output_b(0x12)
D) output_b(12) E) output_b(012)



2. ÖĞRENME BİRİMİ: UYGULAMA DEĞERLENDİRME TABLOSU

Aşağıda uygulama değerlendirme ölçütleri (Tablo 2.11) görülmektedir. Birinci öğrenme biriminde yapılan uygulama çalışmalarını bu ölçütlere göre değerlendiriniz. Her ölçütün değerlendirme sonucunu karşısındaki puanlama tablosunda işaretleyiniz (v) ve tablodan alınan toplam puanı hesaplayınız.

Tablo 2.11: Uygulama Değerlendirme Tablosu

Değerlendirme Ölçütleri		Puanlama Ölçütleri					
		Kötü (0)	Yeterli (1)	Orta (2)	İyi (3)	Çok İyi (4)	Mükemmel (5)
1.	Projede kullanılacak mikrodenetleyiciyi belirlendi.						
2.	Proje kodları uygun yazılım editöründe oluşturuldu.						
3.	Proje kodları derlendi ve kontrol edildi.						
4.	Proje kodları mikrodenetleyiciye yüklendi.						
5.	Proje devre tasarımı şema çizim programında çizildi.						
6.	Proje devresinin simülasyonunu yapıldı.						
7.	Projenin kurulacağı alan (breadboard, PCB vb.) belirlendi.						
8.	Projenin baskı devresinin tasarımı ya da breadboard kurulumu yapıldı.						
9.	Projedeki entegrelerin ve elemanların kullanılacak uçları kataloglar yardımıyla belirlendi.						
10.	Projede kullanılan elemanların sağlamlık kontrolleri yapıldı.						
11.	Proje malzemeleri devreye uygun şekilde yerleştirildi.						
12.	Projeye uygun kablolar belirlendi.						
13.	Proje kablo bağlantıları doğru yapıldı.						
14.	Projede, belirlenen parçalar kullanıldı.						
15.	Çalışmanın gerçekleştirilme aşamaları düzenli olarak kontrol edildi.						
16.	Projede oluşturulan devre istenilen şekilde çalıştı.						
17.	Projeye oluşturulan devrenin çalışması detaylı olarak incelendi.						
18.	Uygulamalar verilen sürede tamamlandı						
19.	Proje, zamanında teslim edildi.						
20.	Proje raporu eksiksiz teslim edildi.						
TOPLAM							
Değerlendirme sonucunda 100 puan üzerinden toplam puan almıştır.							
Öğrencinin Adı-Soyadı:							
Sınıfı-Numarası:							
İmza:							
Öğretmenin Adı-Soyadı:							
İmza:							



EK - A

PROJE UYGULAMALARI

1. AVOMetre Projesi
2. Engel Algılayan Robot Projesi
3. PV Panel ile Sokak Aydınlatması Projesi
4. SMD Eleman Lehimleme ve Sökme Cihazı Projesi
5. Göstergeli Zaman Ayarlı Dijital Grup Priz Projesi
6. Dimmerlenen Uzaktan Kontrollü Lamba Projesi
7. Robot Kol Kontrol Projesi
8. Termometreli Kayan Yazılı Saat Devre Projesi
9. Ayarlı DC Güç Kaynağı Projesi
10. ARDUINO Deney Seti Projesi



UYGULAMA YÖNERGESİ



Bu öğrenme birimine ait 10 farklı proje bulunmaktadır. Aşağıdaki tabloda üç farklı mikroişlemci modeline göre verilen projeler görülmektedir. Uygulamaları aşağıdaki yönerge doğrultusunda gerçekleştiriniz.

- İlgili uygulamada kullanılacak mikroişlemci modelini belirleyiniz.
- Uygulamayı yapacak grubu 2 ila 6 kişi olacak şekilde oluşturunuz. Her grup üyesinin uygulama sürecindeki görevini belirleyiniz.
- Belirlenen mikroişlemci modeli için verilen uygulama devresinin simülasyonunu elektronik simülasyon yazılımı ile yapınız.
- Belirlenen mikroişlemci modeli için verilen uygulama devresinin baskı devre şemasını baskı devre tasarım yazılımı ile yapınız.
- Belirlenen mikroişlemci modeli için verilen uygulama devresini baskı devre plaketi üzerine kurunuz. Bu doğrultuda malzeme listesi oluşturunuz ve malzemelerin sağlamlıklarını kontrol ediniz.
- Devrenin kodlarını ilgili editörde oluşturup derleyiniz ve mikroişlemciye yükleyiniz.
- Devrenin simülasyon ve programlama dosyalarını uygun isimde kaydediniz ve öğretmeninize teslim ediniz.
- İşlem sonunda devrenin raporunu öğretmeninizin yönergesi doğrultusunda (soruları da cevaplayarak) hazırlayıp öğretmeninize teslim ediniz.

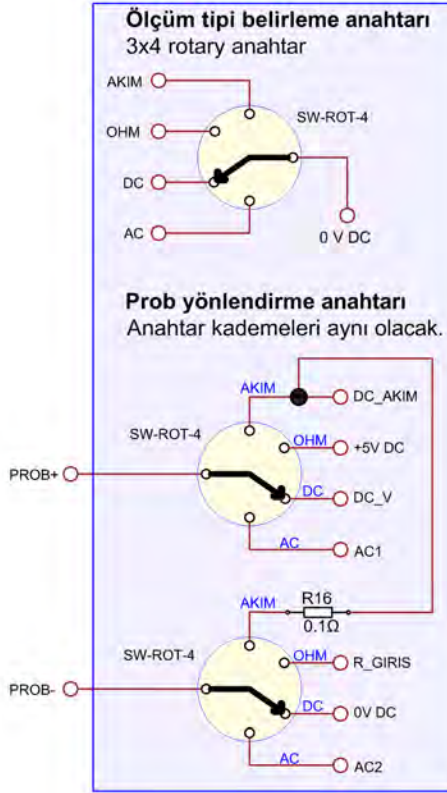
UYGULAMA PROJELER

No.	Uygulama Adı	Mikrodenetleyici
01	AVOmetre Projesi	ARDUINO UNO
02	Engel Algılayan Robot Projesi	ARDUINO UNO
03	PV Panel ile Sokak Aydınlatması Projesi	ARDUINO NANO
04	SMD Lehimleme ve Sökme Cihazı Projesi	ARDUINO UNO
05	Göstergeli Zaman Ayarlı Dijital Grup Priz Projesi	ARDUINO NANO
06	Dimmerlenen Uzaktan Kontrollü Lamba Projesi	ARDUINO NANO
07	Robot Kol Kontrol Projesi	ARDUINO UNO
08	Termometreli Kayan Yazılı Saat Devre Projesi	ARDUINO UNO
09	Ayarlı DC Güç Kaynağı Projesi	ARDUINO NANO
10	ARDUINO Deney Seti Projesi	ARDUINO UNO

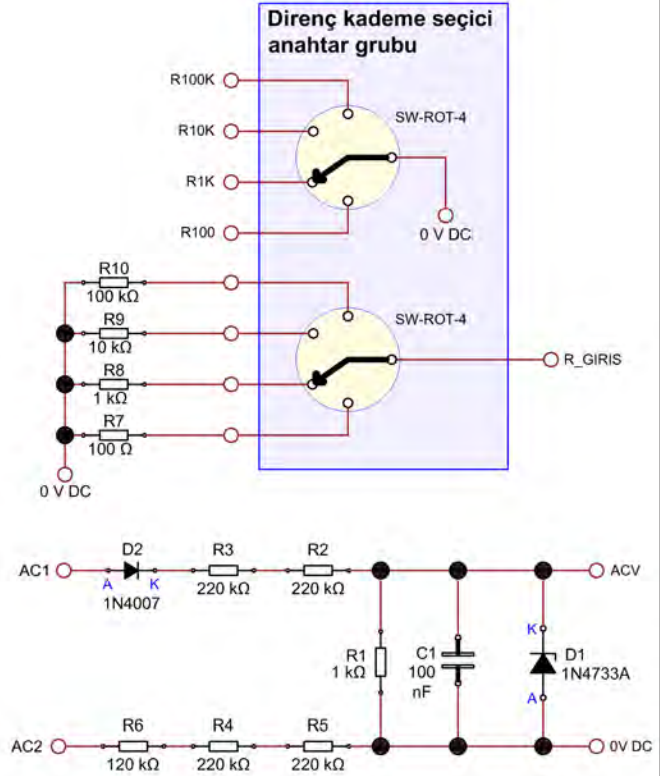
DİKKAT!

- Uygulama ile ilgili tüm dosyalara yanda verilen karekoddan ulaşabilirsiniz.
- Ekteki kısımdan devrelerde kullanılan elemanların ayak bağlantılarına, baskı devre şemalarına ve detaylı teknik bilgilerine ulaşabilirsiniz.

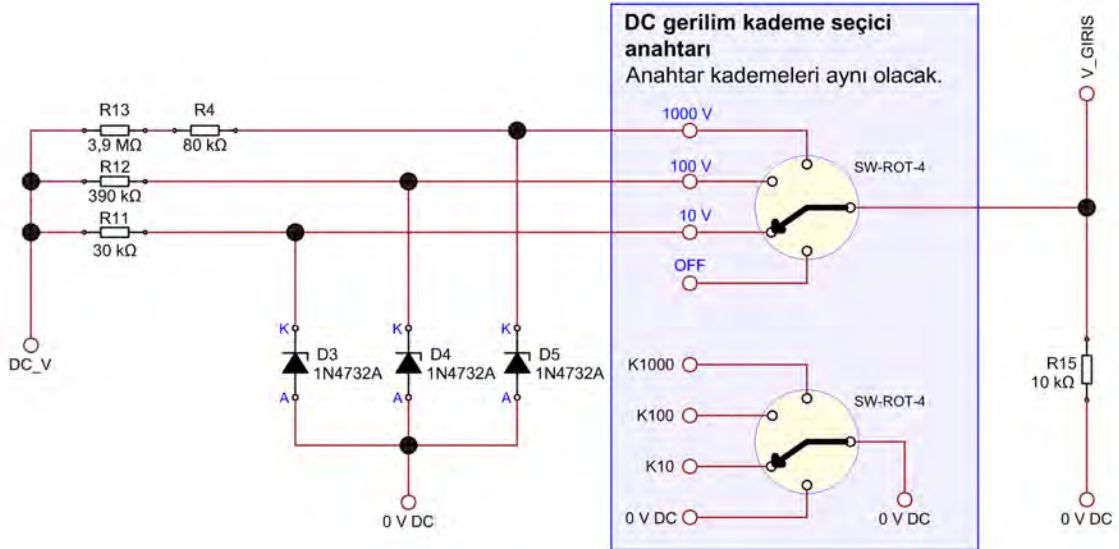
SW2 3x4 rotary anahtar



SW3 3x4 rotary anahtar

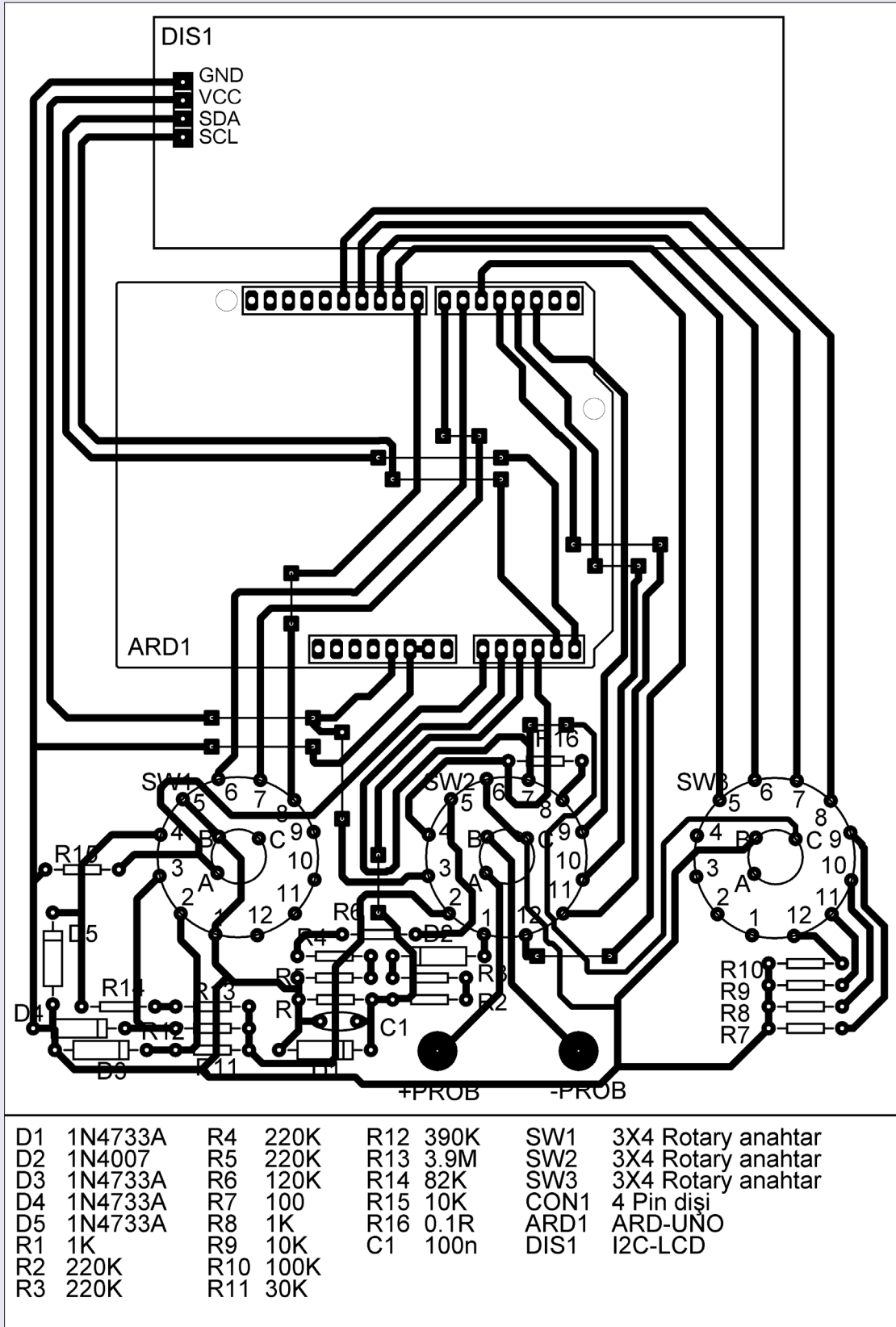


SW1 3x4 rotary anahtar



Görsel A.1: ARDUINO UNO ile AVOmetre devresi (b)

Baskı Devre Şeması: Görsel A.2’de devrenin baskı devre şeması görülmektedir.



Görsel A.2: ARDUINO UNO ile yapılan AVometre baskı devresi



Malzemeler: Tablo A.1’de devrenin malzeme listesi görülmektedir.

Tablo A.1: Devrede Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
LM016-I2C modülü	Seri iletişim kartlı LCD display	1 adet
3x4 rotary anahtar	3 kutup 4 çıkışlı	3 adet
1N4733A diyot	5 V zener diyot	4 adet
1N4001 diyot	Kristal diyot	1 adet
Direnç 0,25W	100 Ω	1 adet
Direnç 0,25W	1k Ω	2 adet
Direnç 0,25W	10 k Ω	2 adet
Direnç 0,25W	30 k Ω	1 adet
Direnç 0,25W	82 k Ω	1 adet
Direnç 0,25W	100 k Ω	1 adet
Direnç 0,25W	120 k Ω	1 adet
Direnç 0,25W	220 k Ω	4 adet
Direnç 0,25W	390 k Ω	1 adet
Direnç 0,25W	3.9 M Ω	1 adet
Direnç 2W	0.1 Ω	1 adet
Pin header	4 pin dişi header	1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-

Kod Yazılımları: Tablo A.2’de devrenin kod yazılımı görülmektedir.

Tablo A.2: ARDUINO UNO’lu AVOMetre Devresinin Kod Yazılımı

No.	Kodlar
1	//kütüphaneler ve tanımlamalar
2	#include <LiquidCrystal_I2C.h>
3	LiquidCrystal_I2C lcd(0x23,16,2);
4	#define volt_giris A0
5	#define akima A3
6	#define K10 6
7	#define K100 7
8	#define K1000 8
9	#define R100 9
10	#define R1K 10
11	#define R10K 11
12	#define R100K 12
13	//değişkenler
14	const int AC = 2;
15	const int DC = 3;
16	const int OHM = 4;
17	const int AKIM = 5;
18	boolean AC_A=false, DC_A=false, OHM_A=false, AKIM_A=false;
19	int dirPin = 0;
20	float R1 = 100;
21	float R2 = 0;
22	unsigned long int o_volt;
23	float voltaj;
24	//alt programlar
25	void kademe_sec ()
26	{
27	if (!digitalRead(AC)){
28	lcd.clear();
29	lcd.print("AC VOLTAJ OLCUMU");
30	AC_A=true, DC_A=false, OHM_A=false, AKIM_A=false;}
31	if (!digitalRead(DC)){
32	lcd.clear();
33	lcd.print("DC VOLTAJ OLCUMU");
34	AC_A=false, DC_A=true, OHM_A=false, AKIM_A=false;}
35	if (!digitalRead(OHM)){
36	lcd.clear();

```

37 lcd.print("DIRENC OLCUMU");
38 AC_A=false, DC_A=false, OHM_A=true, AKIM_A=false;}
39 if (!digitalRead(AKIM)){
40 lcd.clear();
41 lcd.print("DC AKIM OLCUMU");
42 AC_A=false, DC_A=false, OHM_A=false, AKIM_A=true;}
43 }
44 uint16_t olcum_al()
45 {
46 uint16_t gerilim = 0;
47 for(uint8_t i = 0; i < 100; i++) {
48 uint16_t ac_ger = analogRead(A2);
49 if(gerilim < ac_ger) gerilim = ac_ger;
50 delayMicroseconds(200);}
51 return gerilim;
52 }
53 //genel ayarlar
54 void setup(void)
55 {
56 lcd.begin();
57 lcd.clear();
58 lcd.setCursor(0, 0);
59 lcd.print(" AVOMETRE ");
60 lcd.setCursor(0, 1);
61 lcd.print(" UYGULAMASI ");
62 pinMode(AC, INPUT_PULLUP);
63 pinMode(DC, INPUT_PULLUP);
64 pinMode(OHM, INPUT_PULLUP);
65 pinMode(AKIM, INPUT_PULLUP);
66 pinMode(K10, INPUT_PULLUP);
67 pinMode(K100, INPUT_PULLUP);
68 pinMode(K1000, INPUT_PULLUP);
69 pinMode(R100, INPUT_PULLUP);
70 pinMode(R1K, INPUT_PULLUP);
71 pinMode(R10K, INPUT_PULLUP);
72 pinMode(R100K, INPUT_PULLUP);
73 delay(1000);
74 }
75 //ana program
76 void loop() {
77 kademe_sec ();
78 if (AC_A==true) {
79 char temp[10];
80 uint32_t ac_v = olcum_al();
81 ac_v = ac_v * 5000/1023;
82 ac_v /= sqrt(2);
83 sprintf(temp, "%01u VOLT", ac_v);
84 lcd.setCursor(0, 1);
85 lcd.print(temp);}
86 if (DC_A==true){
87 if (!digitalRead(K10)){
88 o_volt=analogRead(volt_giris);
89 voltaj=0.0048828125*o_volt*4;
90 if (voltaj>10){
91 lcd.setCursor(0, 1);
92 lcd.print("YUKSEK VOLTAJ");
93 else{
94 lcd.setCursor(0, 1);
95 lcd.print(voltaj);lcd.print(" V");}}
96 if (!digitalRead(K100)){
97 o_volt=analogRead(volt_giris);
98 voltaj=0.0048828125*o_volt*40;
99 if (voltaj>100){
100 lcd.setCursor(0, 1);
101 lcd.print("YUKSEK VOLTAJ");}
102 else{
103 lcd.setCursor(0, 1);
104 lcd.print(voltaj);lcd.print(" V");}}
105 if (!digitalRead(K1000)){
106 o_volt=analogRead(volt_giris);
107 voltaj=0.0048828125*o_volt*400;
108 if (voltaj>1010){
109 lcd.setCursor(0, 1);
110 lcd.print("YUKSEK VOLTAJ");}
111 else{
112 lcd.setCursor(0, 1);
113 lcd.print(voltaj);lcd.print(" V");}}
114 if (OHM_A==true){
115 int dir_o = analogRead(A1);
116 if(dir_o){
117 if (!digitalRead(R100)){
118 float direnc = dir_o * 5;
119 float direnc_o = (direnc)/1024.0;
120 direnc = (5/direnc_o) - 1;
121 R2 = R1 * direnc;
122 if (R2<0.9){
123 lcd.setCursor(0, 1);
124 lcd.print("KADEME AZALTIN");}
125 else if (R2>100){
126 lcd.setCursor(0, 1);
127 lcd.print("KADEME ARTIRIN");}
128 else{
129 lcd.setCursor(0, 1);
130 lcd.print(R2); lcd.print(" OHM");}}

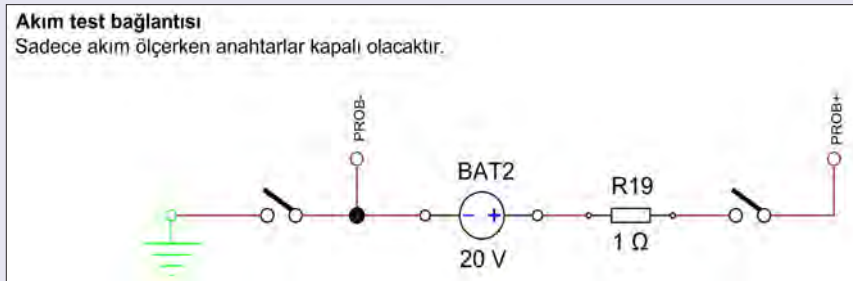
```

132	if (!digitalRead(R1K)){	153	lcd.print(R2); lcd.print(" KILO OHM");}}
133	float direnc = dir_o * 5;	154	if (!digitalRead(R100K)){
134	float direnc_o = (direnc)/1024.0;	155	float direnc = dir_o * 5;
135	direnc = (5/direnc_o) - 1;	156	float direnc_o = (direnc)/1024.0;
136	R2 = R1 * direnc*10;	157	direnc = (5/direnc_o) - 1;
137	if (R2>=1000) {	158	R2 = R1 * direnc;
138	lcd.setCursor(0, 1);	159	if (R2>1000){
139	lcd.print("KADEME ARTIRIN");}	160	R2=R2/1000;
140	else{	161	lcd.setCursor(0, 1);
141	lcd.setCursor(0, 1);	162	lcd.print(R2); lcd.print(" MEGA OHM");}
142	lcd.print(R2); lcd.print(" OHM");}}	163	else {
143	if (!digitalRead(R10K)){	164	lcd.setCursor(0, 1);
144	float direnc = dir_o * 5;	165	lcd.print(R2); lcd.print(" KILO OHM");}}}}
145	float direnc_o = (direnc)/1024.0;	166	if (AKIM_A==true){
146	direnc = (5/direnc_o) - 1;	167	float akim_a = analogRead(akima);
147	R2 = R1 * direnc/10;	168	float aout = (akim_a /100.0)*4.882;
148	if (R2>100){	169	lcd.setCursor(0, 1);
149	lcd.setCursor(0, 1);	170	lcd.print(aout); lcd.print(" AMPER");}
150	lcd.print("KADEME ARTIRIN");}	171	delay(1000);
151	else{	172	}
152	lcd.setCursor(0, 1);		

Devrenin Çalışması

ARDUINO UNO AVOMetre devresinde akım gerilim ve direnç ölçümü yapılır. ARDUINO UNO AVO-metre ile akım ölçme simülasyonu şu adımlar doğrultusunda yapılır:

- Akım test bağlantısındaki anahtarlar kapatılır (Görsel A.3).

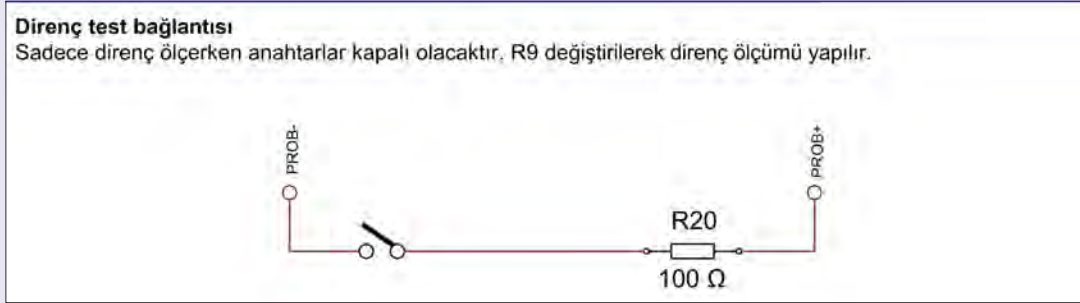


Görsel A.3: ARDUINO UNO AVOMetre ile akım ölçme

- Diğer test bağlantılarındaki anahtarlar açılır.
- SW2 3x4 rotary anahtar grubundaki tüm anahtarlar AKIM kademesine getirilir.
- Simülasyon başlatılır.
- R19 değiştirilerek akım ölçülür.

ARDUINO UNO AVOMetre ile direnç ölçme simülasyonu şu şekilde yapılır:

- Direnç test bağlantısındaki anahtarlar kapatılır (Görsel A.4).

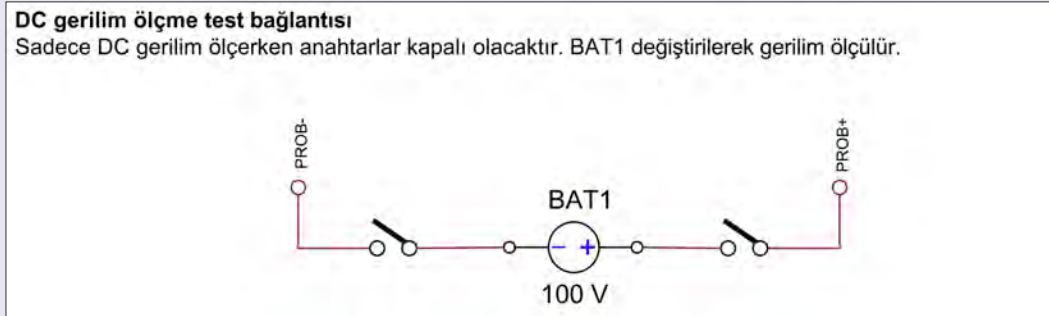


Görsel A.4: ARDUINO UNO AVOMetre ile direnç ölçme

- Diğer test bağlantılarındaki anahtarlar açılır.
- SW2 3x4 rotary anahtar grubundaki tüm anahtarlar OHM kademesine getirilir.
- SW3 rotary anahtar ile direnç ölçüm aralığı seçilir. SW3'te bulunan anahtar grupları aynı kademede olmalıdır.
- Simülasyon başlatılır.
- R20 değiştirilerek direnç ölçülür.

ARDUINO UNO AVOMetre ile DC gerilim ölçme simülasyonu şu şekilde yapılır:

- DC gerilim ölçme test bağlantısındaki anahtarlar kapatılır (Görsel A.5).



Görsel A.5: ARDUINO UNO AVOMetre ile DC gerilim ölçme

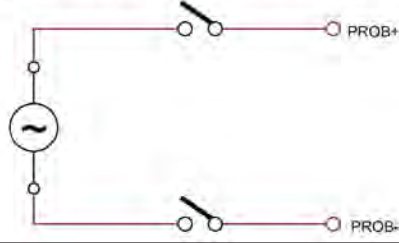
- Diğer test bağlantılarındaki anahtarlar açılır.
- SW2 3x4 rotary anahtar grubundaki tüm anahtarlar DC kademesine getirilir.
- SW1 rotary anahtar ile DC gerilim ölçüm aralığı seçilir. SW1'de bulunan anahtar grupları aynı kademede olmalıdır.
- Simülasyon başlatılır.
- BAT1 değiştirilerek DC gerilim ölçülür.

ARDUINO UNO AVOMetre ile AC gerilim ölçme simülasyonu şu şekilde yapılır:

- AC gerilim ölçme test bağlantısındaki anahtarlar kapatılır (Görsel A.6).

AC gerilim ölçme test bağlantısı

Sadece AC gerilim ölçerken anahtarlar kapalı olacaktır. Alternatör değiştirilerek gerilim ölçülür.



Görsel A.6: ARDUINO UNO AVometre ile AC gerilim ölçme

- Diğer test bağlantılarındaki anahtarlar açılır.
- SW2 3x4 rotary anahtar grubundaki tüm anahtarlar AC kademesine getirilir.
- Simülasyon başlatılır.
- BAT1 değiştirilerek DC gerilim ölçülür.
- Alternatörden AC gerilim maksimum (V_m) değeri verilir. AC ölçümde gerilimin etkin değeri (V_{rms}) ölçülür ($V_{rms}=V_m * 0,707$).

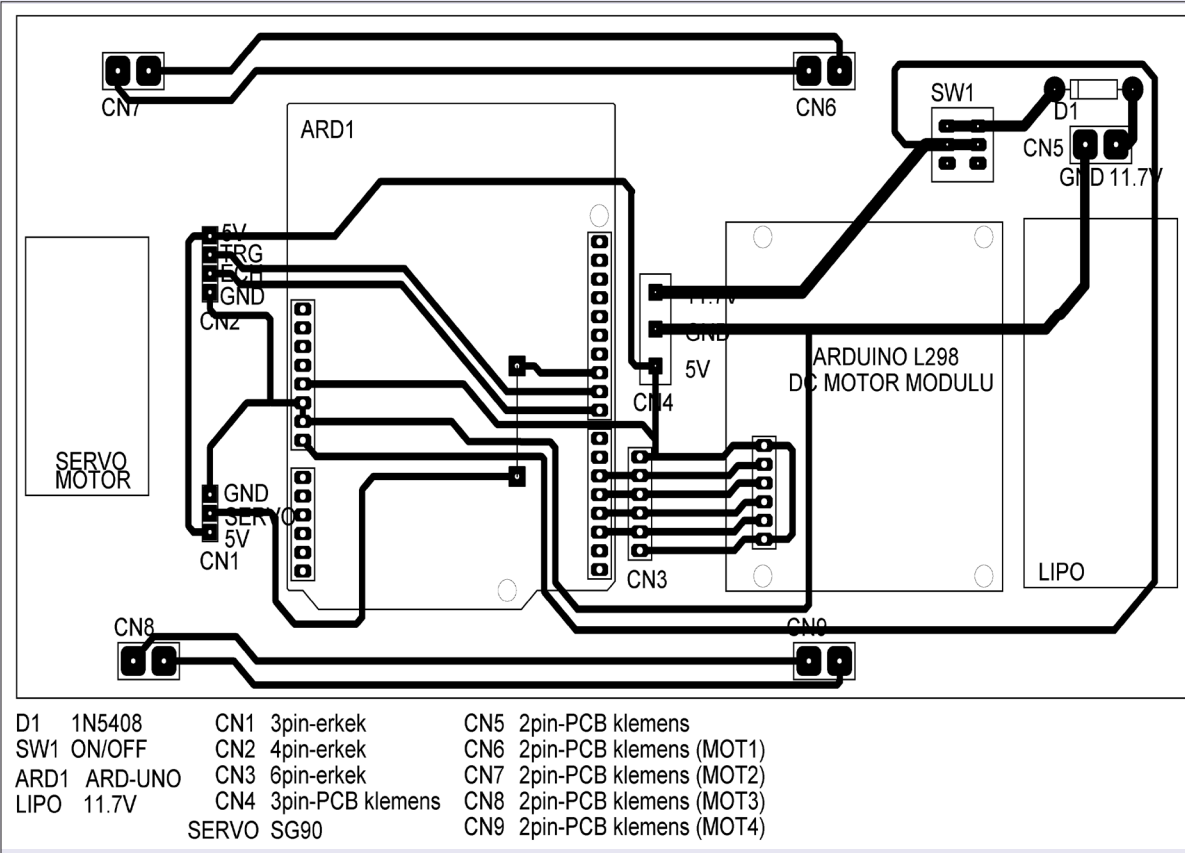
Sorular

1. Direnç ölçümünde ölçme aralığı belirlenmesin sebebi nedir?
2. DC gerilim ölçümünde ölçme aralığı belirlenmesinin sebebi nedir?
3. Ölçümlerde anahtar gruplarının aynı kademelerde olmasının sebebi nedir?

Çalışmanın Değerlendirilmesi

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

Baskı Devre Şeması: Görsel A.8’de devrenin baskı devre şeması görülmektedir.



Görsel A.8: ARDUINO UNO ile engel algılayan robot baskı devresi

Malzemeler: Tablo A.3’te devrenin malzeme listesi görülmektedir.

Tablo A.3: Devrede Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
HC-SR04	Ultrasonik mesafe ölçer	1 adet
Servo motor	SG90	1 adet
DC motor	6 V şanzımanlı	4 adet
L298 modül	ARDUINO motor sürücü modülü	1 adet
1N5408 diyot	Kristal diyot	1 adet
Lipo pil	11,7 V, 450 mA	1 adet
PCB klemens	2 pin	5 adet
PCB klemens	3 pin	1 adet
Pin header	40 pin dişi	1 adet
20cmx30cm fleksiglas	2 mm istenilen renkte	1 adet
Anahtar	ON/OFF anahtar	1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-

Kod Yazılımları: Tablo A.4’de devrenin kod yazılımı görülmektedir.

Tablo A.4: ARDUINO UNO’lu Engel Algılayan Robot Devresinin Kod Yazılımı

No.	Kodlar
1	//kütüphaneler ve tanımlamalar
2	#include <Servo.h>
3	//değişkenler
4	//L298N kontrol pinleri
5	const int Sol_ileri = 2;
6	const int Sol_geri = 3;
7	const int Sag_ileri = 4;
8	//HC-SR04 sensor pinleri
9	const int trigPin = 8;
10	const int echoPin = 9;
11	boolean ilerigit = false;
12	int mesafe = 100, cm;
13	long sure;
14	Servo servo_motor;
15	//alt programlar
16	int saga_bak(){
17	servo_motor.write(50);
18	delay(500);
19	mesafe = mesafe_oku();
20	digitalWrite(Sag_geri, LOW);
21	delay(100);
22	servo_motor.write(115);
23	return mesafe;}
24	int sola_bak(){
25	servo_motor.write(170);
26	delay(500);
27	mesafe = mesafe_oku();
28	delay(100);
29	servo_motor.write(115);
30	return mesafe;
31	delay(100);}
32	int mesafe_oku(){
33	delay(70);
34	digitalWrite(trigPin, LOW);
35	delayMicroseconds(2);
36	digitalWrite(trigPin, HIGH);
37	delayMicroseconds(10);
38	digitalWrite(trigPin, LOW);
39	sure = pulseIn(echoPin, HIGH);
40	cm= sure*0.034/2;
41	return cm;}
42	void durdur()
43	{
44	digitalWrite(Sol_ileri, LOW);
45	digitalWrite(Sol_geri, LOW);
46	digitalWrite(Sag_ileri, LOW);
47	digitalWrite(Sag_geri, LOW);
48	}
49	void ileri()
50	{
51	if(!ilerigit){
52	ilerigit=true;
53	digitalWrite(Sol_ileri, HIGH);
54	digitalWrite(Sag_ileri, HIGH);
55	digitalWrite(Sol_geri, LOW);
56	digitalWrite(Sag_geri, LOW);}
57	}
58	void geri()
59	{
60	ilerigit=false;
61	digitalWrite(Sol_geri, HIGH);
62	digitalWrite(Sag_geri, HIGH);
63	digitalWrite(Sol_ileri, LOW);
64	digitalWrite(Sag_ileri, LOW);}
65	void saga_don()
66	{
67	digitalWrite(Sol_ileri, HIGH);
68	digitalWrite(Sag_geri, HIGH);
69	digitalWrite(Sol_geri, LOW);
70	digitalWrite(Sag_ileri, LOW);
71	delay(250);
72	digitalWrite(Sol_ileri, HIGH);
73	digitalWrite(Sag_geri, HIGH);
74	digitalWrite(Sol_geri, LOW);
75	digitalWrite(Sag_ileri, LOW);}
76	void sola_don()
77	{
78	digitalWrite(Sol_geri, HIGH);
79	digitalWrite(Sag_ileri, HIGH);
80	digitalWrite(Sol_ileri, LOW);
81	digitalWrite(Sag_geri, LOW);
82	delay(250);
83	digitalWrite(Sol_geri, HIGH);
84	digitalWrite(Sag_ileri, HIGH);
85	digitalWrite(Sol_ileri, LOW);
86	digitalWrite(Sag_geri, LOW);}



87	//genel ayarlar	110	int sol_mesafe = 0;
88	void setup()	111	delay(50);
89	{Serial.begin(9600);	112	if (mesafe <= 35){
90	pinMode(Sol_ileri, OUTPUT);	113	durdur();
91	pinMode(Sol_geri, OUTPUT);	114	delay(300);
92	pinMode(Sag_ileri, OUTPUT);	115	geri();
93	pinMode(Sag_geri, OUTPUT);	116	delay(400);
94	pinMode(trigPin, OUTPUT);	117	durdur();
95	pinMode(echoPin, INPUT);	118	delay(300);
96	servo_motor.attach(10);	119	sag_mesafe = saga_bak();
97	servo_motor.write(115);	120	delay(300);
98	delay(2000);	121	sol_mesafe = sola_bak();
99	mesafe = mesafe_oku();	122	delay(300);
100	delay(100);	123	if (mesafe >= sol_mesafe){
101	mesafe = mesafe_oku();	124	saga_don();
102	delay(100);	125	durdur();}
103	mesafe = mesafe_oku();	126	else{
104	delay(100);	127	sola_don();
105	mesafe = mesafe_oku();	128	durdur();}}
106	delay(100);}	129	else{
107	//ana program	130	ileri();}
108	void loop(){	131	mesafe = mesafe_oku();
109	int sag_mesafe = 0;	132	Serial.println(mesafe);}

Devrenin Çalışması

Devrede ON/OFF anahtarı kapatıldığında önce ön taraftaki mesafe, ardından sol taraftaki mesafe ve son olarak sağ taraftaki mesafe ölçülür. Hangi taraftaki mesafe fazla ise robot o yöne doğru hareket eder. Hareket yönündeki mesafe yaklaşık 35 cm'nin altına düştüğünde mesafeler tekrar ölçülür. Hangi yönde mesafe fazlaysa robot o yöne dönerek ilerlemesine devam eder.

Sorular

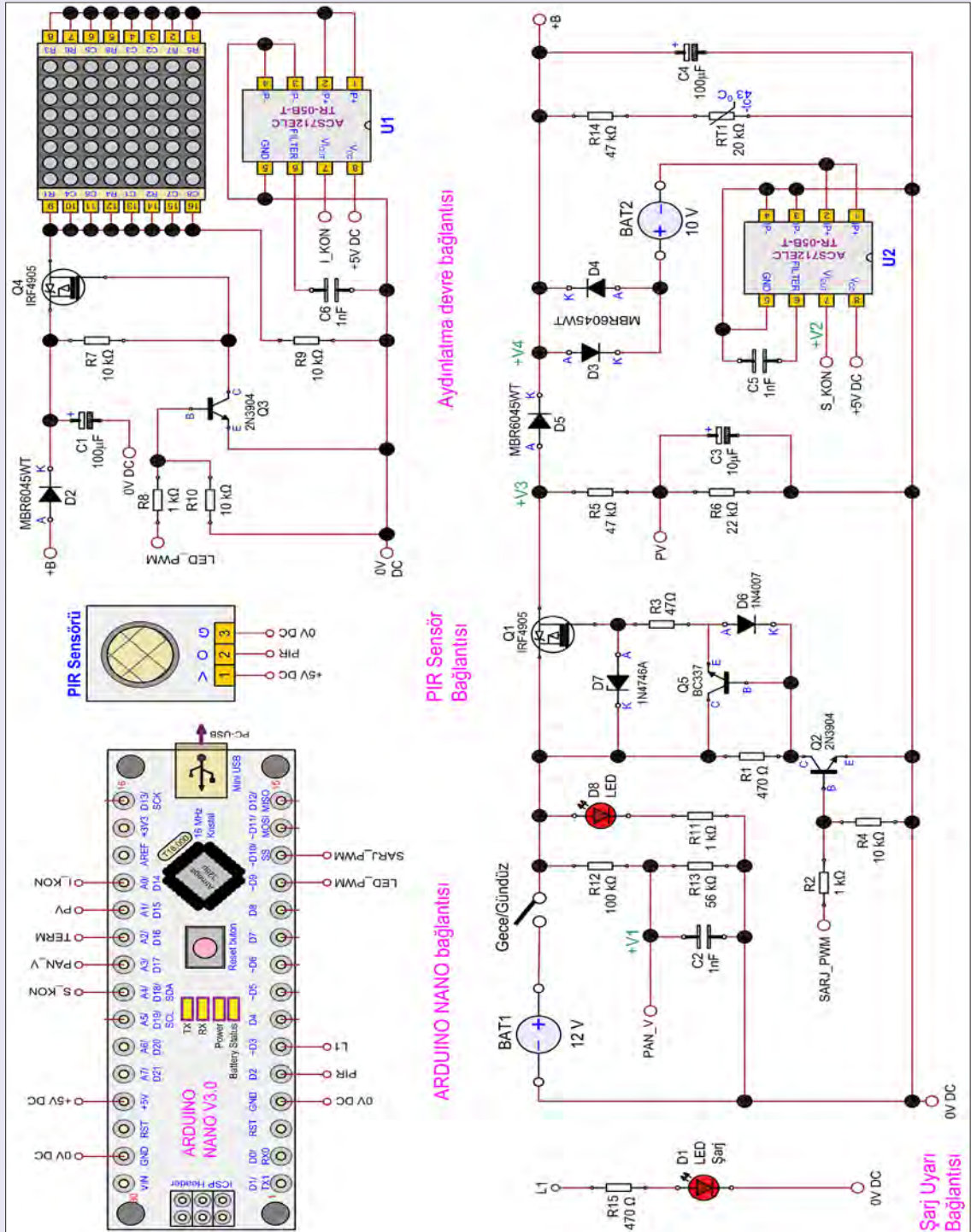
1. HC-SR04 mesafe ölçerin servo motor üzerine montajlanmasının sebebi nedir?
2. Sağ motorlar ileri, sol motorlar geri döndüğünde robot hangi yöne hareket eder?
3. Servo motorun 90 derecelik açılarla döndürülmesinin sebebi nedir?

Çalışmanın Değerlendirilmesi

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

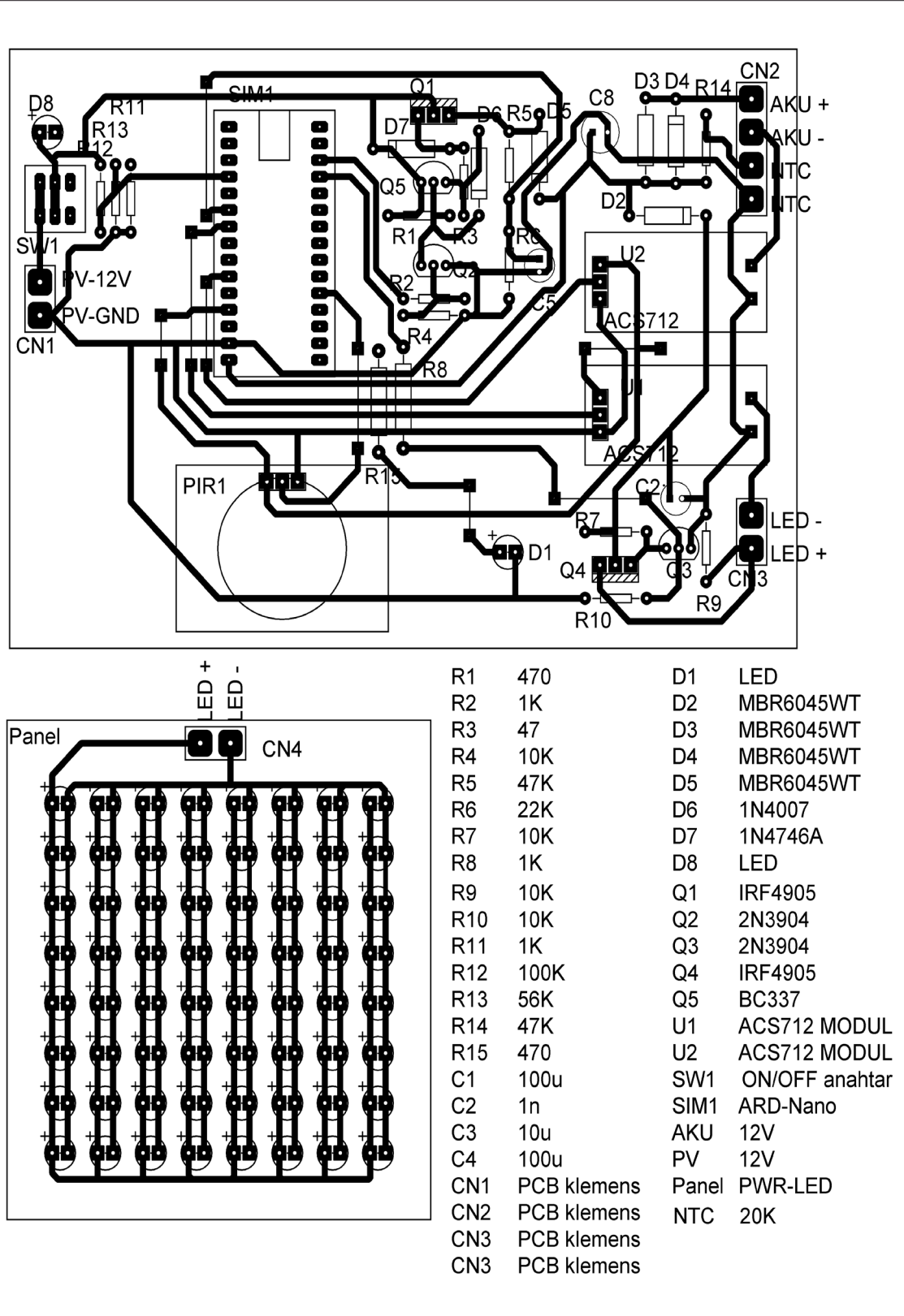
Proje Adı	PV Panel ile Sokak Aydınlatması Projesi	Proje No. : 3
Amaç	Verilen devreyi mikrodenetleyiciye göre yaparak çalıştırmak.	Süre: 7 ders saati

İstenenler: ARDUINO NANO kullanılarak PV panelli sokak aydınlatma devresi yapılacak ve isteğe göre devre, kutu içine yerleştirilecektir. ARDUINO NANO kullanılarak tasarlanan PV panel ile sokak aydınlatması devresi Görsel A.9'da görülmektedir.



Görsel A.9: ARDUINO NANO ile PV panelli sokak aydınlatması devresi

Baskı Devre Şeması: Görsel A.10'da devrenin baskı devre şeması görülmektedir.



Görsel A.10: ARDUINO NANO ile PV panelli sokak aydınlatması baskı devresi

Malzemeler: Tablo A.5'te devrenin malzeme listesi görülmektedir.

Tablo A.5: Devrede Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO NANO	Mikrodenetleyicili kart	1 adet
PIR dedektör	Dijital çıkışlı	1 adet
Akım sensörü	2XACS712	2 adet
Direnç 0,25W	47 Ω	1 adet
Direnç 0,25W	470 Ω	2 adet
Direnç 0,25W	1 k Ω	3 adet
Direnç 0,25W	10 k Ω	4 adet
Direnç 0,25W	22 k Ω	1 adet
Direnç 0,25W	47 k Ω	2 adet
Direnç 0,25W	56 k Ω	1 adet
Direnç 0,25W	100 k Ω	1 adet
Kondansatör	1nF, 10 μ F, 2x100 μ F	4 adet
LED diyot	Kırmızı, yeşil renkte	2 adet
MBR6045WT diyot	Şotki	4 adet
1N4007 diyot	Kristal diyot	1 adet
1N4746A diyot	5 V zener diyot	1 adet
Transistör	BC337, 2x2N3904	3 adet
Mosfet	2xIRF4905	2 adet
Akü	11.7V 3S şarjlı pil (BAT2)	1 adet
Güneş paneli	12 V (BAT1)	1 adet
PWR-LED	5 mm beyaz	64 adet
NTC	20 k Ω	1 adet
PCB klemens	2 pin	2 adet
PCB klemens	4 pin	1 adet
Anahtar	ON/OFF anahtar	1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-

Kod Yazılımları: Tablo A.6'da devrenin kod yazılımı görülmektedir.

Tablo A.6: ARDUINO UNO'lu Engel Algılayan Robot Devresinin Kod Yazılımı

No.	Kodlar
1	//kütüphaneler ve tanımlamalar
2	#define led 9
3	#define sarj 10
4	#define pir_dedektor 2
5	#define sistem_ok 3
6	#define akim_kont A0
7	#define mos_aktif A1
8	#define pil_isi A2
9	//#define panel_gerilim A3
10	#define sarj_akim A4



```
11 //değişkenler
12 int panel_g=0, mos, sıcaklik, akim, yuk_akim;
13 int panel_gerilim=A3;
14 //genel ayarlar
15 void setup()
16 {
17   pinMode(led, OUTPUT);
18   pinMode(sistem_ok, OUTPUT);
19   pinMode(sarj, OUTPUT);
20   pinMode(pir_dedektor, INPUT_PULLUP);
21   Serial.begin(9600);
22 }
23 //ana program
24 void loop()
25 {
26   panel_g=analogRead(panel_gerilim);
27   panel_g=map(panel_g,0,1023,0,255);
28   if (panel_g>200) {
29     analogWrite (led,0);
30     sıcaklik=analogRead(pil_isi);
31     sıcaklik=map(sıcaklik,0,1023,0,255);
32     if (sıcaklik<70){
33       digitalWrite(sistem_ok,LOW);
34       analogWrite (sarj,0);    }
35     else{
36       akim=analogRead(sarj_akim);
37       akim=map(akim,0,1023,0,255);
38       Serial.println(akim);
39       if (akim>150){
40         delay(500);
41         akim=analogRead(sarj_akim);
42         akim=map(akim,0,1023,0,255);
43         if (akim>150){
44           digitalWrite(sistem_ok,LOW);
45           analogWrite (sarj,0);}}
46         else{
47           analogWrite (sarj,125);
48           delay(10);
49           mos=analogRead(mos_aktif);
50           mos=map(mos,0,1023,0,255);
51           if (mos>50){
52             digitalWrite(sistem_ok,HIGH);}
53           else{
54             digitalWrite(sistem_ok,LOW);
55             analogWrite (sarj,0);}}}}
56         else{
57           digitalWrite(sistem_ok,LOW);
```

```

58 analogWrite (sarj,0);
59 if (digitalRead(pir_dedektor)){
60 analogWrite (led,125);
61 yuk_akim=analogRead(akim_kont);
62 yuk_akim=map(yuk_akim,0,1023,0,255);
63 if (yuk_akim>130){
64 delay(250);
65 yuk_akim=analogRead(akim_kont);
66 yuk_akim=map(yuk_akim,0,1023,0,255);
67 if (yuk_akim>130){
68 analogWrite (led,0);}}}
69 else{
70 analogWrite (led,0);}}
71 }

```

Devrenin Çalışması

Gece/gündüz anahtarı kapalı durumda gündüzü, açık durumunda geceyi simüle eder. Anahtar gündüz konumundayken akü (BAT2) şarja başlar. RT1, akünün sıcaklığını ölçer. Akü aşırı ısınır (60 derece) ARDUINO NANO şarjı keser. Akü sıcaklığı 45 dereceye düştüğünde tekrar şarja başlanır. U2, aküden geçen şarj akımını ölçer. Şarj akımı yüksekse ARDUINO NANO şarjı keser. Şarj göstergesi D1 LED diyotudur. Bu diyot şarj esnasında ışık verir. Anahtar gece konumundayken PIR1 sensörü çıkışından (O) lojik 1 bilgisi gelince (logicstate 1 konumuna alınır.) LED panel yanar. PIR1 çıkışı (O) lojik 0 olunca ya kadar ışık verir (logicstate 0 konumuna getirilir, PIR1 sensör çıkışının lojik 0 olması beklenir.). U1 dot, LED panelden geçen akımı ölçer. LED panelden aşırı akım geçmesi durumunda ARDUINO NANO LED panel voltajını keser. LED panel yanarken anahtar gündüz konumuna alınır LED panel voltajı kesilir ve akü şarjına başlanır.

Sorular

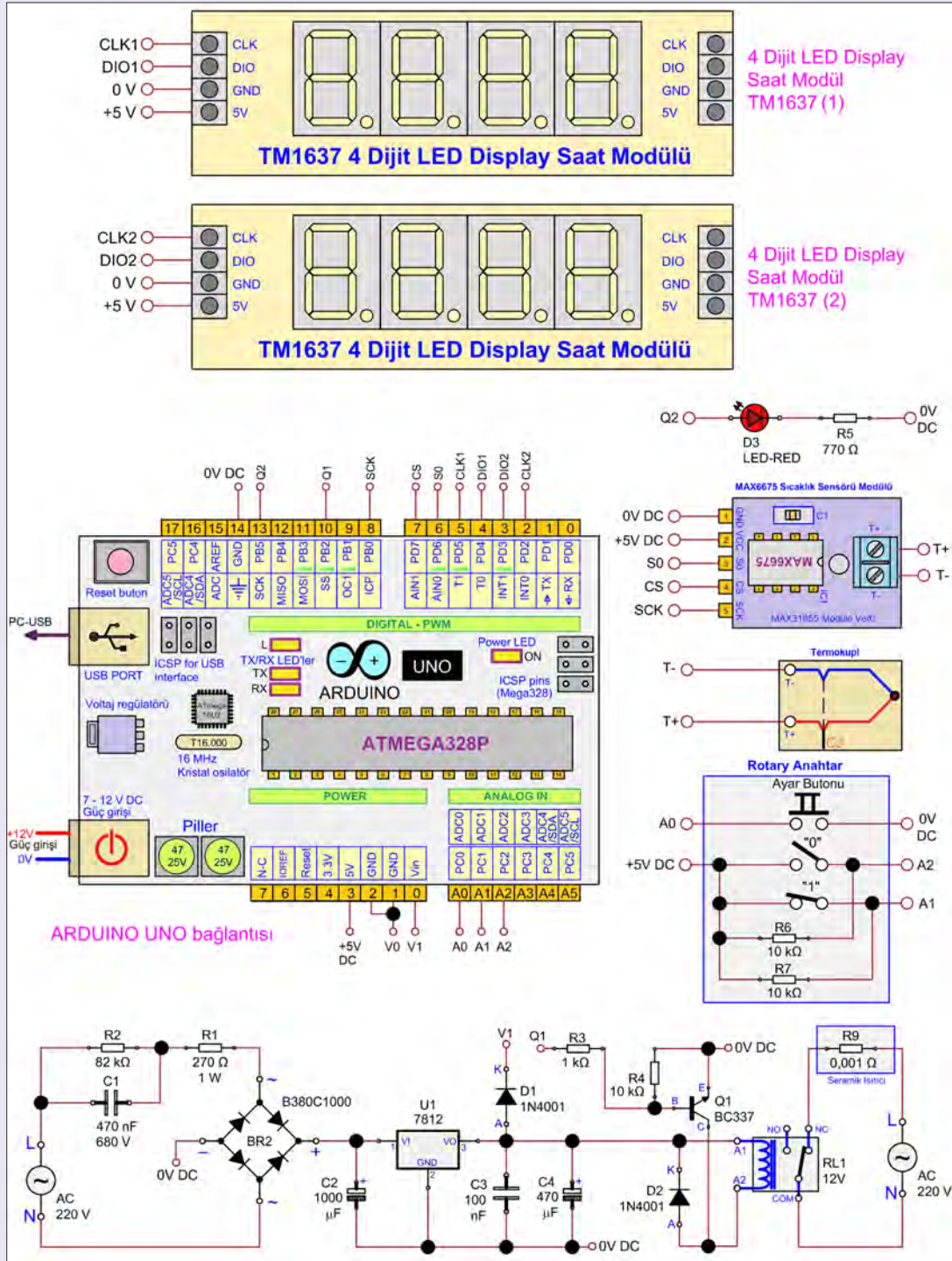
1. LED panel ve şarj akımının PWM yöntemiyle sağlanmasında amaç nedir?
2. PIR dedektörün görevi nedir?
3. Devrede kullanılan NTC'nin görevi nedir?

Çalışmanın Değerlendirilmesi

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroislemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

Proje Adı	SMD Eleman Lehimleme ve Sökme Cihazı Projesi	Proje No. : 4
Amaç	Verilen devreyi mikrodnetleyiciye göre yaparak çalıştırmak.	Süre: 7 ders saati

İstenenler: ARDUINO UNO kullanılarak SMD lehimleme ve lehim sökme devresi yapılacak ve isteğe göre devre, kutu içerisine yerleştirilecektir. ARDUINO UNO kullanılarak tasarlanan SMD eleman lehimleme ve sökme devresi Görsel A.11’de görülmektedir.



Görsel A.11: ARDUINO UNO ile SMD lehimleme ve sökme devresi



Kod Yazılımları: Tablo A.8’de devrenin kod yazılımı görülmektedir.

Tablo A.8: ARDUINO UNO’lu SMD Lehimleme ve Sökme Devresi Kod Yazılımları

No.	Kodlar
1	//kütüphaneler ve tanımlamalar
2	#include <max6675.h>
3	#include <ARDUINO.h>
4	#include <ATM1637Display.h>
5	#define CLK 2
6	#define DIO 3
7	#define CLKA 4
8	#define DIOA 5
9	#define bekle 2000
10	//değişkenler
11	const int ayar_buton = A0;
12	const int CLKBUT = A1;
13	const int DTBUT = A2;
14	const int ROLE = 10;
15	int counter = 0, onlar=0, onlar_t=0;
16	int simdikiCLK;
17	int oncekiCLK;
18	String currentDir ="";
19	unsigned long lastButtonPress = 0;
20	uint8_t data[] = { 0xff, 0xff, 0xff, 0xff};
21	uint8_t SG_DN[] = {
22	SEG_G, // -
23	SEG_G, // -
24	SEG_G, // -
25	SEG_G // -
26	};
27	int thermoDO = 6;
28	int thermoCS = 7;
29	int thermoCLK = 8;
30	MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);
31	ATM1637Display display(CLKA, DIOA);
32	//genel ayarlar
33	void setup()
34	{
35	Serial.begin(9600);
36	pinMode(ayar_buton, INPUT_PULLUP);
37	pinMode(CLKBUT, INPUT);
38	pinMode(DTBUT, INPUT);
39	pinMode(ROLE, OUTPUT);
40	pinMode(13, OUTPUT);
41	oncekiCLK = digitalRead(CLKBUT);
42	display.setBrightness(0x0f);
43	display.setSegments(SG_DN);

```

44 ATM1637Display display(CLK, DIO);
45 display.setBrightness(0x0f);
46 display.setSegments(SG_DN);
47 }
48 //ana program
49 void loop()
50 {
51 if(!digitalRead(ayar_buton))
52 {
53 digitalWrite(13,HIGH);
54 delay(50);
55 while(!digitalRead(ayar_buton));
56 delay(50);
57 while(digitalRead(ayar_buton))
58 {
59 simdikiCLK = digitalRead(CLKBUT);
60 if (simdikiCLK != oncekiCLK && simdikiCLK == 1){
61 if (digitalRead(DTBUT) != simdikiCLK) {
62 counter ++;
63 if (counter>400){
64 counter=400;}}
65 else {
66 counter --;
67 if (counter<=0){
68 counter=0;}}
69 ATM1637Display display(CLK, DIO);
70 display.setBrightness(0x0f);
71 data[0] = display.encodeDigit(counter/100);
72 onlar=counter%100;
73 data[1] = display.encodeDigit(onlar/10);
74 data[2] = display.encodeDigit(onlar%10);
75 data[3] = 0x39;
76 display.setSegments(data);}
77 oncekiCLK = simdikiCLK; }
78 while(!digitalRead(ayar_buton));
79 delay(500); }
80 digitalWrite(13,LOW);
81 int temp=thermocouple.readCelsius();
82 Serial.println(temp);
83 ATM1637Display display(CLKA, DIOA);
84 display.setBrightness(0x0f);
85 SG_DN[0] = display.encodeDigit(temp/100);
86 onlar_t=temp%100;
87 SG_DN[1] = display.encodeDigit(onlar_t/10);
88 SG_DN[2] = display.encodeDigit(onlar_t%10);
89 SG_DN[3] = 0x39;
90 display.setSegments(SG_DN);

```



```
91 delay(1000);
92 if (temp>=counter)
93 digitalWrite(ROLE,LOW);
94 else
95 digitalWrite(ROLE,HIGH);
96 }
```

Devrenin Çalışması

Simülasyon başlatıldığında 1. ekranda seramik ısıtıcının sıcaklığı görülür. 2. ekranda ayarlanacak sıcaklık değeri beklenir ve ekranda “----” değeri görülür. Sıcaklık ayarı için sıcaklık ayar butonu basılı tutulur (rotary anahtar butonu) ve ikaz LED yanar. Rotary anahtar sağa doğru çevrilerek sıcaklık değeri artırılır, sola doğru çevrilerek sıcaklık değeri düşürülür. Simülatörde rotary anahtar B girişi lojik 0 iken A girişi sırasıyla lojik 1, lojik 0 yapılırsa sıcaklık değeri artar. Rotary anahtar B girişi lojik 1 iken A girişi sırasıyla lojik 1, lojik 0 yapılırsa sıcaklık değeri azalır. İstenilen sıcaklık derecesi ayarlanır. Sıcaklık ayar butonuna basılıp bırakılır, ikaz LED’i söner. Ayarlanan sıcaklık değeri ısıtıcı sıcaklık değerinden büyükse röle kontakları kapanır ve ısıtıcı çalışmaya başlar. Isıtıcı sıcaklığı ayarlanan sıcaklık değerine yükseldiğinde röle kontakları açılır.

Sorular

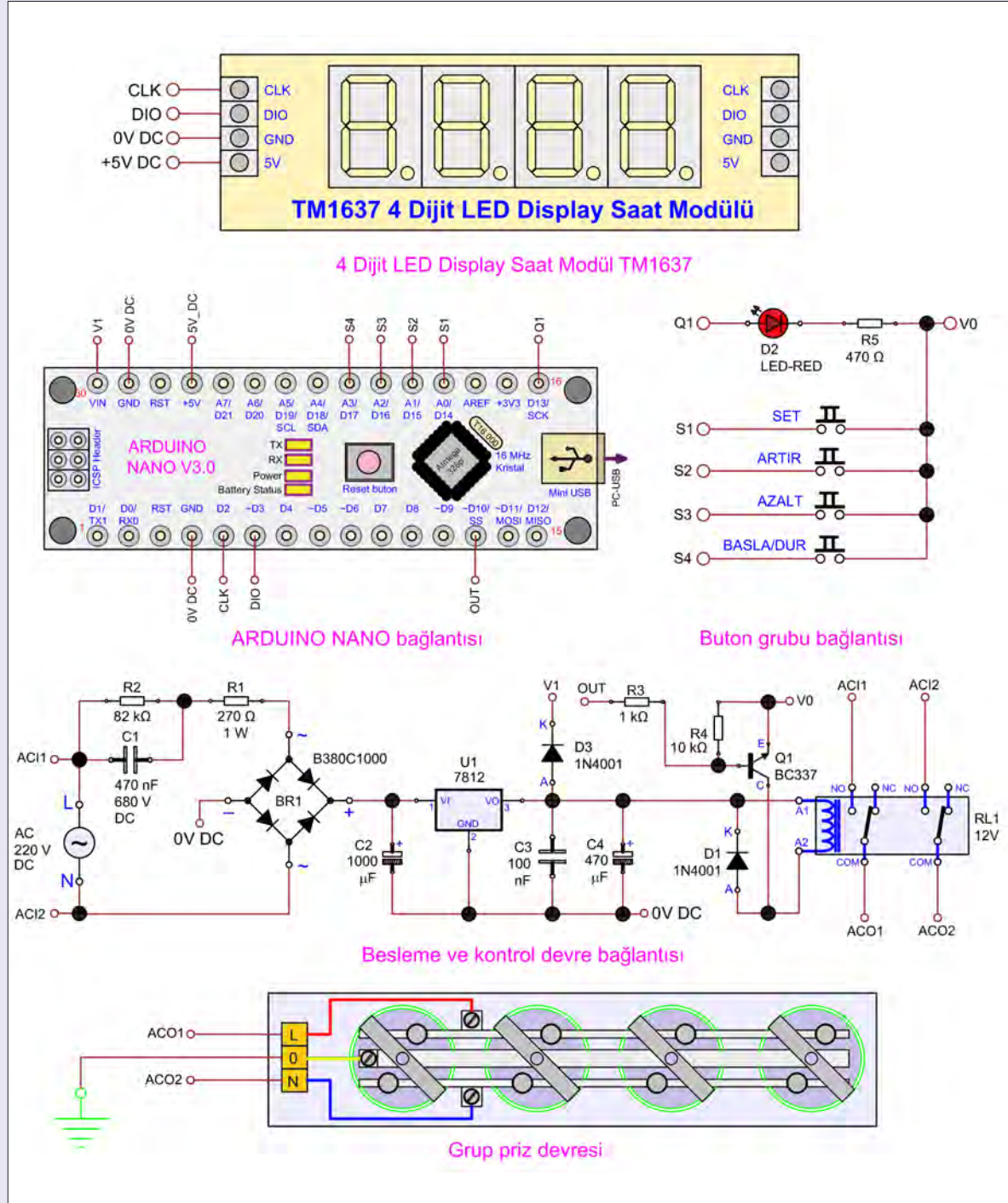
1. MAX6675 sıcaklık sensörünün görevi nedir?
2. Seramik ısıtıcının röle kontakları üzerinden çalıştırılmasının sebebi nedir?
3. Devredeki BC337 transistörünün görevi nedir?

Çalışmanın Değerlendirilmesi

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

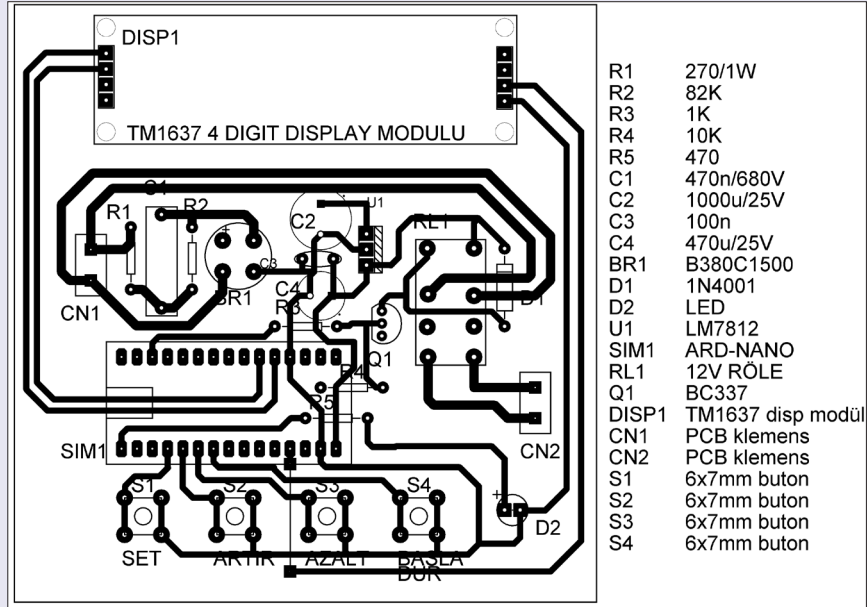
Proje Adı	Göstergeli Zaman Ayarlı Dijital Grup Priz Projesi	Proje No. : 5
Amaç	Verilen devreyi mikrodenetleyiciye göre yaparak çalıştırmak.	Süre: 7 ders saati

İstenenler: ARDUINO NANO kullanılarak göstergeli zaman ayarlı grup priz devresi yapılacak ve isteğe göre bu devre, kutu içine yerleştirilecektir. ARDUINO NANO kullanılarak tasarlanan göstergeli zaman ayarlı dijital grup priz devresi Görsel A.13'te görülmektedir.



Görsel A.13: ARDUINO NANO ile göstergeli zaman ayarlı dijital grup priz devresi

Göstergeli zaman ayarlı dijital grup priz devresinin baskı devresi Görsel A.14'te görülmektedir.



Görsel A.14: ARDUINO NANO ile göstergeli zaman ayarlı dijital grup priz devresi baskı devresi

Malzemeler: Tablo A.9'da devrenin malzeme listesi görülmektedir.

Tablo A.9: Devrede Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO NANO	Mikrodenetleyicili kart	1 adet
TM1637 ekran modülü	4 digit	1 adet
Direnç 0,25W	470 Ω	1 adet
Direnç 0,25W	1 k Ω	1 adet
Direnç 0,25W	10 k Ω	3 adet
Direnç 0,25W	82 k Ω	1 adet
Direnç 1W	270 Ω	1 adet
Kondansatör	100 nF	1 adet
Kondansatör	470 nF / 680 V	1 adet
Kondansatör	470 μ F / 25 V	1 adet
Kondansatör	1000 μ F / 25 V	1 adet
LED diyot	Kırmızı	1 adet
1N4001 diyot	Kristal diyot	2 adet
B380C1500 diyot	Köprü diyot	1 adet
Anahtar	6x7 mm mikro anahtar	1 adet
Transistör	BC337	1 adet
LM7812 entegre	12 V gerilim regülatörü	1 adet
Röle	12 V çift kontak	1 adet
PCB klemens	2 pin	2 adet
Pin header	40 pin dişi	1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-

Kod Yazılımları: Tablo A.10'da devrenin kod yazılımı görülmektedir.

Tablo A.10: ARDUINO NANO'lu Göstergeli Zaman Ayarlı Dijital Grup Priz Devresinin Kod Yazılımları

No.	Kodlar
1	//kütüphaneler ve tanımlamalar
2	#include <ARDUINO.h>
3	#include <ATM1637Display.h>
4	// Modül bağlantı pinleri (Dijital Pinler)
5	#define CLK 2
6	#define DIO 3
7	#define bekle 2000
8	//değişkenler
9	const int ayar_buton = A0;
10	const int artir = A1;
11	const int azalt = A2;
12	const int basla = A3;
13	const int ROLE = 10;
14	int saat = 0, dakika=0, saatvar=0, dakvar=0, nokta=0, sayi=0;
15	uint8_t data[] = { 0xff, 0xff, 0xff, 0xff};
16	uint8_t SG_DN[] = {
17	SEG_G, // -
18	SEG_G, // -
19	SEG_G, // -
20	SEG_G // -};
21	//alt programlar
22	void setupTimer1()
23	{noInterrupts();
24	TCCR1A = 0;
25	TCCR1B = 0;
26	TCNT1 = 0;
27	// 0.5 Hz (16000000/((15624+1)*512))
28	OCR1A = 7812;
29	TCCR1B = (1 << WGM12);
30	TCCR1B = (1 << CS12) (1 << CS10);
31	TIMSK1 = (1 << OCIE1A);}
32	ISR(TIMER1_COMPA_vect)
33	{noInterrupts();
34	sayi++;
35	if (sayi>=120){
36	sayi=0;
37	dakika--;
38	if (dakika<0){
39	dakika=59;
40	saat--;
41	if (saat<0){
42	dakika=0;
43	saat=0;}}
44	nokta=!nokta;
45	interrupts();}
46	void ekran()
47	{ATM1637Display display(CLK, DIO);
48	display.setBrightness(0x0f);
49	if (saatvar){
50	data[0] = display.encodeDigit(saat/10);
51	data[1] = display.encodeDigit(saat%10);
52	data[2] = SEG_G;
53	data[3] = SEG_G;
54	display.setSegments(data);}
55	if (dakvar){
56	data[0] = SEG_G;
57	data[1] = SEG_G;
58	data[2] = display.encodeDigit(dakika/10);
59	data[3] = display.encodeDigit(dakika%10);
60	display.setSegments(data);}
61	if ((!dakvar) && (!saatvar)){
62	data[0] = display.encodeDigit(saat/10);
63	data[1] = display.encodeDigit(saat%10);
64	if (nokta){
65	bitSet (data[1],7);}
66	data[2] = display.encodeDigit(dakika/10);
67	data[3] = display.encodeDigit(dakika%10);
68	display.setSegments(data);}
69	//genel ayarlar
70	void setup()
71	{pinMode(ayar_buton, INPUT_PULLUP);
72	pinMode(artir, INPUT_PULLUP);
73	pinMode(azalt, INPUT_PULLUP);
74	pinMode(basla, INPUT_PULLUP);
75	pinMode(ROLE, OUTPUT);
76	pinMode(13, OUTPUT);
77	ATM1637Display display(CLK, DIO);
78	display.setBrightness(0x0f);
79	display.setSegments(SG_DN);
80	delay(1000);
81	setupTimer1();}
82	//ana program
83	void loop()
84	{if(!digitalRead(ayar_buton)){
85	digitalWrite(13,HIGH);
86	delay(50);



```
87  saatvar=1;
88  ekran();
89  while(!digitalRead(ayar_buton));
90  delay(50);
91  while(digitalRead(ayar_buton)){
92  if (!digitalRead(artir)){
93  saatvar=1;
94  saat ++;
95  if (saat>23){
96  saat=0;}
97  while(!digitalRead(artir)){
98  ekran();}}
99  if (!digitalRead(azalt)){
100 saatvar=1;
101 saat --;
102 if (saat<=0){
103 saat=0;}
104 while(!digitalRead(azalt)){
105 ekran();}}}
106 while(!digitalRead(ayar_buton));
107 saatvar=0;
108 dakvar=1;
109 ekran();
110 while(digitalRead(ayar_buton)){
111 if (!digitalRead(artir)){
112 dakvar=1;
113 dakika ++;
114 if (dakika>59){
115 dakika=0;}
116 while(!digitalRead(artir)){
117 ekran();}}
118 if (!digitalRead(azalt)){
119 dakvar=1;
120 dakika --;
121 if (dakika<=0){
122 dakika=0;}
123 while(!digitalRead(azalt)){
124 ekran();}}}
125 while(!digitalRead(ayar_buton));
126 dakvar=0;
127 digitalWrite(13,LOW);
128 ekran();
129 delay(250);
130 if (!digitalRead(basla)){
131 if ((dakika>0) || (saat>0)){
132 digitalWrite(ROLE, !digitalRead(ROLE));
133 interrupts();
```

```

134 else
135 digitalWrite(ROLE, LOW);
136 noInterrupts();}
137 while(!digitalRead(basla)){
138 ekran();}
139 if ((!saat) && (!dakika)){
140 digitalWrite(ROLE, LOW);
141 noInterrupts();}

```

Devrenin Çalışması

Simülasyon başlatıldığında ekranda önce “----” değeri sonra 0000 rakamları görülür. Süre ayarı için SET butonuna basılı tutulur ve ayar LED’i (D2) yanar. Ekranda saat bilgisi girilmesi beklenir. ARTIR, AZALT butonlarıyla saat ayarlanır. SET butonuna basılır. Ekranda dakika bilgisi girilmesi beklenir. ARTIR, AZALT butonlarıyla dakika ayarlanır. SET butonuna basılır. Ayar LED’i (D2) söner. BASLA/DUR butonuna basılarak şehir şebeke gerilimi, röle kontakları üzerinden grup prize yönlendirilir. Sayıcı geri saymaya başlar. Süre sonunda saat ve dakika sıfırlanınca röle, kontaklarını bırakır ve bunun sonucunda grup prizdeki gerilim otomatikman kesilir. Çalışma esnasında BASLA/DUR butonuna basılırsa grup priz gerilimi kesilir. Geri sayma işlemi durur.

Sorular

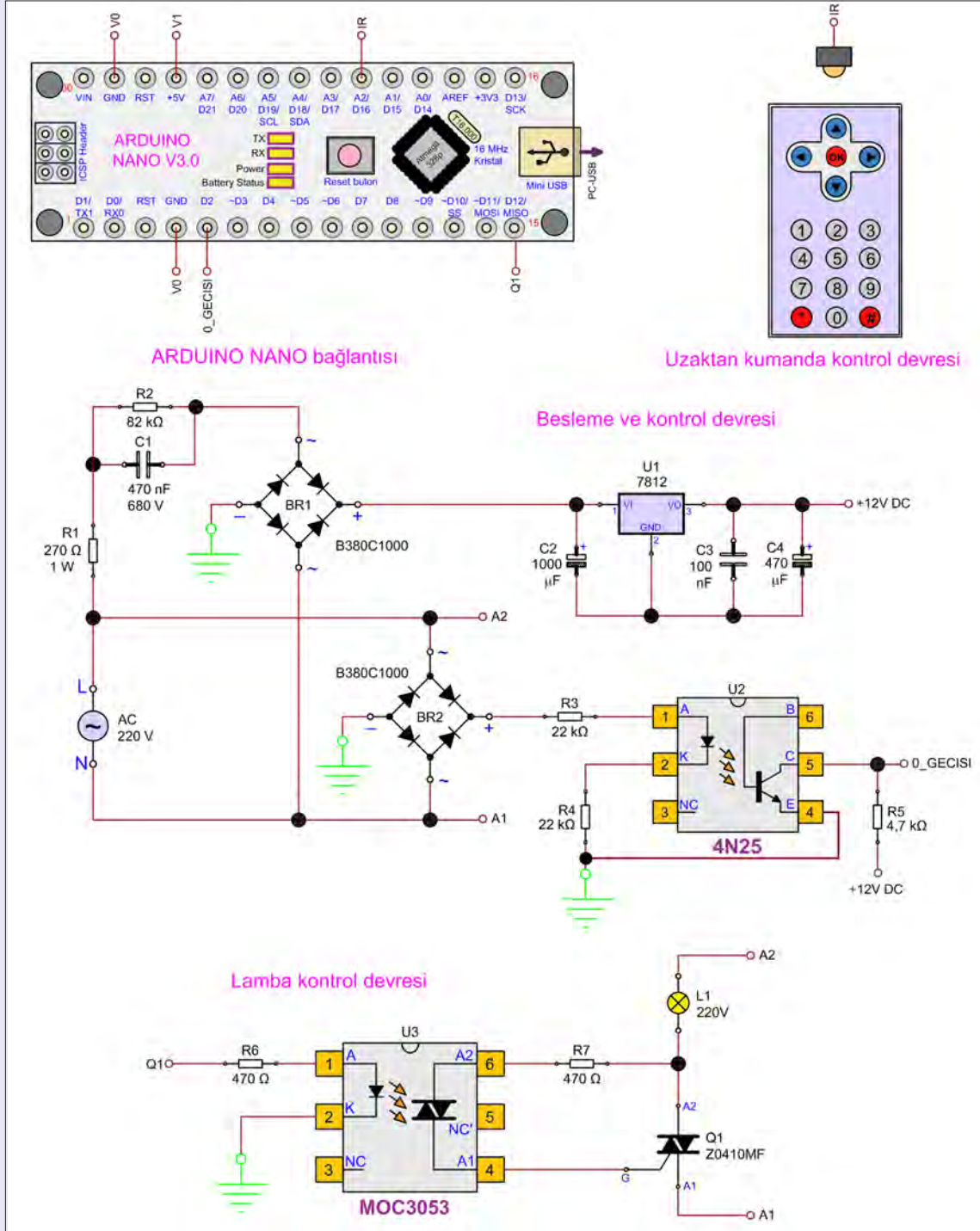
1. Grup priz projesinde çift kontaklı röle kullanılmasının sebebi nedir?
2. Devrenin tasarımını örnek aydınlatma sistemi üzerinden yapınız.

Çalışmanın Değerlendirilmesi

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

Proje Adı	Dimmerlenen Uzaktan Kontrollü Lamba Projesi	Proje No. : 6
Amaç	Verilen devreyi mikrodnetleyiciye göre yaparak çalıştırmak.	Süre: 7 ders saati

İstenenler: ARDUINO NANO kullanılarak dimmerlenen uzaktan kontrollü lamba devresi yapılacak ve isteğe göre bu devre, kutu içine yerleştirilecektir. ARDUINO NANO kullanılarak tasarlanan dimmerlenen uzaktan kontrollü lamba devresi Görsel A.15'te görülmektedir.



Görsel A.15: Dimmerlenen uzaktan kontrollü lamba devresi



Kod Yazılımları: Tablo A.12’de devrenin kod yazılımları görülmektedir.

Tablo A.12: ARDUINO NANO’lu Dimmerlenen Uzaktan Kontrollü Lamba Devresinin Kod Yazılımları

No.	Kodlar
1	//kütüphaneler ve tanımlamalar
2	#include <IRremote.h>
3	#define triyak 12
4	//değişkenler
5	const int RECV_PIN = A2;
6	const byte interruptPin = 2;
7	bool gecis = 0;
8	uint16_t tetikleme=0;
9	IRrecv irrecv(RECV_PIN);
10	decode_results results;
11	//alt programlar
12	void sifirgecis()
13	{gecis = 1;}
14	//genel ayarlar
15	void setup()
16	{
17	pinMode(12, OUTPUT);
18	pinMode(triyak, OUTPUT);
19	digitalWrite(triyak, LOW);
20	attachInterrupt(digitalPinToInterrupt(interruptPin), sifirgecis, FALLING);
21	Serial.begin(9600);
22	irrecv.enableIRIn();}
23	//ana program
24	void loop()
25	{
26	if (irrecv.decode(&results)){
27	Serial.println(results.value, HEX);
28	switch(results.value){
29	case 0xFF6897:
30	tetikleme =6000;
31	break;
32	case 0xFF9867:
33	tetikleme =4000;
34	break;
35	case 0xFFB04F:
36	tetikleme =2000;
37	break;
38	case 0xFF30CF:
39	tetikleme =1000;
40	break;
41	case 0xFF18E7:
42	tetikleme =500;
43	break;

```

44 case 0xFF02FD:
45   tetikleme =0;
46   break;
47 default:
48   break;
49   results.value=0;}
50   irrecv.resume();
51   if(gecis){
52     if(tetikleme > 0 && tetikleme <= 9500) {
53       delayMicroseconds(tetikleme);
54       digitalWrite(triyak, HIGH);
55       delayMicroseconds(200);
56       digitalWrite(triyak, LOW);
57       gecis = 0;}
58   else{
59     digitalWrite(triyak, LOW);}}

```

Devrenin Çalışması

Simülasyon başlatıldığında lamba ışık vermez. Uzaktan kumandanın 1, 2, 3, 4 ve 5 tuşlarına basılarak lambanın parlaklığı ayarlanır. Kumandadan OK tuşuna basılarak lamba söndürülür.

Sorular

1. Lambanın parlaklığını değiştirmede kullanılan triyak sürücü metodu nedir?
2. 4N25 optokuplörün görevi nedir?

Çalışmanın Değerlendirilmesi

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroislemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



Kod Yazılımları: Tablo A.14'te devrenin kod yazılımları görülmektedir.

Tablo A.14: ARDUINO UNO'lu Robot Kol Kontrol Devresinin Kod Yazılımları

No.	Kodlar
1	//kütüphaneler ve tanımlamalar
2	#include "ARDUINO.h"
3	#include "JoystickLib.h"
4	Joystick stick(A0, A1);
5	#include <Servo.h>
6	Servo myservo;
7	#include <Stepper.h>
8	#define MOT1A 11
9	#define MOT1B 10
10	#define MOT2A 9
11	#define MOT2B 8
12	#define SINIR1 3
13	#define SINIR2 2
14	#define ADIM 20
15	//değişkenler
16	const int DC = A2;
17	const int STEP = A3;
18	const int SERVO = A4;
19	int pos = 0;
20	Stepper stepper(ADIM, 7, 6, 5, 4);
21	//alt programlar
22	void dc_motor(int a, int b, int c, int d)
23	{
24	digitalWrite(MOT1A,a);
25	digitalWrite(MOT1B,b);
26	digitalWrite(MOT2A,c);
27	digitalWrite(MOT2B,d);
28	}
29	//genel ayarlar
30	void setup()
31	{Serial.begin(9600);
32	myservo.attach(12);
33	pinMode(SINIR1, INPUT_PULLUP);
34	pinMode(SINIR2, INPUT_PULLUP);
35	pinMode(DC, INPUT_PULLUP);
36	pinMode(STEP, INPUT_PULLUP);
37	pinMode(SERVO, INPUT_PULLUP);
38	pinMode(MOT1A, OUTPUT);
39	pinMode(MOT1B, OUTPUT);
40	pinMode(MOT2A, OUTPUT);
41	pinMode(MOT2B, OUTPUT);
42	stick.calibrate();}
43	//ana program
44	void loop()
45	{if(!digitalRead(DC)){
46	stick.loop();
47	if (stick.isUp()) {
48	dc_motor(LOW, HIGH, LOW,HIGH);
49	delay(500);}
50	if (stick.isRight()) {
51	dc_motor(LOW, HIGH, HIGH,LOW);
52	delay(500);}
53	if (stick.isDown()) {
54	dc_motor(HIGH, LOW, HIGH,LOW);
55	delay(500);}
56	if (stick.isLeft()) {
57	dc_motor(HIGH, LOW, LOW,HIGH);
58	delay(500); }
59	dc_motor(LOW, LOW, LOW,LOW);}
60	if(!digitalRead(STEP)){
61	stick.loop();
62	if (digitalRead(SINIR1)){
63	if (stick.isUp()) {
64	int yon = 1;
65	stepper.setSpeed(100);
66	stepper.step(yon);
67	delay(50);}}
68	if (digitalRead(SINIR2)){
69	if (stick.isDown()) {
70	int yon = -1;
71	stepper.setSpeed(100);
72	stepper.step(yon);
73	delay(50);}}}
74	if(!digitalRead(SERVO)){
75	stick.loop();
76	if (stick.isUp()) {
77	myservo.write(180);
78	delay(500);}
79	if (stick.isDown()) {
80	myservo.write(0);
81	delay(500);}}}

Devrenin Çalışması

Simülasyon başlatılır. DC anahtar kapalı (GND) konumda diğer anahtarlar açık konumdayken joystick yardımıyla robot kol platformu sağa sola ve ileri geri hareket ettirilir. STEP anahtar kapalı (GND) konumda diğer anahtarlar açık konumdayken joystick yardımıyla robot kol aşağı ve yukarı hareket ettirilir. SNR1 ve SNR2 sınır anahtarları robot kolun aşağı ve yukarı hareketinde sınırları belirler. Sınır anahtarları kapalı konuma geldiğinde step motor durur. SERVO anahtar kapalı (GND) konumda diğer anahtarlar açık konumdayken joystick ileri geri hareket ettirilerek robot kol parça tutma mekanizması açılıp kapanır.

Sorular

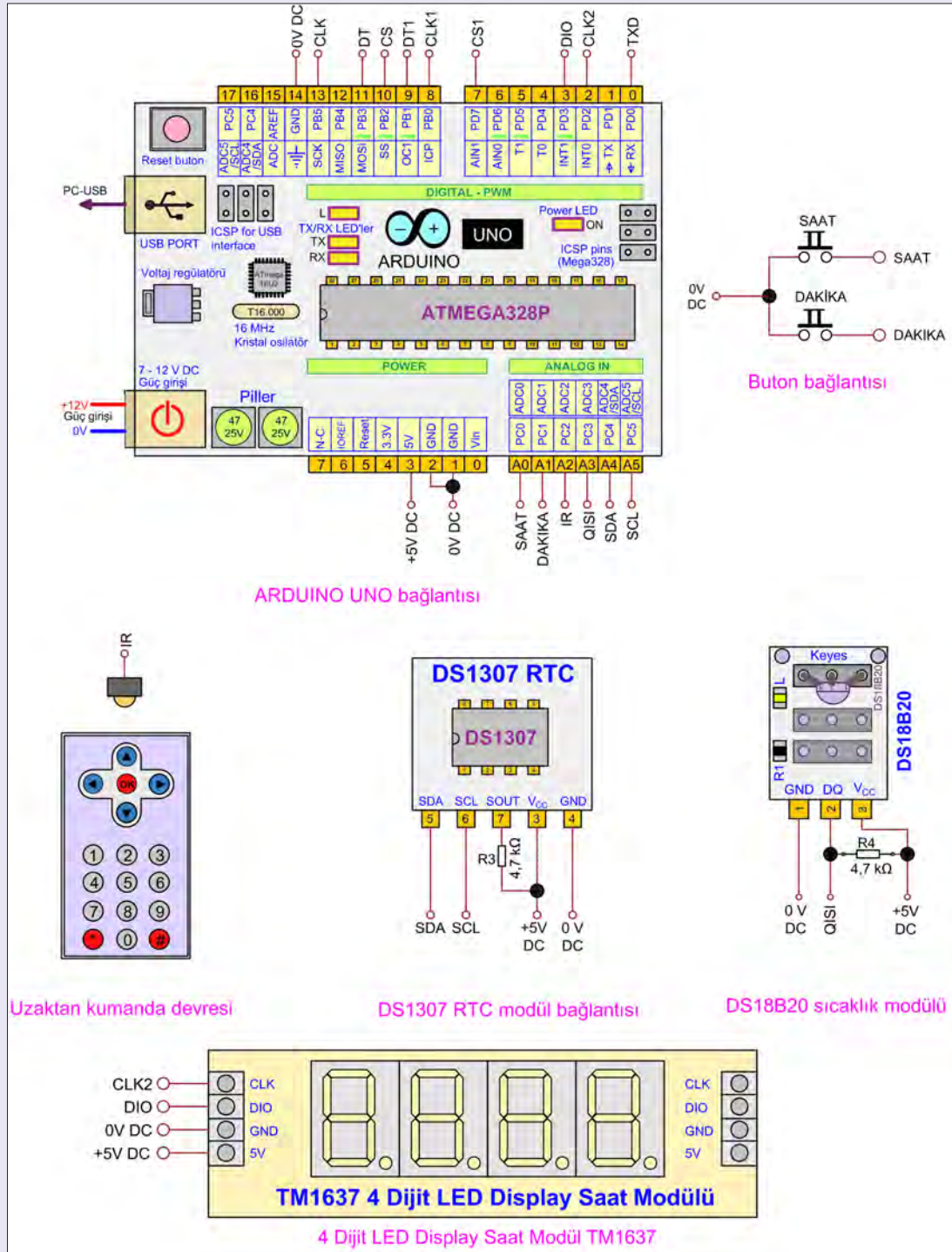
1. Robot kolun aşağı yukarı hareket etmesinde step motor kullanılmasının sebebi nedir?
2. Robot kolun parça tutup bırakmasında servo motor kullanılmasının sebebi nedir?

Çalışmanın Değerlendirilmesi

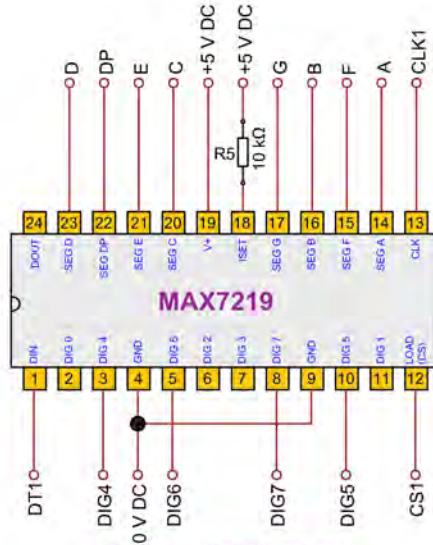
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroislemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

Proje Adı	Termometreli Kayan Yazılı Saat Devre Projesi	Proje No. : 8
Amaç	Verilen devreyi mikrodeneleyiciye göre yaparak çalıştırmak.	Süre: 7 ders saati

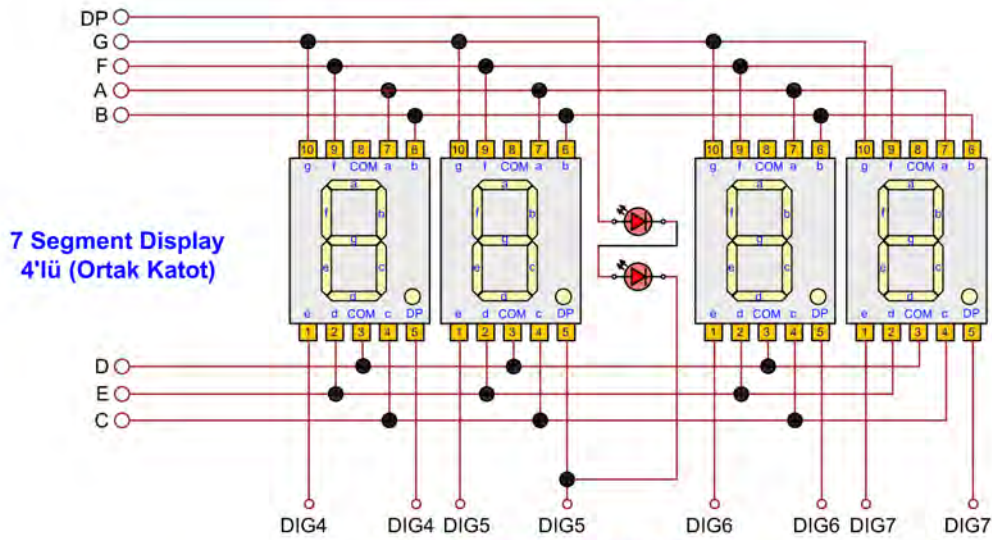
İstenenler: ARDUINO UNO kullanılarak termometreli kayan yazılı saat devresi yapılacak ve isteğe göre bu devre, kutu içine yerleştirilecektir. ARDUINO UNO kullanılarak tasarlanan termometreli kayan yazılı saat devresi Görsel A.19 ve Görsel A.20’de görülmektedir.



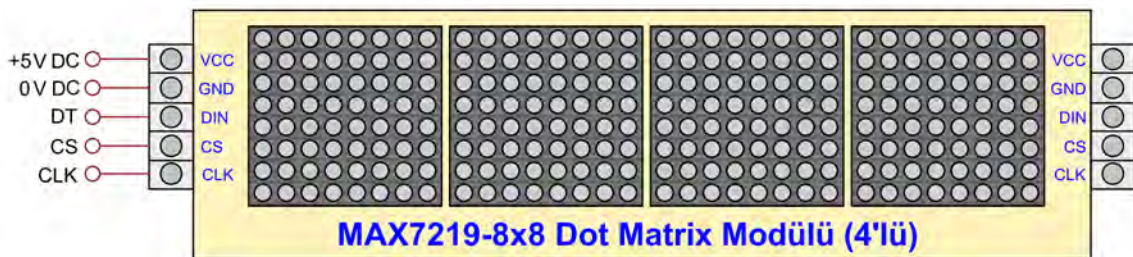
Görsel A.19: ARDUINO UNO ile termometreli kayan yazılı saat devresi (1)



Display sürücü devresi



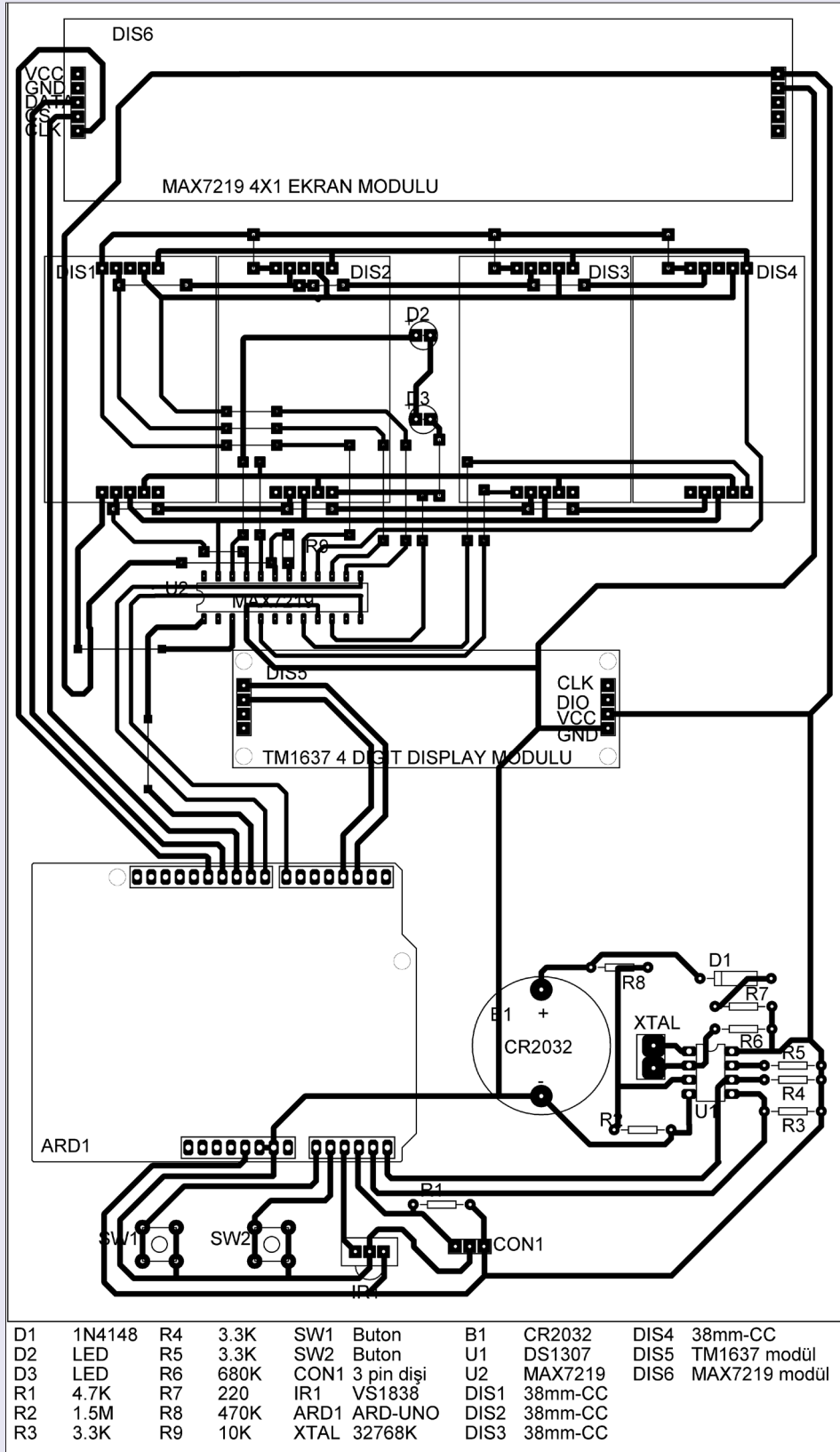
7 Segment display devresi



Kayan yazı devresi

Görsel A.20: ARDUINO UNO ile termometreli kayan yazılı saat devresi (2)

Baskı Devre Şeması: Görsel A.21’de devrenin baskı devre şeması görülmektedir.



Görsel A.21: ARDUINO UNO ile termometreli kayan yazılı saat baskı devresi

Malzemeler: Tablo A.15'te devrenin malzeme listesi görülmektedir.

Tablo A.15: Devrede Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
MAX7219 ekran modülü	4 parçalı 8x8 dot matris ARDUINO ekran modülü	1 adet
Tm1637 ekran modülü	4 digit ARDUINO ekran modülü	1 adet
ARDUINO IR modülü	Uzaktan kumanda ve IR alıcısı (VS1838B)	1 adet
Direnç 0,25W	220 Ω	1 adet
Direnç 0,25W	3,3 k Ω	3 adet
Direnç 0,25W	4,7 k Ω	1 adet
Direnç 0,25W	10 k Ω	1 adet
Direnç 0,25W	470 k Ω	1 adet
Direnç 0,25W	680 k Ω	1 adet
Direnç 0,25W	1,5 M Ω	1 adet
Display	38 mm ortak katot	4 adet
LED diyot	Kırmızı	2 adet
1N4148 diyot	Kristal diyot	1 adet
MAX7219 Entegre	LED sürücü	1 adet
DS1307	Gerçek zamanlı saat	1 adet
DS18B20	Çubuk tip sıcaklık sensörü	1 adet
Kristal	32768 kHz	1 adet
Pil yuvası	CR2032	1 adet
CR2032 pil	Buton pil	1 adet
Buton	6x6 mm 4 pin touch buton	2 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-

Kod Yazılımları: Tablo A.16'da devrenin kod yazılımları görülmektedir.

Tablo A.16: ARDUINO UNO'lu Termometreli Kayan Yazılı Saat Devresinin Kod Yazılımları

No.	Kodlar
1	//kütüphaneler ve tanımlamalar
2	#include <SPI.h>
3	#include <Adafruit_GFX.h>
4	#include <Max72xxPanel.h>
5	#include "LedControl.h"
6	#include <Wire.h>
7	#include <OneWire.h>
8	#include <DallasTemperature.h>
9	#include <TM1637Display.h>
10	#include <Time.h>
11	#include <DS1307RTC.h>
12	#include <IRremote.h>
13	//değişkenler
14	int animasyon=0, temp, temp1, temp3=2,temp2=8, sn=0, saat2;



```
15 uint8_t data[]={ 0xff, 0xff, 0xff, 0xff, 0xff, 0xff };
16 const int changeH = A0; //saat ayar butonu
17 const int changeM = A1; //dakika ayar butonu
18 int pinCS = 10;
19 tmElements_t sr;
20 int numberOfHorizontalDisplays = 1;
21 int numberOfVerticalDisplays = 4;
22 LedControl yt = LedControl(9, 8, 7, 1);
23 char coden[]={ 0x7E, 0x30, 0x6D, 0x79, 0x33, 0x5B, 0x5F, 0x70, 0x7F, 0x73}; ///KATOT
24 const int RECV_PIN = A2;
25 I2CDevice irrecv(RECV_PIN);
26 Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays, numberOfVerticalDisplays);
27 String mesaj = ""; //TIRNAK İŞARETİ İÇERİSİNE YAZILAN YAZI EKRANDA KAYAR.
28 char dizi[100];
29 int bekle = 40; //YAZI KAYMA HIZI MIN 1 MAKS 100
30 int bosluk = 1;
31 int kar_uzunluk = 5 + bosluk;
32 int tsay=0;
33 #define ONE_WIRE_BUS A3
34 #define CLK 2
35 #define DIO 3
36 OneWire oneWire(ONE_WIRE_BUS);
37 DallasTemperature sensors(&oneWire);
38 decode_results results;
39 //alt programlar
40 void setupTimer1()
41 {
42   noInterrupts();
43   TCCR1A = 0;
44   TCCR1B = 0;
45   TCNT1 = 0;
46   // 0.5 Hz (16000000/((15624+1)*512))
47   OCR1A = 7812;
48   TCCR1B |= (1 << WGM12);
49   TCCR1B |= (1 << CS12) | (1 << CS10);
50   TIMSK1 |= (1 << OCIE1A);
51   interrupts();
52 }
53 ISR(TIMER1_COMPA_vect)
54 {
55   noInterrupts();
56   sn++;
57   if (sn%2==0){
58     bitSet (saat2,7);
59     yt.setRow(0,5,saat2);}
60   else{
61     bitClear (saat2,7);
```

```

62  yt.addRow(0,5,saat2);}
63  interrupts();
64  }
65  void DS1307_display()
66  {
67    RTC.read(sr);
68    yt.addRow(0,4,coden[sr.Hour/10]);
69    yt.addRow(0,5,coden[sr.Hour%10]);
70    yt.addRow(0,6,coden[sr.Minute/10]);
71    yt.addRow(0,7,coden[sr.Minute%10]);
72    saat2=coden[sr.Hour%10];
73  }
74  void ISI()
75  {
76    tsay++;
77    sensors.requestTemperatures();
78    temp = sensors.getTempCByIndex(0);
79    temp1= round(temp);
80    TM1637Display display(CLK, DIO);
81    display.clear();
82    display.setBrightness(0x0f);
83    if ((tsay==1) && (temp1=85)){
84      data[0] = 0x6E;
85      data[1] = 0x78;
86      data[2] = 0x77;
87      data[3] = 0x6D;
88      display.setSegments(data);
89      display.setBrightness(0x0f);}
90    else{
91      data[0] = display.encodeDigit(temp1/10);
92      data[1] = display.encodeDigit(temp1%10);
93      data[2] = 0x63;
94      data[3] = 0x39;
95      display.setSegments(data);
96      display.setBrightness(0x0f);}
97    }
98    //genel ayarlar
99    void setup()
100   {
101     Serial.begin(9600);
102     irrecv.enableIRIn();
103     pinMode(changeH, INPUT_PULLUP);
104     pinMode(changeM, INPUT_PULLUP);
105     yt.shutdown(0, false);
106     yt.setIntensity(0, 5);
107     yt.clearDisplay(0);
108     matrix.setIntensity(13); // LED PANEL PARLAKLIK AYARI (MAKS 15 MİN 1)

```



```
109 matrix.setRotation(3); // YAZININ HANGİ YÖNDEN KAYACAĞI (DISPLAY YERLEŞİMİNE GÖRE 1 İLE 4 ARASI )
110 sensors.requestTemperatures();
111 sensors.requestTemperatures();
112 temp = sensors.getTempCByIndex(0);
113 delay(250);
114 setupTimer1();
115 }
116 //ana program
117 void loop()
118 {
119   if (Serial.available() > 0) {
120     mesaj = Serial.readString();
121     ISI();
122     DS1307_display();
123     for ( int i = 0 ; i < kar_uzunluk * mesaj.length() + matrix.width() - 1 - bosluk; i++ ) {
124       matrix.fillScreen(LOW);
125       int karakter = i / kar_uzunluk;
126       int sat = (matrix.width() - 1) - i % kar_uzunluk;
127       int sut = (matrix.height() - 8) / 2;
128       while (sat + kar_uzunluk - bosluk >= 0 && karakter >= 0) {
129         if ( karakter < mesaj.length() ) {
130           if (irrecv.decode(&results)){
131             switch(results.value){
132               case 0xFFA25D:
133                 RTC.read(sr);
134                 sr.Hour++;
135                 if (sr.Hour>=24){
136                   sr.Hour=0;}
137                 RTC.write(sr);
138                 DS1307_display();
139                 break;
140               case 0xFFE21D:
141                 RTC.read(sr);
142                 sr.Minute++;
143                 if (sr.Minute>=60){
144                   sr.Minute=0;}
145                 RTC.write(sr);
146                 DS1307_display();
147                 break;}
148             irrecv.resume();
149             matrix.drawChar(sat, sut, mesaj[karakter], HIGH, LOW, 1);}
150           karakter--;
151           sat -= kar_uzunluk;}
152         if(!digitalRead(changeM)){
153           RTC.read(sr);
154           sr.Minute++;
155           if (sr.Minute>=60){
```

```

157 sr.Minute=0;}
158 RTC.write(sr);
159 DS1307_display();
160 while(!digitalRead(changeM));}
161 if(!digitalRead(changeH)){
162   RTC.read(sr);
163   sr.Hour++;
164   if (sr.Hour>=24){
165     sr.Hour=0;}
166   RTC.write(sr);
167   DS1307_display();
168   while(!digitalRead(changeH));
169   delay(bekle);}
170   int i=0;
171 }

```

Devrenin Çalışması

Simülasyon başlatılır. Virtual terminalden (bilgisayar seri port) kayan yazı panelinde gösterilecek yazı girilir. Yazı DIS6 ekranında görüntülenir. DS1307 hafızasındaki saat bilgisi DIS1, DIS2, DIS3 ve DIS4 displaylerde görüntülenir. Saniye durumu D2 ve D3 LED diyotlarda görüntülenir. Saat ayarı yapılmak istenirse saat, SW1 butonundan veya uzaktan kumandanın “1” tuşundan saat bilgisi, SW2 butonundan veya uzaktan kumandanın “3” tuşundan dakika bilgisi girilerek değiştirilir. Çubuk tipi (metal gövdeli) DS18B20 sıcaklık sensöründen alınan sıcaklık bilgisi DIS5’te görüntülenir.

Sorular

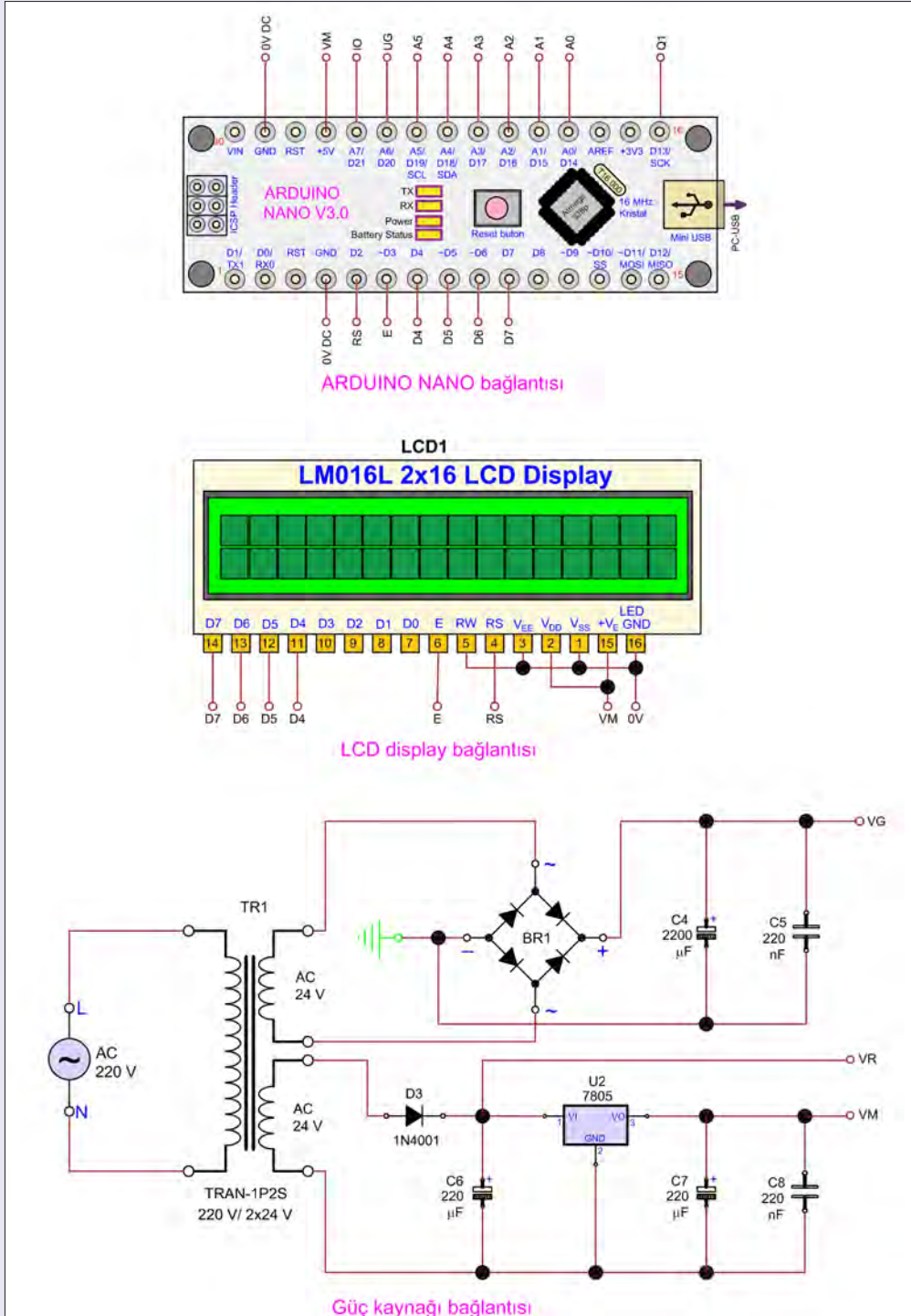
1. CR2032 pilin görevi nedir?
2. Çubuk tipi DS18B20 kullanılmasıdaki amaç nedir?

Çalışmanın Değerlendirilmesi

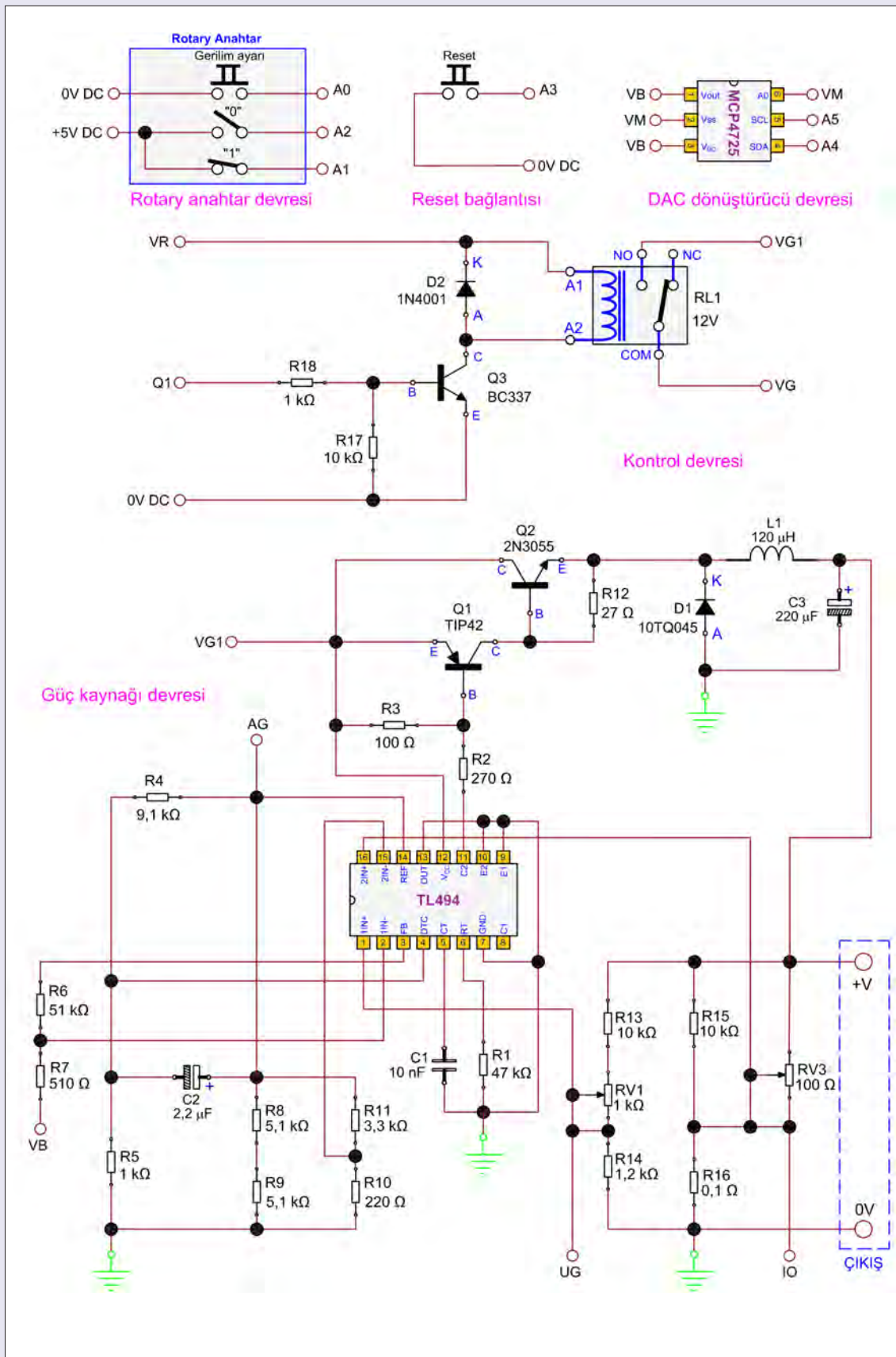
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroislemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

Proje Adı	Ayarlı DC Güç Kaynağı Projesi	Proje No. : 9
Amaç	Verilen devreyi mikrodnetleyiciye göre yaparak çalıştırmak.	Süre: 7 ders saati

İstenenler: ARDUINO NANO kullanılarak ayarlı DC güç kaynağı devresi yapılacak ve isteğe göre bu devre kutu içine yerleştirilecektir. ARDUINO NANO kullanılarak tasarlanan ayarlı DC güç kaynağı devresi Görsel A.22’de görülmektedir.

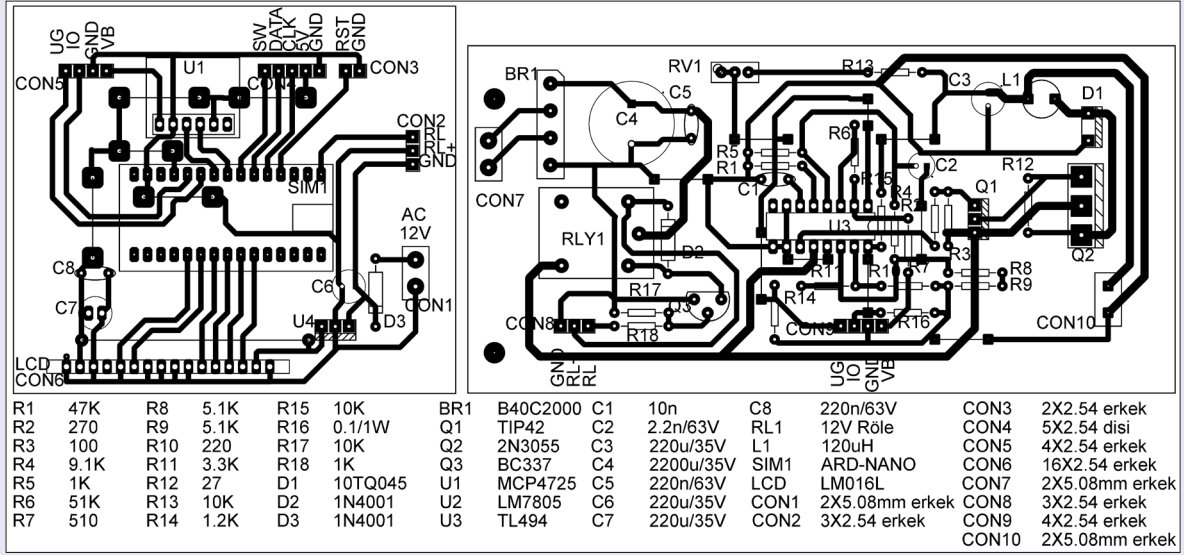


Görsel A.22: Ayarlı DC güç kaynağı projesi devresi (a)



Görsel A.22: Ayarlı DC güç kaynağı projesi devresi (b)

Baskı Devre Şeması: Görsel A.23'te devrenin baskı devre şeması görülmektedir.



Görsel A.23: ARDUINO NANO ile ayarlı DC güç kaynağı baskı devresi

Malzemeler: Tablo A.17'de devrenin malzeme listesi görülmektedir.

Tablo A.17: Devrede Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO NANO	Mikrodenetleyicili kart	1 adet
MCP4725 DAC modülü	Dijital analog çevirici	1 adet
LM016L	16x2 LCD ekran	1 adet
Direnç 0,25W	27 Ω	1 adet
Direnç 0,25W	100 Ω	1 adet
Direnç 0,25W	220 Ω	1 adet
Direnç 0,25W	270 Ω	1 adet
Direnç 0,25W	510 Ω	1 adet
Direnç 0,25W	1 k Ω	2 adet
Direnç 0,25W	1,2 k Ω	1 adet
Direnç 0,25W	3,3 k Ω	1 adet
Direnç 0,25W	5,1 k Ω	2 adet
Direnç 0,25W	9.1 k Ω	1 adet
Direnç 0,25W	10 k Ω	3 adet
Direnç 0,25W	47 k Ω	1 adet
Direnç 0,25W	51 k Ω	1 adet
Direnç 1W	0,1 Ω / 1W	1 adet
Kondansatör	2,2 nF / 63V	1 adet
Kondansatör	10 nF / 63V	1 adet
Kondansatör	220 nF / 63V	2 adet
Kondansatör	220 μ F / 35V	3 adet
Kondansatör	2200 μ F / 35V	1 adet
10TQ045 diyot	Şotki	1 adet
1N4001 diyot	Kristal diyot	2 adet

B40C2000 diyot	Köprü diyot	1 adet
Transistör	BC337	1 adet
Transistör	TIP42	1 adet
Transistör	2N3055	1 adet
LM7805	5 V voltaj regülatörü	1 adet
TL494	Voltaj referans entegresi	1 adet
Röle	12 V tek kontak	1 adet
Bobin	120 μ H	1 adet
2.54mm konektör	2 pin erkek dişi takım	1 adet
2.54mm konektör	3 pin erkek dişi takım	2 adet
2.54mm konektör	4 pin erkek dişi takım	2 adet
2.54mm konektör	5 pin erkek dişi takım	1 adet
5.08mm Konektör	2 pin erkek dişi takım	3 adet
Pin header	40 pin erkek	1 adet
Pin header	40 pin dişi	1 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-

Kod Yazılımları: Tablo A.18’de devrenin kod yazılımları görülmektedir.

Tablo A.18: ARDUINO NANO’lu Ayarlı DC Güç Kaynağı Devresi Kod Yazılımları

No.	Kodlar
1	<code>//kütüphaneler ve tanımlamalar</code>
2	<code>#include<Wire.h></code>
3	<code>#include<ARDUINO.h></code>
4	<code>#include <LiquidCrystal.h></code>
5	<code>#define MCP4725 0x61</code>
6	<code>#define gerilim A6</code>
7	<code>#define akim A7</code>
8	<code>LiquidCrystal lcd(2,3,4,5,6,7);</code>
9	<code>//değişkenler</code>
10	<code>const int ayar_buton = A0;</code>
11	<code>const int CLKBUT = A1;</code>
12	<code>const int DTBUT = A2;</code>
13	<code>const int RESET = A3;</code>
14	<code>int start = 13;</code>
15	<code>int vdeger, adeger;</code>
16	<code>int counter = 0, onlar=0, onlar_t=0;</code>
17	<code>int simdikiCLK;</code>
18	<code>int oncekiCLK;</code>
19	<code>string currentDir = "";</code>
20	<code>unsigned long lastButtonPress = 0;</code>
21	<code>float vout = 0.0;</code>
22	<code>float aout = 0.0;</code>
23	<code>byte buffer[3];</code>
24	<code>//genel ayarlar</code>
25	<code>void setup()</code>
26	<code>{pinMode(ayar_buton, INPUT_PULLUP);</code>
27	<code>pinMode(CLKBUT, INPUT);</code>
28	<code>pinMode(DTBUT, INPUT);</code>
29	<code>pinMode(RESET, INPUT_PULLUP);</code>
30	<code>pinMode(gerilim, INPUT);</code>
31	<code>pinMode(akim, INPUT);</code>
32	<code>pinMode(start, OUTPUT);</code>
33	<code>digitalWrite(start,LOW);</code>
34	<code>Wire.begin();</code>
35	<code>counter= 400;</code>
36	<code>buffer[0] = 0b01000000;</code>
37	<code>buffer[1] = counter >> 4;</code>
38	<code>buffer[2] = counter << 4;</code>
39	<code>Wire.beginTransmission(MCP4725);</code>
40	<code>Wire.write(buffer[0]);</code>
41	<code>Wire.write(buffer[1]);</code>
42	<code>Wire.write(buffer[2]);</code>
43	<code>Wire.endTransmission();</code>
44	<code>lcd.begin(16,2);</code>
45	<code>lcd.print("0-24 V AYARLI GUC");</code>
46	<code>lcd.setCursor(0,1);</code>
47	<code>lcd.print(" KAYNAGI ");</code>
48	<code>lcd.clear();</code>



```

49 void setup()
50 {pinMode(ayar_buton, INPUT_PULLUP);
51 pinMode(CLKBUT, INPUT);
52 pinMode(DTBUT, INPUT);
53 pinMode(RESET, INPUT_PULLUP);
54 pinMode(gerilim, INPUT);
55 pinMode(akim, INPUT);
56 pinMode(start, OUTPUT);
57 digitalWrite(start, LOW);
58 Wire.begin();
59 counter= 400;
60 buffer[0] = 0b01000000;
61 buffer[1] = counter >> 4;
62 buffer[2] = counter << 4;
63 Wire.beginTransmission(MCP4725);
64 Wire.write(buffer[0]);
65 Wire.write(buffer[1]);
66 Wire.write(buffer[2]);
67 Wire.endTransmission();
68 lcd.begin(16,2);
69 lcd.print("0-24 V AYARLI GUC");
70 lcd.setCursor(0,1);
71 lcd.print(" KAYNAGI ");
72 lcd.clear();
73 oncekiCLK = digitalRead(CLKBUT);
74 digitalWrite(start, HIGH);}
75 //ana program
76 void loop()
77 {vdeger = analogRead(gerilim);
78 vout = (vdeger*50)/1023.00;
79 adeger = analogRead(akim)*10;
80 aout = adeger/200.00;
81 if (aout>=3){
82 digitalWrite(start, LOW);
83 lcd.clear();
84 lcd.print("ASIRIM AKIM");
85 lcd.setCursor(0,1);
86 lcd.print("RST BUT. BASINIZ");
87 while(digitalRead(RESET));
88 digitalWrite(start, HIGH);}
89 lcd.clear();
90 lcd.print("GERILIM:");
91 lcd.print(vout);
92 lcd.print("V");

93 lcd.setCursor(0,1);
94 lcd.print(" AKIM:");
95 lcd.print(aout);
96 lcd.print("A");
97 if(!digitalRead(ayar_buton)){
98 lcd.clear();
99 lcd.print("GERILIM AYARI");
100 lcd.setCursor(0,1);
101 lcd.print("BUTONU BIRAKINIZ");
102 while(!digitalRead(ayar_buton));
103 vdeger = analogRead(gerilim);
104 vout = (vdeger*50)/1023;
105 lcd.clear();
106 lcd.print("GERILIM:");
107 lcd.print(vout);
108 while(digitalRead(ayar_buton)){
109 simdikiCLK = digitalRead(CLKBUT);
110 if (simdikiCLK != oncekiCLK && simdikiCLK == 1){
111 if (digitalRead(DTBUT) != simdikiCLK) {
112 vout=vout+0.1;
113 if (vout>=24){
114 vout=24;
115 } else {
116 vout=vout-0.1;
117 if (vout<=0){
118 vout=0;}}
119 lcd.clear();
120 lcd.print("GERILIM AYARI:");
121 lcd.setCursor(0,1);
122 lcd.print(vout);
123 lcd.print("V");}
124 oncekiCLK = simdikiCLK;}
125 while(!digitalRead(ayar_buton));
126 counter= (vout*1000)/12.2;
127 buffer[0] = 0b01000000;
128 buffer[1] = counter >> 4;
129 buffer[2] = counter << 4;
130 Wire.beginTransmission(MCP4725);
131 Wire.write(buffer[0]);
132 Wire.write(buffer[1]);
133 Wire.write(buffer[2]);
134 Wire.endTransmission();
135 delay(500);}
136 delay(500);}

```

Devrenin Çalışması

Simülasyon başlatılır. LCD ekran açılıp ekranda bilgilendirme mesajının gelmesi beklenir. İlk çalıştırmada güç kaynağı çıkışı 5V gerilimde sabittir. Gerilim değerini değiştirmek için gerilim ayar butonuna (rotary anahtar) basılı tutulur. LCD ekranda bilgilendirme mesajı görülünce buton bırakılır. Gerilim değeri, rotary anahtarın sağa çevrilmesiyle artırılır, sola çevrilmesiyle azaltılır (Simülatörde rotary anahtar B girişi lojik 0 iken A girişi lojik 1, lojik 0 yapılırsa gerilim artar. Rotary anahtar B girişi lojik 1 iken A girişi lojik 1, lojik 0 yapılırsa gerilim azalır.). Gerilim ayarı yapıldıktan sonra gerilim ayar butonuna basılıp bırakılır. Ayarlanan gerilim değeri güç kaynağı çıkışında görülür. 3A akım üstü veya kısa devre durumunda güç kaynağı kendini korumaya alır. Güç kaynağını koruma durumundan çıkarmak için kısa devre veya aşırı akım durumu giderilir. Reset butonuna basılıp bırakılır.

Sorular

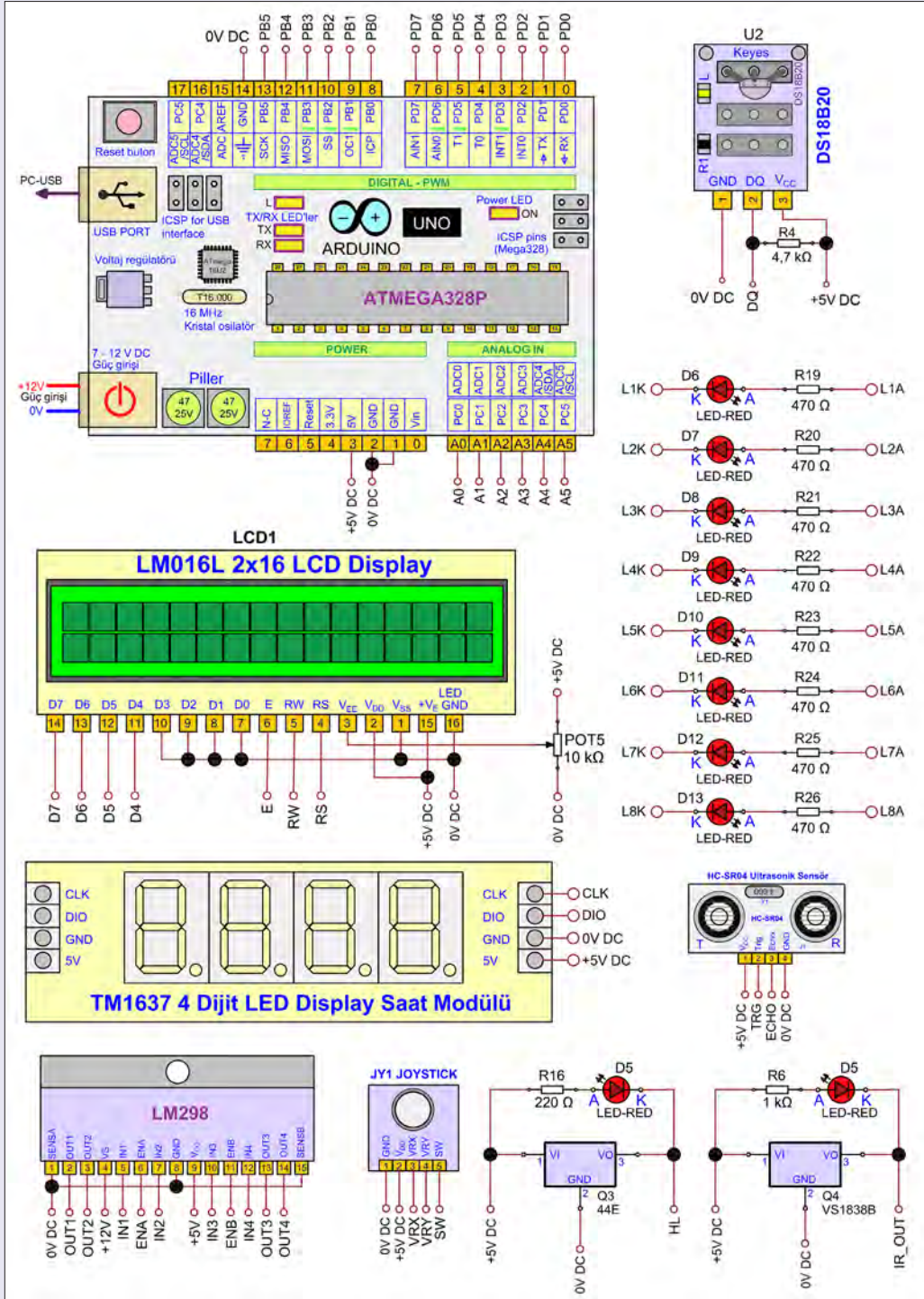
1. Devrede röle kullanılmasının amacı nedir?
2. TL494 entegresinin devredeki görevi nedir?

Çalışmanın Değerlendirilmesi

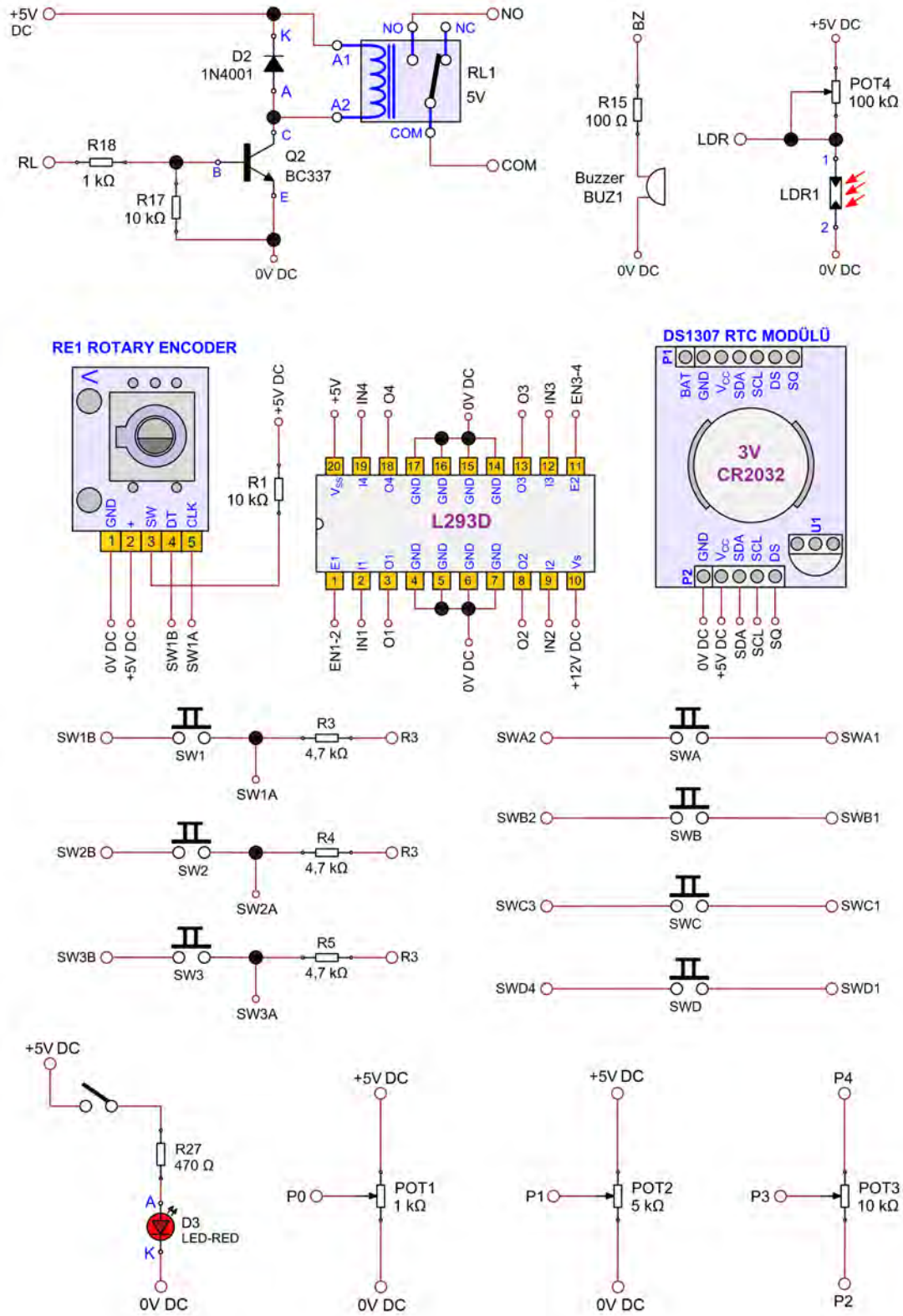
No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	

Proje Adı	ARDUINO Deney Seti Projesi	Proje No. : 10
Amaç	Verilen devreyi mikrodeneleyiciye göre yaparak çalıştırmak.	Süre: 7 ders saati

İstenenler: ARDUINO UNO kullanılarak deney seti devresi yapılacak ve isteğe göre bu devre, kutu içine yerleştirilecektir. ARDUINO UNO kullanılarak tasarlanan deney seti devresi Görsel A.24'te görülmektedir.

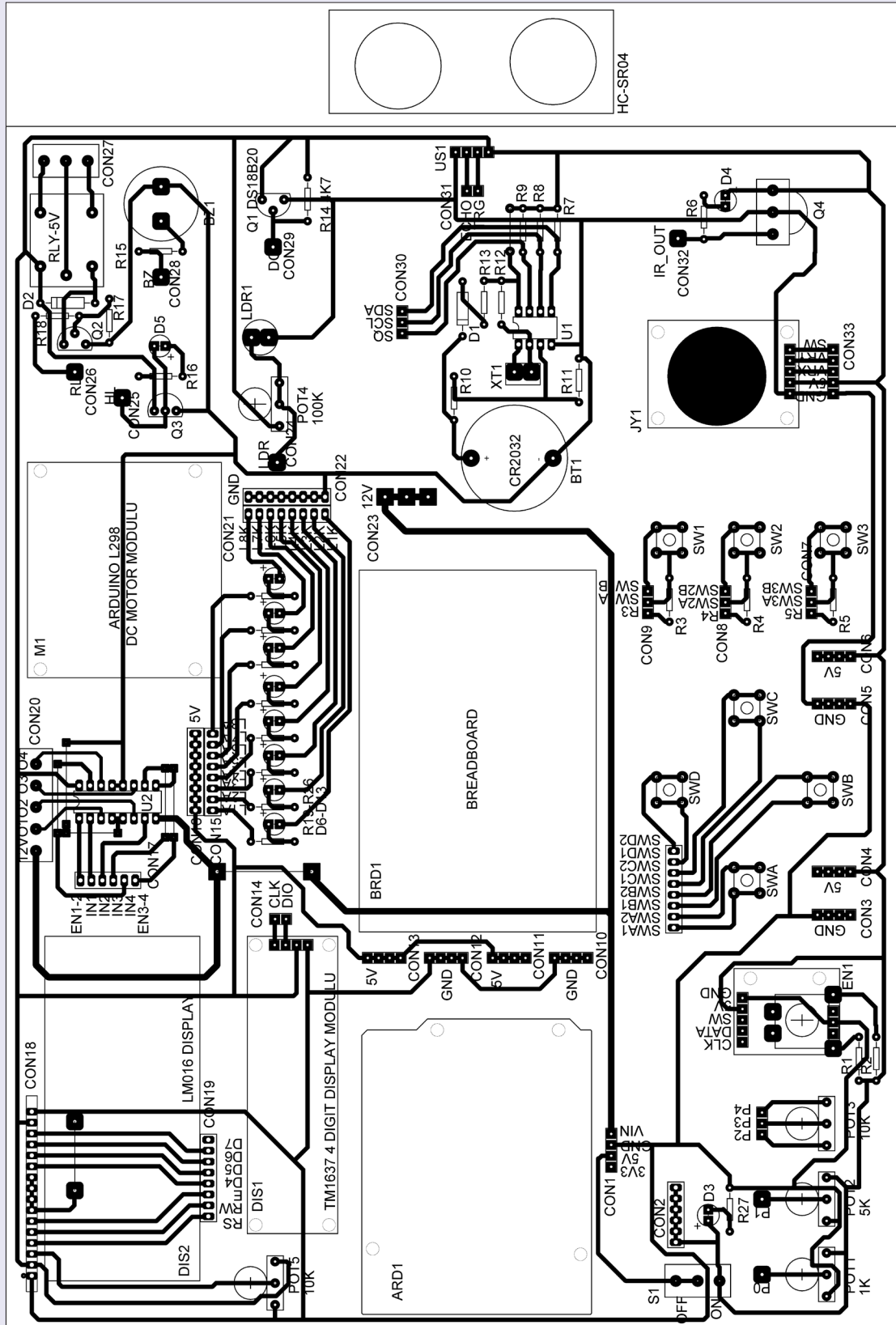


Görsel A.24: ARDUINO deney seti projesi devresi (a)



Görsel A.24: ARDUINO deney seti projesi devresi (b)

Baskı Devre Şeması: Görsel A.25'te devrenin baskı devre şeması görülmektedir.



Görsel A.25: ARDUINO deney seti projesi baskı devresi

Görsel A.25'teki ARDUINO UNO kullanılarak tasarlanan deney seti baskı devre malzemeleri Görsel A.26'da görülmektedir.

R1	10K	R25	470	BZ1	5V-12mm	CON10	4 pin erkek
R2	10K	R26	470	RL1	5V	CON11	4 pin erkek
R3	4.7K	R27	470	JY1	Arduino joystick	CON12	6 pin erkek
R4	4.7K	D1	1N4148	EN1	Arduino encoder	CON13	4 pin erkek
R5	4.7K	D2	1N4007	MD1	Arduino L298	CON14	2 pin erkek
R6	1K	D3	LED	DIS1	TM1637 disp. modülü	CON15	8 pin erkek
R7	3.3K	D4	LED	DIS2	LM0106	CON16	8 pin erkek
R8	3.3K	D5	LED	US1	HC-SR04	CON17	6 pin erkek
R9	3.3K	D6	LED	SW1	6x6mm tact	CON18	16 pin dişi
R10	470K	Q1	DS18B20	SW2	6x6mm tact	CON19	8 pin erkek
R11	1.5M	Q2	BC337	SW3	6x6mm tact	CON20	4 pin PCB klemens
R12	680K	Q3	44E	SWA	6x6mm tact	CON21	8 pin erkek
R13	220	Q4	VS1838B	SWB	6x6mm tact	CON22	8 pin erkek
R14	4.7K	U1	DS1307	SWC	6x6mm tact	CON23	3 pin erkek
R15	100	U2	L293	SWD	6x6mm tact	CON24	1 pin erkek
R16	220	POT1	1K	CON1	4 pin erkek	CON25	1 pin erkek
R17	10K	POT2	5K	CON2	6 pin erkek	CON26	1 pin erkek
R18	1K	POT3	10K	CON3	4 pin erkek	CON27	3 pin PCB klemens
R19	470	POT4	100K	CON4	4 pin erkek	CON28	1 pin erkek
R20	470	POT5	10K	CON5	4 pin erkek	CON29	1 pin erkek
R21	470	S1	IC205	CON6	4 pin erkek	CON30	3 pin erkek
R22	470	BT1	CR2032	CON7	3 pin erkek	CON31	2 pin erkek
R23	470	XT1	32768K	CON8	3 pin erkek	CON32	1 pin erkek
R24	470	LDR1	10mm	CON9	3 pin erkek	CON33	5 pin erkek

Görsel A.26: ARDUINO deney seti PCB malzeme özellikleri

Malzemeler: Tablo A.19'da devrenin malzeme listesi görülmektedir.

Tablo A.19: Devrede Kullanılan Malzemeler

Malzemenin Adı	Malzemenin Özelliği	Miktarı
ARDUINO UNO	Mikrodenetleyicili kart	1 adet
HC-SR04 modülü	Mesafe sensör modülü	1 adet
LM016L	16x2 LCD ekran	1 adet
TM1637 Ekran modülü	4 dijit ARDUINO ekran modülü	1 adet
L298 ARDUINO modülü	Motor sürücü modülü	1 adet
Joystick modülü	ARDUINO X-Y eksenli joystick	1 adet
Rotary anahtar modülü	ARDUINO rotary anahtar	1 adet
ARDUINO IR modülü	Uzaktan kumanda ve IR alıcısı (VS1838B)	1 adet
L293 entegre	Step motor sürücü	1 adet
DS1307	Gerçek zamanlı saat	1 adet
Direnç 0,25W	100 Ω	1 adet
Direnç 0,25W	220 Ω	2 adet
Direnç 0,25W	470 Ω	9 adet
Direnç 0,25W	1 k Ω	2 adet
Direnç 0,25W	3,3 k Ω	3 adet
Direnç 0,25W	4,7 k Ω	4 adet
Direnç 0,25W	10 k Ω	3 adet
Direnç 0,25W	470 k Ω	1 adet
Direnç 0,25W	680 k Ω	1 adet
Direnç 0,25W	1,5 M Ω	1 adet
Potansiyometre	1 k Ω	1 adet
Potansiyometre	5 k Ω	1 adet
Potansiyometre	10 k Ω	2 adet



Potansiyometre	100 kΩ	1 adet
Kristal	32,768 kHz	1 adet
Pil yuvası	CR2032	1 adet
CR2032 pil	Buton pil	1 adet
VS1838B	38 kHz	1 adet
LDR	10 mm	1 adet
Buzzer	5V-12 mm	1 adet
LED	Kırmızı	11 adet
1N4007 diyot	Kristal diyot	1 adet
1N4148 diyot	Kristal diyot	1 adet
Transistör	BC337	1 adet
DS18B20	Entegre tip sıcaklık sensörü	1 adet
TL494	Voltaj referans entegresi	1 adet
44E	Hall (manyetik) sensör	1 adet
Röle	5 V tek kontak	1 adet
Bobin	120 μH	1 adet
Klemens	3 pin PCB	1 adet
Klemens	5 pin PCB	1 adet
Buton	6x6 mm touch	7 adet
Anahtar	ON/OFF 3 pin IC-205	1 adet
Pin header	40 pin erkek	2 adet
Pin header	40 pin dişi	5 adet
Zil teli	0,5 veya 0,75 mm ²	3-5 m
DC güç kaynağı	5...24 V DC	1 adet
El aletleri	Yankeski, tornavida, pense, kargaburnu vb.	-

Çalışmanın Değerlendirilmesi

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1.	Devre elemanlarının kontrol edilmesi	10	
2.	Devre yazılım kodlarının oluşturulması	10	
3.	Devre yazılım kodlarının kontrol edilmesi	10	
4.	Devre yazılım kodlarının mikroişlemciye yüklenmesi	10	
5.	Devre elemanlarının breadboarda yerleştirilmesi	10	
6.	Elemanlar arasında devre kablo bağlantılarının yapılması	10	
7.	Devre bağlantılarının kontrol edilmesi ve devreye enerji verilmesi	10	
8.	Devrenin çalıştırılması ve kontrol edilmesi	10	
9.	Devre yazılım kodlarının kaydedilmesi ve teslim edilmesi	10	
10.	Devrenin teslim edilmesi ve soruların cevaplanması	10	
TOPLAM PUAN		100	



EK-B

UYGULAMA PROJELERİ

1. Proje: Gelişmiş Dijital Saat
2. Proje: Dijital Pano
3. Proje: Uzaktan Kumandalı Robot
4. Proje: Bilimsel Hesap Makinesi
5. Proje: PV'li Aydınlatma
6. Proje: Direnç Ölçümü
7. Proje: Alarmlı Şifreli Kilit
8. Proje: Trafik Kavşak Kontrol
9. Proje: Işık Ve Ses Efekt
10. Proje: Bilgisayar Kontrollü Kayan Yazı
11. Proje: Dron



UYGULAMA YÖNERGESİ



Bu bölümde devre şeması ve kod bilgileri olmayan projeler yer almaktadır. Bu projeler aşağıda verilen yönerge doğrultusunda uygulanacaktır.

- Projeler grup hâlinde ya da bireysel olarak yapılabilir.
- Verilen projeler içinden seçim yapılacaktır.
- Proje seçiminin ardından gruplar oluşturulacaktır.
- Gruplar, proje türüne bağlı olarak 2 ila 12 kişiden oluşturulabilir.
- Her grubun bir başkanı olacaktır.
- Projenin uygulama aşaması Tablo B.1'deki gibi yürütülecektir.

Tablo B.1: Proje Uygulama Tablosu

Hafta	Yapılacaklar
1.-3. Hafta	Grupların oluşturulması ve yapılabilecek projelerin araştırılması
4.-6. Hafta	Yapılacak projenin seçilmesi ve grup içerisinde iş bölümünün yapılması
7.-13. Hafta	Proje ile ilgili araştırmaların yapılması ve verilerin toplanması
14.-16. Hafta	Proje ile ilgili devre şemalarının oluşturulması ve malzemelerin sipariş edilmesi
17.-20. Hafta	Proje ile ilgili devre simülasyonlarının yapılması ve CAD'de devrelerin çizilmesi
21.-24. Hafta	Malzemelerin teslim alınması, kontrol edilmesi ve baskı devrelerin çizilmesi
25.-27. Hafta	Baskı devre kartlarının hazırlanması, delinmesi ve kontrollerinin yapılması
28.-30. Hafta	Baskı devre kartlarına malzemelerin ve elemanların lehimlenmesi
31.-32. Hafta	Yazılım kodlarının hazırlanması, derlenmesi ve mikrodeneleyiciye yüklenmesi
33.-34. Hafta	Yapılan devrelerin kutuya/zemine monte edilmesi ve bağlantılarının yapılması
35. Hafta	Yapılan projenin çalıştırılması ve testlerinin yapılması
36. Hafta	Dosya raporu ile uygulamaların ve yazılımların teslim edilmesi

- Yapılacak devreler CAD veya devre çizim yazılımlarında çizilecek ve uygun programda devrelerin simülasyonları yapılacaktır. Baskı devre yazılımında ise baskı devre tasarımı yapılacaktır.
- Baskı devrede ütüleme veya mini CNC kazıma yöntemlerinden uygun olanı kullanılacaktır.
- Entegreler için entegre soketleri kullanılacaktır.
- Yapılan çalışmalar ile ilgili tüm veriler dosyaya işlenecektir. Dosya en az 20 sayfa olacaktır.
- Simülasyon, baskı devre ve çizim dosyaları uygun CD ortamına aktararak teslim edilecektir.
- Projede tek tip mikrodeneleyici kullanılabileceği gibi birden çok aynı veya farklı tip mikrodeneleyici de kullanılabilir.
- Dosyada mikrodeneleyicilerin ve kullanılan elemanların teknik özellikleri açıklanacaktır.
- Proje, verilen açıklamalar doğrultusunda yürütülecektir.
- Proje tanımında verilmeyen kısımlar gruplar tarafından belirlenecektir.
- Yapılan çalışmalar haftalık olarak öğretmene bildirilecek ve raporlanacaktır.
- 36 hafta sonunda projeyi teslim etmeyen grup başarısız sayılacaktır.

- Tüm çalışmalar **İş Sağlığı ve Güvenliği (İSG)** tedbirleri alınarak gerçekleştirilecektir.
- Projeler uygun maket formatında, kutuda veya malzeme üzerinde yapılacaktır.
- Proje değerlendirmeleri Tablo B.2’de verilen değerlendirme ölçütlerine göre yapılacaktır.

Tablo B.2: Proje Değerlendirme Ölçütleri

No.	Yapılacaklar	Puanı
1	Grupların oluşturulması ve araştırmaların yapılarak konunun belirlenmesi	05
2	Devrelerin uygun yazılımda çizilmesi ve simülasyonlarının yapılması	20
3	Malzemelerin temin edilmesi ve kontrolü	05
4	Baskı devrelerin tasarlanması, asit banyosuna atılması (mini CNC ile kazınması)	05
5	Baskı devre kartlarının uygun şekilde delinmesi ve elemanların lehimlenmesi	10
6	Lehimlenen devrelerin kutu veya zemin üzerinde montajı ve kablo bağlantıları	10
7	Yazılım kodlarının oluşturulması, derlenmesi ve mikrodenetleyiciye yüklenmesi	10
8	Devrenin çalıştırılması ve testlerinin yapılması	05
9	Proje dosyasının hazırlanması ve ek materyaller ile zamanında teslim edilmesi	20
10	Çalışmaların haftalık olarak kontrollerinin yapılması	10

PROJELER

Aşağıda bu projelerin isimleri görülmektedir. Bu projelerden birisini seçiniz.

Proje No.	Proje Adı
01	Gelişmiş Dijital Saat
02	Dijital Pano
03	Uzaktan Kumandalı Robot
04	Bilimsel Hesap Makinesi
05	PV’li Aydınlatma
06	Direnç Ölçümü
07	Alarmlı Şifreli Kilit
08	Trafik Kavşak Kontrol
09	Işık ve Ses Etki
10	Bilgisayar Kontrollü Kayan Yazı
11	Dron

DİKKAT!

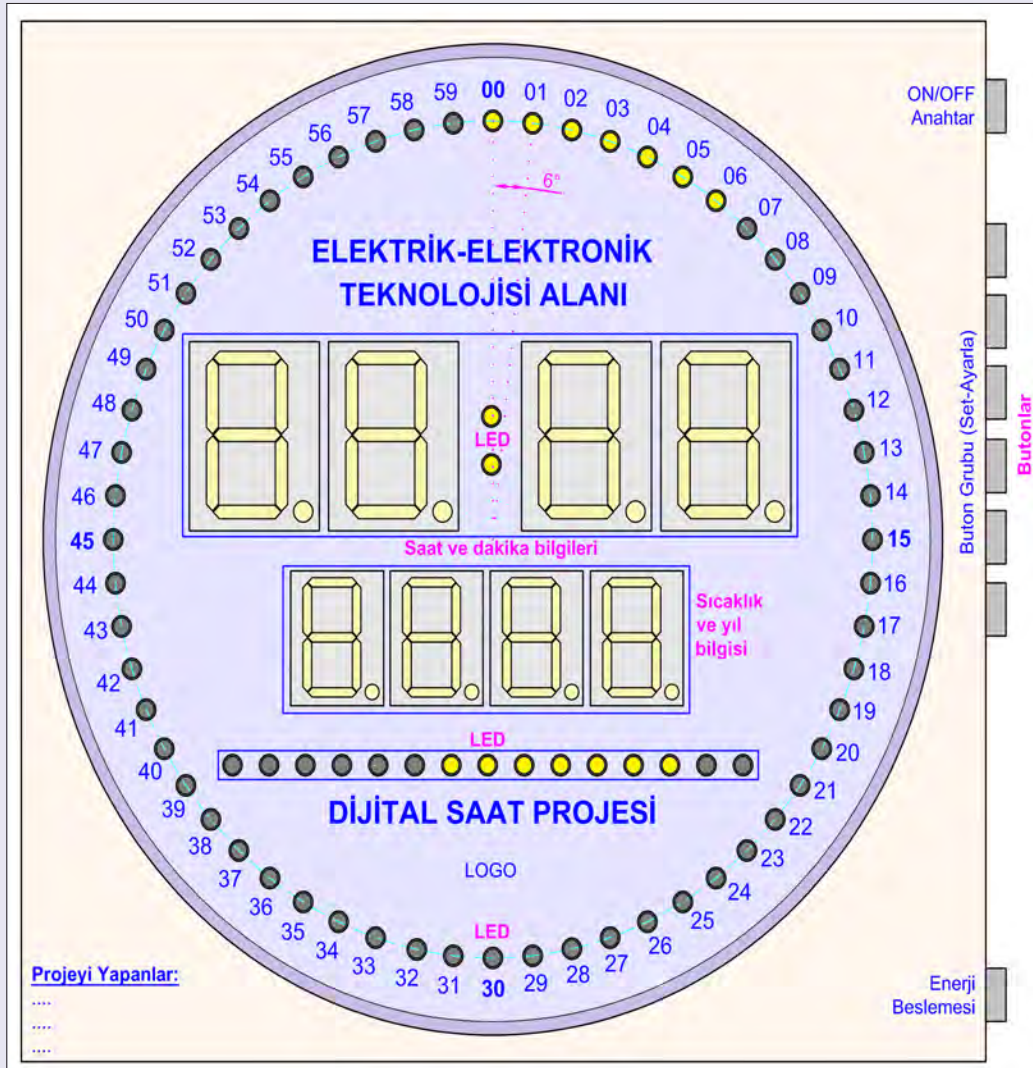
- Uygulama ile ilgili tüm dosyalara yanda verilen karekoddan ulaşabilirsiniz.
- Ekteki kısımdan devrelerde kullanılan elemanların ayak bağlantılarına, baskı devre şemalarına ve detaylı teknik bilgilerine ulaşabilirsiniz.



1. Proje: Gelişmiş Dijital Saat



Görsel B.1’de CAD ortamında tasarlanan gelişmiş dijital saat projesi görülmektedir.



Görsel B.1: Gelişmiş dijital saat projesi

İSTENENLER

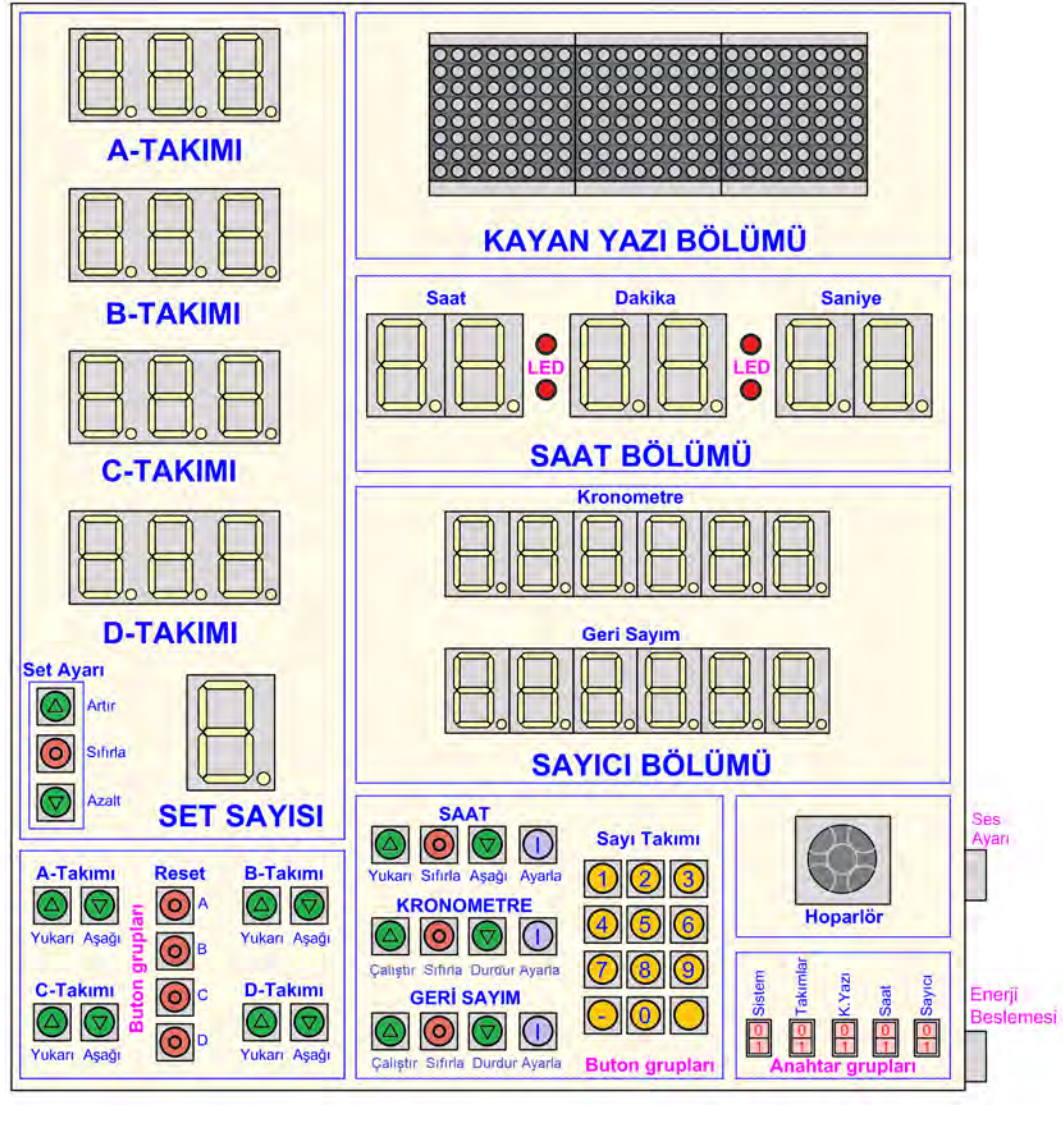
- Dijital saat, daire bir plaka üzerinde oluşturulacaktır (Görsel B.1).
- Devre, uygun bir pano veya kapağın içine yerleştirilecek ve üzeri saydam cam, plastik vb. malzeme ile kapatılacaktır.
- Devrede dört adet büyük ve dört adet küçük tip 7 segment display kullanılacaktır. Büyük display üzerinde saat ve dakika gösterimi yapılacaktır. Küçük displayde ise yıl ve sıcaklık gösterimleri sıra ile yapılacaktır. Bu displayler plakanın tam ortasına yerleştirilecektir.
- Daire levhanın kenarlarına saniyeyi göstermek için orantılı şekilde 60 adet LED yerleştirilecektir. Bu LED’lerin yerleşimi dairenin açısına uygun olacak şekilde gerçekleştirilecektir.
- Saniyeyi gösteren 60 LED, üstten başlayacak şekilde saat yönünde sıralı olarak yanacak ve son LED de yandıktan sonra sıfırlanarak baştan yanmaya başlayacaktır.

- Saati, dakikayı, saniyeyi, sıcaklığı, yılı ve animasyonu ayarlamak ve kontrolünü sağlamak için gerekli sayıda buton ve anahtar kullanılacaktır. Bunların yerleşimi kutunun sağ kenarına ortalanacak şekilde alt alta yerleştirilecektir.
- Saati açıp kapamak için açma/kapama (0-1) anahtar kullanılacaktır.
- Cihazı beslemek için uygun DC veya AC girişleri saatin kenarına monte edilecektir. Burada klemens kullanılabileceği gibi soketler de tercih edilebilecektir.
- Büyük displayin ortasına iki adet LED bağlanacak ve LED'lerin yanıp sönme işlemi birer saniye gösterimi olacaktır.
- Displaylerin altına yerleştirilen LED'ler üzerinde çeşitli animasyonlar yapılabilecektir.
- Cam veya plastik üzerine dijital saat için gerekli etiketler yapıştırılacak veya uygun kalem ile yazılar yazılacaktır.
- Enerji kesildiğinde saat bilgilerinin kaybolmaması için haricî pil kullanılacaktır.
- Duvara asmak için kutu arkasına uygun delik veya askı aparatı yerleştirilecektir.
- Kutunun sol alt kısmına projeyi yapan grup üyelerinin adları alt alta gelecek şekilde yazılacaktır. Burada grup başkanı en üst sırada yer alacaktır.
- Saatin orta üstüne “Elektrik-Elektronik Teknolojisi Alanı” ve orta altına ise “Dijital Saat Projesi” yazı etiketleri eklenecektir. Ayrıca okul veya alan logosu en alt orta kısma eklenebilir.
- 60 LED'in kenarlarına saniyeleri gösteren uygun etiketler yerleştirilecektir.
- Saatin dış kutu alanına butonların özelliklerini gösteren etiketler uygun şekilde (ön panel kısmına ya da kutunun yan kısmına) yerleştirilecektir.
- Kullanılan etiket yazı tipi ve rengi, kutu tasarımı ile uygun olarak olmalıdır.
- Kullanılacak LED'lerin renkleri, ön camdan görünecek şekilde seçilmelidir. LED'ler normal LED, Power LED ya da RGB LED olabilir.
- Ortadaki 15 adet LED grubu animasyon efekt işlemi vermek için yerleştirilecektir. Burada çeşitli animasyon efektleri yapılabilir ve yandaki ilgili buton grubundan bu işlem ayarlanabilir.
- Kutunun ilgili bölümlerinden havalandırma boşlukları açılacaktır.
- Her saat başında kısa aralıklı ses efekti çalacaktır.
- Belirlenen buton/anahtar grubunun haricinde buton ve/veya anahtarlar kullanılabilir. Her anahtar ve buton grubu için etiketler tanımlanacaktır.
- Çalışmaya haricî uygulamalar/fonksiyonlar da ilave edilebilir. Bunun tasarımı ve planlaması grup kontrolündedir.



2. Proje: Dijital Pano

Görsel B.2'de CAD ortamında tasarlanan dijital pano projesi görülmektedir.



Görsel B.2: Dijital pano projesi

İSTENENLER

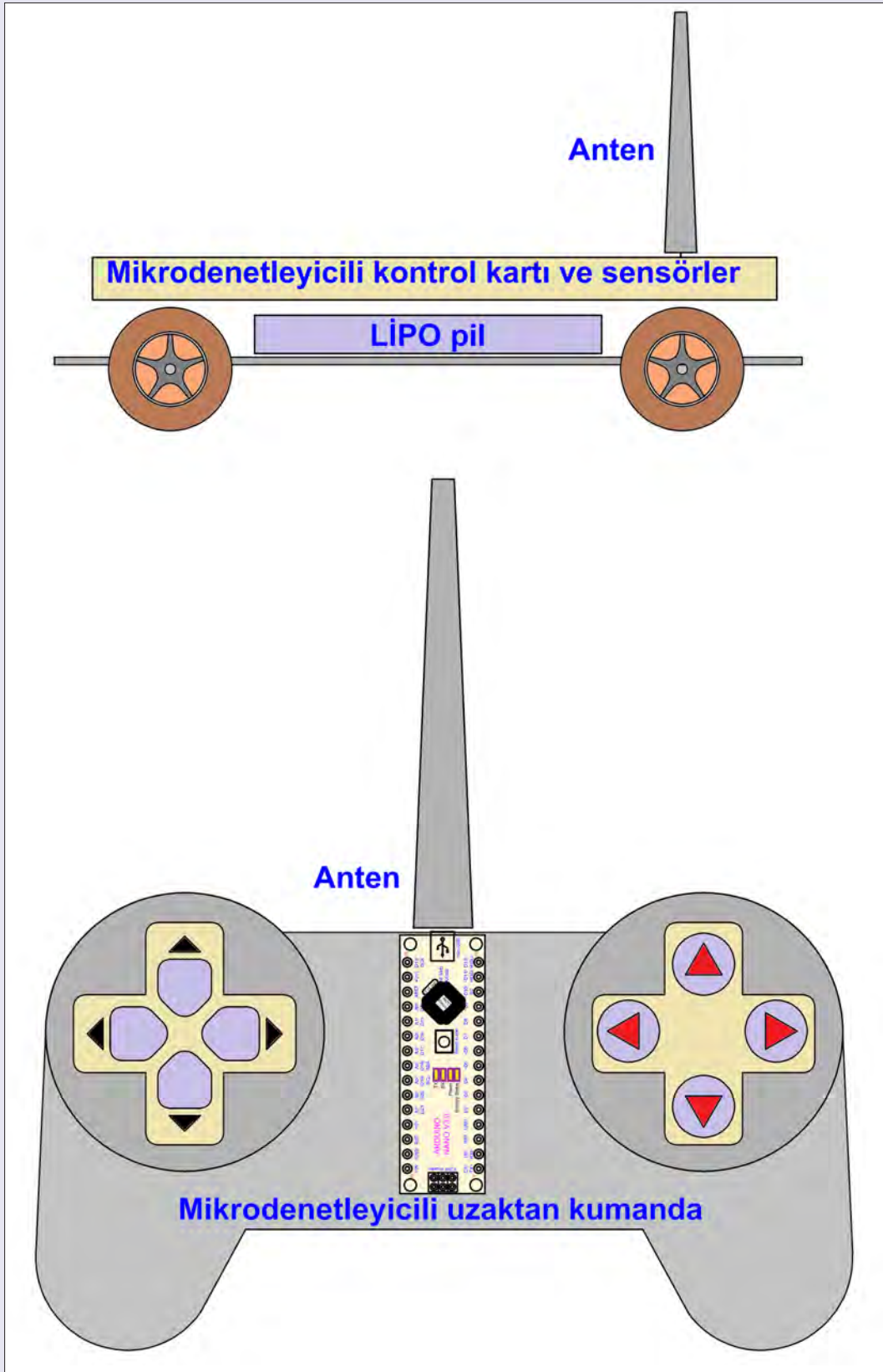
- Dijital pano uygun bir levhaya sahip kutu üzerinde oluşturulacaktır (Görsel B.2).
- Sol bölümde her takımın yarışma verilerinin gözüktüğü ileri-geri sayma bölümleri oluşturulacaktır. Sayma işlemi 0-999 arasında ileri-geri olarak gerçekleştirilecektir. Bu bölümde 3'lü 7 segment display kullanılacaktır. Ayrıca bölümün kontrol butonları alt kısımda verilecektir.
- Set sayısı, takımların kaçınıcı turda olduklarını gösteren bir veridir. Bu bölümde 7 segment display kullanılacak ve 0-9 arası ileri-geri olarak sayacaktır. Hemen yanındaki buton grubundan bu sayma işlemleri ayarlanabilecektir.

- Sağ üstte 3 adet dot matriks displayli kayan yazı bölümü oluşturulacaktır. Bu bölümde belirlenen yazı grubu sürekli olarak sağdan sola hareket edecektir. Bu bölüme yazı girişi bilgisayardan ya da cep telefonundan yapılabilecektir.
- Sağ ortaya 6 adet 7 segment displayli ve dört adet LED'li saat devresi yerleştirilecektir. Bu bölümde belirlenen saat, dakika ve saniye bilgisini de verecek şekilde çalışacak ve LED'ler belirli aralıklar ile yanıp sönecektir. Saat ile ilgili tüm ayarlar ve veri girişleri aşağıdaki buton gruplarından ve sayı tuş takımından yapılabilecektir.
- Sayıcı bölümünde iki farklı devre vardır. Bunlardan biri kronometre, diğeri ise geri sayım devresidir. Kronometre, belirlenen sayıdan itibaren "Çalıştır" butonuna basılınca ileri doğru otomatik olarak saymaya başlar ve "Durdur" butona basılıncaya kadar bu sayma işlemi devam eder. "Ayarla" butonu ile kronometrenin ilk sayı değeri ayarlanır ve tuş takımından sayı değeri girilir.
- Geri sayım devresi, belirlenen sayıdan itibaren "Çalıştır" butonuna basılınca geriye doğru otomatik olarak saymaya başlar ve "Durdur" butona basılıncaya kadar bu geri sayım işlemi devam eder. "Ayarla" butonu ile geri sayımın ilk sayı değeri ayarlanır ve tuş takımından sayı değeri girilir.
- Sağ altta her bir devreyi açıp kapatan anahtar grupları vardır. "Sistem" anahtarı ile tüm devre açılıp kapatılır. "Takımlar" anahtarı ile A-B-C-D takımlarının olduğu bölüm, "K.Yazı" anahtarı ile kayan yazı bölümü, "Saat" anahtarı ile saat bölümü açılıp kapatılır. "Sayıcı" anahtarı ile de kronometre ve geri sayım bölümü açılıp kapatılır.
- Sağ alt köşede hoparlör bölümü bulunacaktır. Bu hoparlörlerden özellikle çalışmaların durdurulduğu anlarda ve saat başlarında kısa süreli ses veya müzik çalacaktır. Hemen yanındaki potansiyometre ile bu sesin ses ayarı yapılabilecektir.
- Takım göstergeleri negatif değer alabilirken diğer tüm göstergeler pozitif değer alacaktır. Tuş takımındaki "-" butonu sadece bu kısımda aktif olur. Ayrıca yanda yer alan boşluk tuşu saat bölümü haricinde göstergelerde boşluk atamak için kullanılacaktır.
- Buton grupları, istenirse uzatma kablolar ile ikinci bir buton paneli oluşturulabilmektedir.
- Levha üzerine uygun tip etiketler yapıştırılacaktır.
- Cihazı beslemek için uygun DC veya AC girişleri saatin kenarına monte edilecektir. Burada klemens kullanılabileceği gibi soketler de tercih edilebilir.
- Kutunun ilgili bölümlerinden havalandırma boşlukları açılacaktır.
- Sayı takımında hazır tip keypad kullanılmamalıdır.
- Birden fazla mikrodenetleyici kullanılabilir.
- Hoparlör yerine buzzer kullanılabilir.
- Belirlenen buton/anahtar grubunun haricinde buton ve/veya anahtarlar kullanılabilir. Her anahtar ve buton grubu için etiketler tanımlanacaktır.
- Çalışmaya haricî uygulamalar/fonksiyonlar da ilave edilebilir. Bunun tasarımı ve planlaması grup kontrolündedir.
- Kutunun duvara monte edilebilmesi veya asılabilmesi için uygun delikler veya askı aparatları arka kısmında kullanılacaktır.
- Tüm elemanlar levhaya direkt monte edilebilir. Display ve paneller levhaya yerleştirilip üzerlerine levha/cam yerleştirilerek monte edilebilir.



3. Proje: Uzaktan Kumandalı Robot

Görsel B.3'te bilgisayar ortamında tasarlanan uzaktan kumandalı robot projesi görülmektedir.



Görsel B.3: Uzaktan kumandalı robot projesi

İSTENENLER

- Çok fonksiyonlu ve uzaktan kumandalı robot yapılacaktır (Görsel B.3).
- Robotun kontrolü uzaktan kumanda ile ya da üzerindeki kontrol tuşları ile de yapılabilir.
- Robot devresinin çeşitli fonksiyonları Tablo B.1’de verilmiştir. Yapılacak robot üzerinde bu fonksiyonlardan en az üçü, en fazla altısı bulunacaktır.

Tablo B.1: Uzaktan Kumandalı Robot Projesinin Fonksiyonları

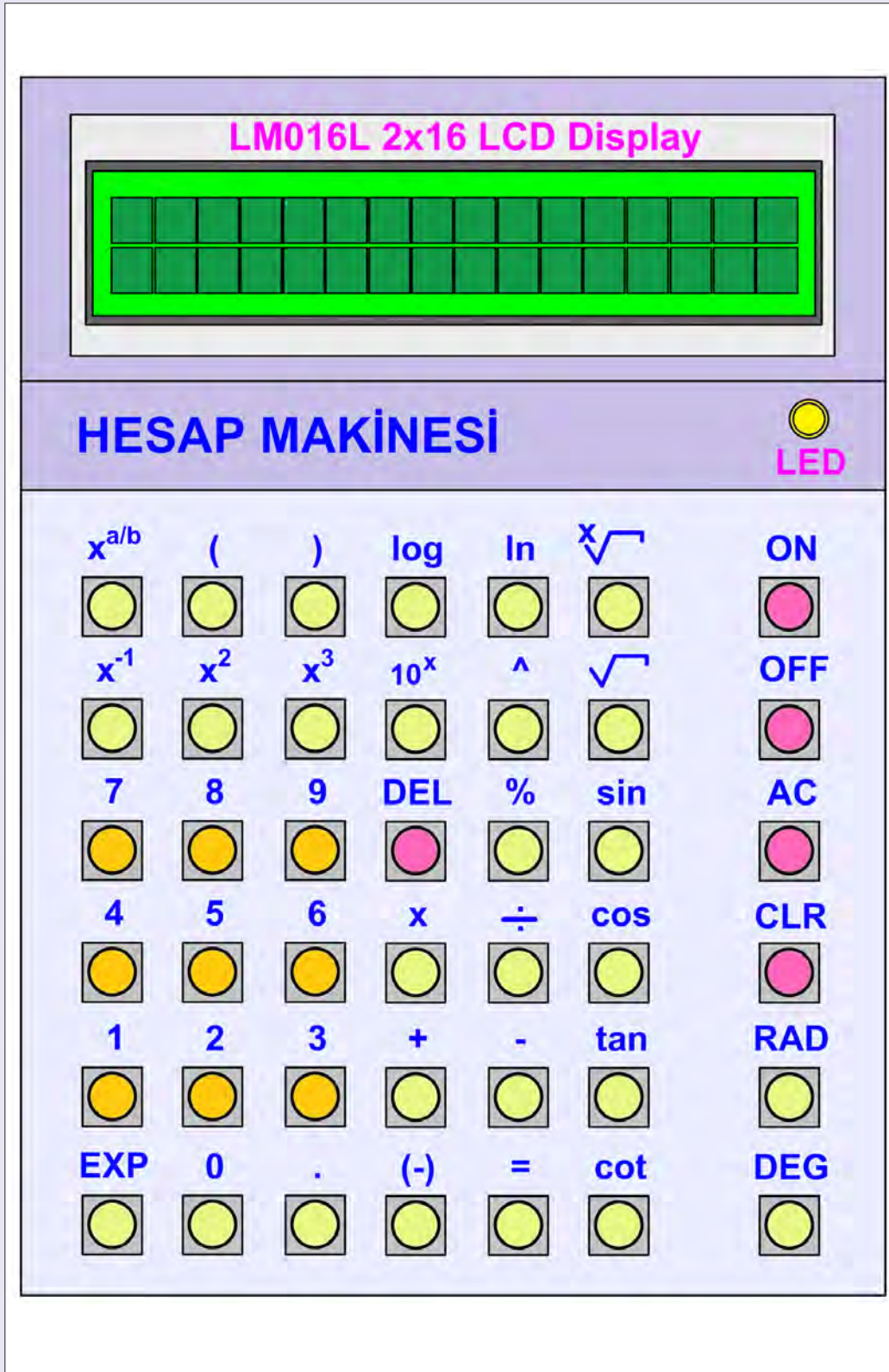
Tipi	Fonksiyon Açıklaması	Tipi	Fonksiyon Açıklaması
A	Uzaktan kontrollü robot uygulaması	D	Yön kontrol eden robot uygulaması
B	Çizgi izleyen robot uygulaması	E	Engel algılayan robot uygulaması
C	Ses kontrollü kontrol edilen robot	F	Işığı takip eden robot uygulaması

- A fonksiyonda robot klasik olarak kontrol edilecektir. Uzaktan kumanda ile robot istenilen yere hareket edecek ve istenilen etkide kol kontrollerini sağlayacaktır.
- B fonksiyonda robot, belirlenen çizgiyi izleyen bir uygulama sağlayacaktır. Çizgi rengi siyah, zemin rengi beyaz ya da çizgi rengi beyaz, zemin rengi siyah olabilir.
- C fonksiyonda robot, seslerin geldiği yerlere belirlenen komut ile hareket edecektir. Sesler belirli komutlara göre belirli düzeyde olacaktır.
- D fonksiyonda robot, belirlenen kullanıcıyı kamera yardımıyla takip edecektir. Takip, robot takipten kopuncaya ya da sonlandır komutu gelinceye kadar devam edecektir.
- E fonksiyonda robot, hareket ederken karşılaştığı engelleri aşacaktır.
- F fonksiyonda robot, belirlenen ışığı algılayarak takip edecektir. Ses kontrollü robot ile benzerlik taşımaktadır.
- Robot üzerinde çeşitli sensörler uygun açılarda yerleştirilecektir.
- Robotun tasarımında üç boyutlu tasarım araçlarından yararlanılacak ve 3D yazıcılar kullanılarak tasarlanan ürünlerin çıktıları alınacaktır.
- Robot üzerinde besleme kaynağı (LIPO piller vb.) olacaktır.
- Uygun tipli mikrodenetleyiciler ve kartlar kullanılacaktır.
- Uzaktan kumanda alıcısı ve vericisinde RF, bluetooth, IR veya WI-FI alıcı verici modüllerinden istenilen kullanılacaktır.
- Joystick olarak Arduino joystick modülü kullanılabilir.



4. Proje: Bilimsel Hesap Makinesi

Görsel B.4'te CAD ortamında tasarlanan bilimsel hesap makinesi projesi görülmektedir. Bu projede yer alan butonların görevleri ise Tablo B.1'de verilmiştir.



Görsel B.4: Bilimsel hesap makinesi projesi

Tablo B.1: Bilimsel hesap makinesi tuşlarının özellikleri

Tuş	Özellikler	Tuş	Özellikler
0-9	Sayı tuşları (0, 1, ... 9)	ON	Çalıştırma butonu
EXP	ex işlemini yapan buton	OFF	Kapatma butonu
.	Virgül atayan buton	AC	Gösterge ışığını ayarlama butonu
(-)	Sayıyı negatif yapan buton	CLR	Resetleme butonu
=	İfadenin sonucunu alan buton	DEL	Bir önceki ifadeyi silme butonu
+	Toplama işlemi yapan buton	RAD	Radyal ifadeye ayarlama butonu
-	Çıkarma işlemi yapan buton	DEG	Derece ifadeye ayarlama butonu
x	Çarpma işlemi yapan buton	sin	Sinüs işlemi yapan buton
/	Bölme işlemi yapan buton	cos	Kosinüs işlemi yapan buton
%	Yüzdelik alan buton	tan	Tanjant işlemi yapan buton
^	a^b üst işlemi yapan buton	cot	Kotanjant işlemi yapan buton
x^{-1}	İşlemin tersini alan ($1/x$) buton	(	Parantez açan buton
x^2	İşlemin karesini alan buton	)	Parantez kapayan buton
x^3	İşlemin küpünü alan buton	log	Logaritma alan buton
$x^{a/b}$	İşlem üzerinde bölme yapan buton	ln	$\ln(\log_e)$ işlemi yapan buton
10^x	$10x$ üslü işlemi yapan buton	$x^{1/x}$	$b^{1/x}$ işlemi yapan buton

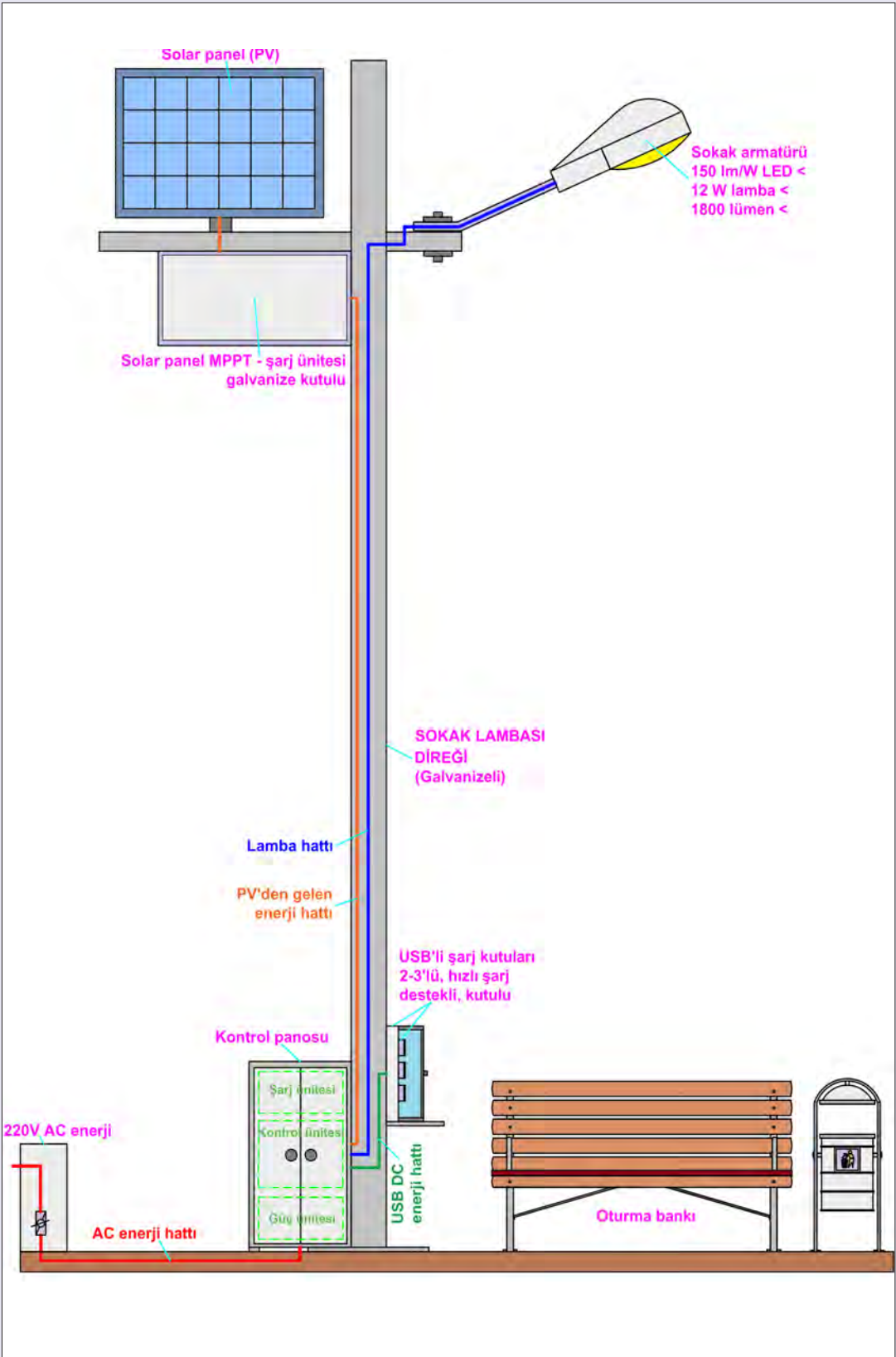
İSTENENLER

- Bilimsel özellikte bir hesap makinesi yapılacaktır (Görsel B.4).
- Uygun bir malzeme üzerine monte edilecektir.
- Uygun genişlikte butonlar kullanılacaktır. Hazır tip tuş takımları kullanılmayacaktır.
- Gösterge için 2x16 ve üzerinde bir LCD panel kullanılacaktır. Burada grafik özellikli LCD paneller de tercih edilebilir.
- Birden fazla mikrodenetleyici kullanılacaktır.
- Tablo B.1'deki tuş görevlerine uygun program hazırlanacaktır.
- Her buton grubunun önüne uygun etiketler yerleştirilecektir.
- Hesap makinesi devresinin arkasına haricî DC beslemesi için 2'li veya 3'lü kalem pil yuvası koyulacaktır. Ayrıca devreyi beslemesi için güç giriş bölümü de kullanılabilir.
- Cihaz üzerine bir adet LED yerleştirilecektir. Hesap makinesinin her işleminde bu led yanıp sönecektir.
- Cihaza haricî özellikler/fonksiyonlar da eklenebilecektir.
- Cihaz içinin havalandırması için gerekli delikler açılacaktır.
- Uygun vidalarla kutu kapatılacaktır.
- Kutu arkasına cihazın eğik durabilmesi için bir ayak yerleştirilebilir.



5. Proje: PV'li Aydınlatma

Görsel B.5'te CAD ortamında tasarlanan PV'li aydınlatma projesi görülmektedir.



Görsel B.5: PV'li aydınlatma projesi

İSTENENLER

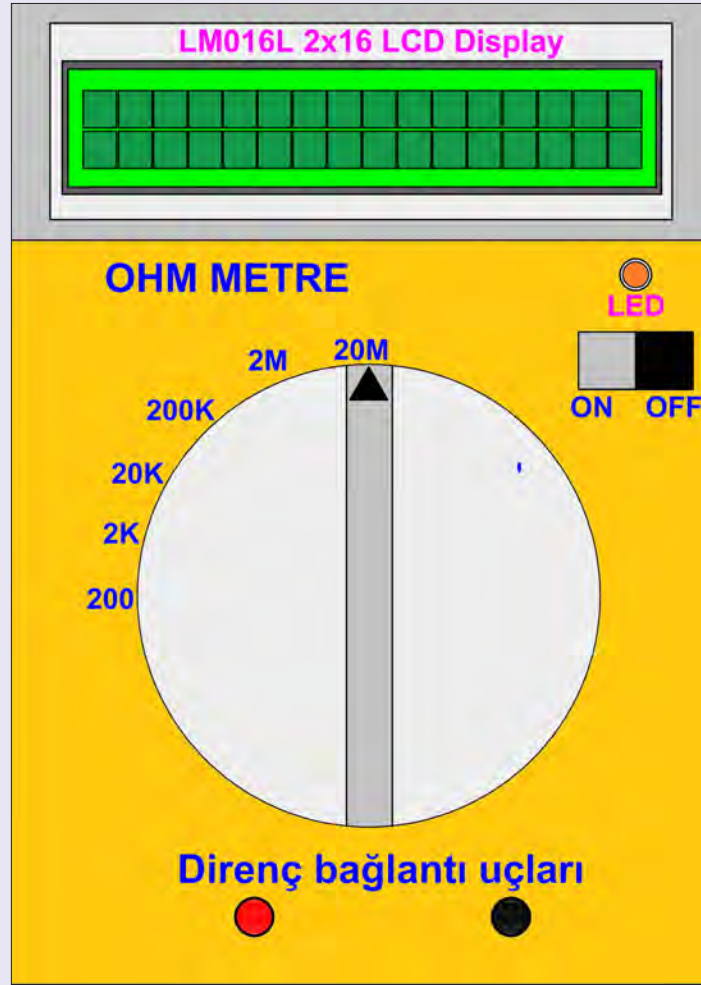
- PV panelli sokak aydınlatma projesi uygun direk üzerinde yapılacaktır (Görsel B.5).
- Uygun güçte ve ölçülere sahip solar panel hücresi kullanılacaktır. Özellikle 100 W üstü PV panelleri seçilmelidir.
- PV panel için MPPT şarj kontrol cihazı kullanılacaktır. Bu cihaz PV ile uyumlu olmalıdır. Özellikle 40 Ah (5 A) üstü bir cihaz tercih edilmelidir.
- Uygun özelliklere sahip LED lambalı sokak armatürü kullanılacaktır. Bu armatürün minimum özellikleri 12-18 W, 1800-3000 lümen ve 150 lm/W olmalıdır.
- Uygun yüksekliğe sahip sokak direği (özellikle galvanize saclı) tercih edilmelidir.
- Armatür ve PV paneli uygun yüksekliğe uygun donanımlar ile monte edilmelidir.
- Direk için kontrol panosu kullanılacaktır. Bu kontrol panosunun ölçüleri yeterli genişlikte olmalıdır.
- Kontrol panosu içerisinde güç, kontrol ve şarj üniteleri bulunacaktır.
- Kontrol ünitesinde çeşitli mikrodenetleyiciler (İstenirse PLC de kullanılabilir.) kullanılacaktır. Bu ünite tüm sistemi kontrol edecek, lamba ve USB şarjlarının kontrollerini sağlayacak, PV'den gelen elektrik üretimi değerlendirecek ve şebekeden gelen AC gerilimi gerekli zamanda yönlendirecektir. Ayrıca PV'de üretilen ve aküde depolanan enerjiyi izleyen bir bölüm de olacaktır.
- Güç ünitesinde gelen DC ve AC enerjileri kontrol ünitesinin yönlendirmesi ile ilgili birimlere (şarj ünitesi, şarj kutusu ve armatür) röleler vasıtasıyla aktaracaktır.
- Şarj ünitesinde belirli sayıda akü grubu vardır ve bu aküler vasıtasıyla gelen enerji şarj edilir. Bu ünite ise gelen DC enerjileri depolar ve talep edilirse çıkışlara yönlendirir.
- Güç ünitesi üzerinden tüm çıkış birimlerine ayrı hat çekilecektir. Bu hatta kullanılacak kablo tipi ve özelliği, belirlenen sistemin akım dayanımına uygun olmalıdır.
- USB çıkışlı bir şarj ünitesi, korunaklı bir kutu içeresine yerleştirilecektir. Bu bölümde en az 3 adet USB şarj çıkışı olacaktır. Bu çıkışlardan en az birisi hızlı şarj özelliğini desteklemelidir.
- Kontrol ünitesi üzerinden haberleşme sağlayacak bir modül eklenecektir. Bu modül ile sistemin takibi yapılabilecektir.
- PV panelinin yönü Güneş'in yönüne uygun şekilde yerleştirilecektir. Bu nedenle armatür ile çakışması durumunda armatür yüksekliği değiştirilecektir.

Not: Çalışma örnek bir sokak lambası üzerinde tasarlanmıştır. İstenirse bahçe aydınlatma, koridor veya oda aydınlatması gibi farklı tasarımlar üzerinde de uygulanabilir. Seçilecek bölgenin gücüne uygun PV kullanılmalıdır.



6. Proje: Direnç Ölçümü

Görsel B.6'da CAD ortamında tasarlanan direnç ölçümü projesi görülmektedir.



Görsel B.6: Direnç ölçümü projesi

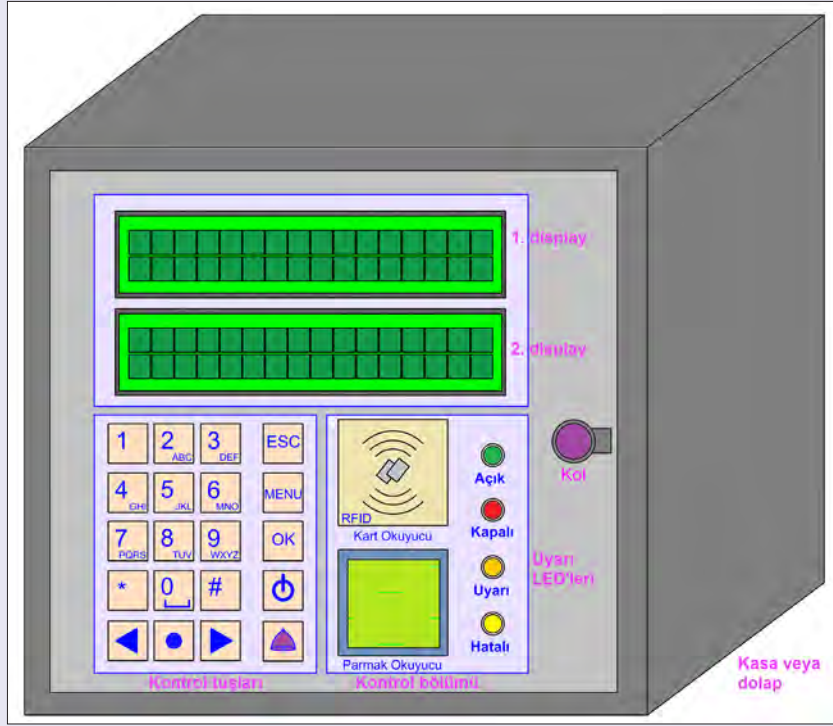
İSTENENLER

- Direnç ölçüm projesinde plastik tip kutu kullanılacaktır (Görsel B.6).
- Direnç ölçüm sonuçları 2x16 LCD display ekranda görüntülenecektir. İsteğe bağlı olarak grafik LCD veya TFT ekran da kullanılabilir.
- Komütatör anahtar olarak 1x6 veya 2x6 rotary anahtar kullanılacaktır.
- Ohm metre beslemesinde 9V pil kullanılacaktır.
- Ohm metrenin kullanılmadığı durumlarda enerji harcanmaması için ON/OFF anahtarı kullanılacaktır.
- Ölçüm problemleri kısa devre olduğunda sesli uyarı verecek ve LED yanacaktır.



7. Proje: Alarmlı Şifreli Kilit

Görsel B.7'de CAD ortamında tasarlanan alarmlı şifreli kilit projesi görülmektedir.



Görsel B.7: Alarmlı şifreli kilit projesi

İSTENENLER

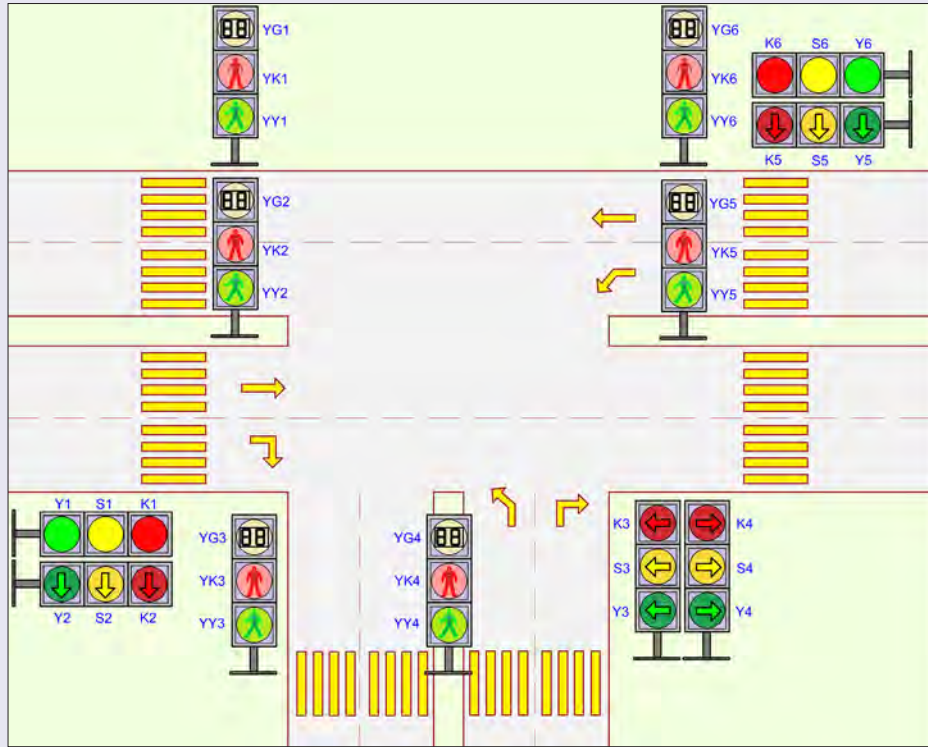
- Alarmlı şifreli kilit uygulaması uygun tip kasa üzerinde yapılacaktır (Görsel B.7).
- Kasa üzerinde iki adet 2x16 LCD display ekran bulunacaktır. İsteğe bağlı olarak grafik LCD veya TFT ekran da kullanılabilir.
- 1. display yazı ve bilgi alanını, 2. display ise şifre ve kod alanını gösterecektir.
- Kasa kontrolü için şifre, RFID okuyucu ve parmak okuyucu kontrolleri kullanılacaktır.
- 4 ila 8 haneli arası şifre girişi yapılabilecek ve belirlenen şifreler hafızada saklanacaktır.
- Hafızada en fazla 10 farklı şifre kaydı saklanabilecektir. Ayrıca bu kullanıcıların RFID ve parmak okuyucu bilgileri de paralelinde saklanabilecektir.
- Kasayı açmak için üç kontrol yönteminden istenilen herhangi biri, ikisi veya üçü seçilecektir.
- Birden fazla mikrodenetleyici kullanılacaktır.
- Şifre üst üste 3 kez hatalı girildiğinde uyarı alarmı belirli süre çalacaktır. Bunu durdurmak için yönetici şifresinin girilmesi, yönetici parmak okuyucunun aktif olması ve/veya yönetici RFID kartının okutulması gerekecektir.
- Her durum için durum LED'leri yerleştirilecektir.
- Ayarların yapılması için sistem içeresine gerekli menüler oluşturulacak ve ilgili tuş takımları da bu sisteme entegre edilecektir.
- Kasaya farklı tip fonksiyonlar/uygulamalar eklenebilecektir.

Not: Kasa haricinde bir oda kapısı veya pano kapağı girişine bu sistem tasarlanabilir. Böyle olursa RFID ve parmak okuyucuların da sistemde olması gerekir.



8. Proje: Trafik Kavşak Kontrol

Görsel B.8'de CAD ortamında tasarlanan trafik kavşak kontrol projesi görülmektedir



Görsel B.8: Trafik kavşak kontrol projesi

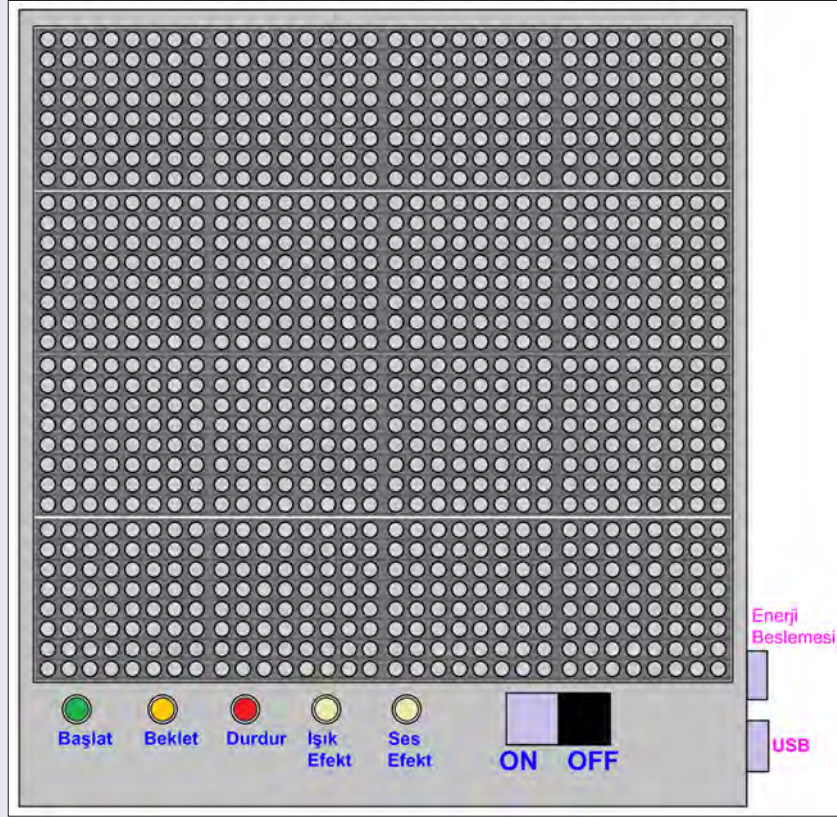
İSTENENLER

- Üç kavşaklı bir bölgenin trafik sinyalizasyon uygulaması yapılacaktır (Görsel B.8).
- Trafik lambalarında işaretli olmayanlar düz yoldan, işaretli olanlar ise belirlenen yoldan araçların geçişini gösterecektir.
- Yaya lambalarında sayıcılar (YG1-YG6) bulunacaktır. Bu sayıcılar, kırmızı ve yeşil ışıkların-yanma sürelerini kendi renklerinde gösterecektir. Belirlenen t1 saniye kırmızı ve t2 saniye yeşil rengi sayacak ve sayma işlemi sondan geriye doğru olacaktır.
- Her yaya yolu ışığı, kendi önündeki yaya alanından sorumlu olacaktır. Bu yaya yolunda yeşil yanarken hiçbir araç geçişi olmayacaktır. Aynı şekilde araç geçişi olduğu anda yaya geçişi olmayacaktır.
- Her lambanın yanma ve sönme sürelerini gösteren grafikler ve bu sürelerle uygun program oluşturulacaktır.
- Bir kırmızı ışığın yanma süresi 10-60 saniye arasında belirlenecektir (K1-K6). Yeşil ışığın yanma süresi değişkenlik gösterebilir (Y1-Y6).
- Yeşilden sarıya geçişte yeşil ışık 5 saniyede 5 defa yanıp sönecek ve ardından sarı lam-ba yanarak ilgili kırmızı lamba aktif olacaktır. Kırmızıdan sarıya geçiş direkt olacaktır.
- Araçların geçme şekilleri uygun biçimde planlanacaktır.
- Devre uygun bir zemin üzerinde maket formatına yakın şekilde yapılacaktır.
- Sayıcılarda 7 segment display (2xtekli veya 1xikili) kullanılacaktır.
- Birden fazla mikrodeneleyici kullanılabilir.
- Çalışmaya haricî fonksiyonlar/tasarımlar/uygulamalar eklenebilir.
- Çalışma için uygun tip zemin ve renkler kullanılacaktır.



9. Proje: Işık ve Ses Efekt

Görsel B.9'da CAD ortamında tasarlanan ışık ve ses efekt projesi görülmektedir.



Görsel B.9: Bilgisayar kontrollü ışık ve ses efekti projesi

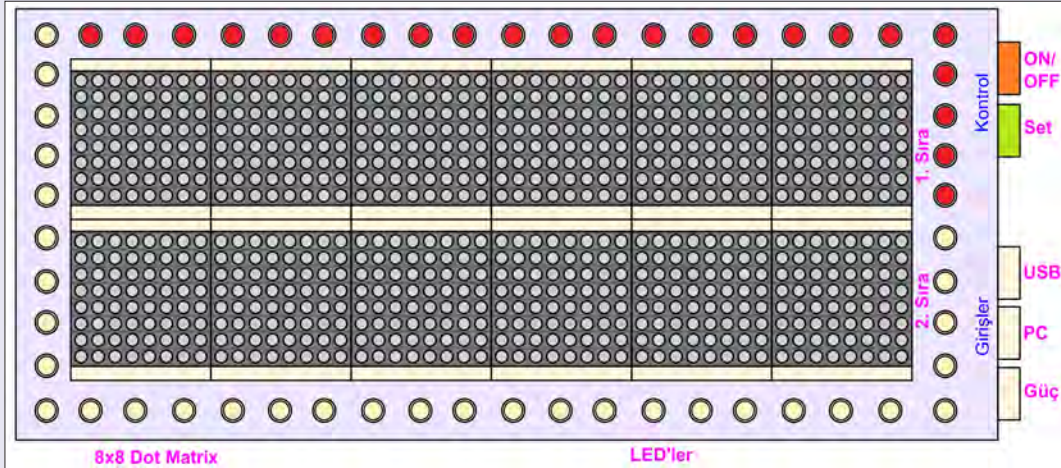
İSTENENLER

- Bilgisayardan kontrollü ışık ve ses efekti üreten proje yapılacaktır (Görsel B.9).
- Baskı devre hazırlandıktan sonra uygun levha sac veya fleksiglas üzerine monte edilecektir.
- 8x8 dot matris kullanılarak ekran hazırlanacaktır. Ekran toplam 32 satır ve 32 sütundan oluşturulacaktır.
- Bilgisayardan oluşturulan yazılım üzerinden ışık animasyon efektleri hazırlanacak ve USB kablo ile panele veri aktarım gerçekleştirilecektir.
- Ses dosyaları *.wav ses dosyası formatında hazırlanarak SD kartta yüklenecektir. Dosya isimleri ses efekt numarasına göre düzenlenecektir (1.wav, 2.wav vb.).
- Kullanılan yazılımdan ışık animasyon hız ayarı yapılabilecektir.
- "Başlat" butonuna basıldığında ışık ve ses efekti başlayacaktır.
- "Duraklat" butonuna basıldığında ışık ve ses efekti bekleme modunda kalacaktır.
- "Durdur" butonuna basıldığında ışık ve ses efekti duracaktır.
- Işık ve ses efekti çalışırken "Işık Efekt" butonuna basıldığında sadece ışık efekti görünlenecektir.
- Işık ve ses efekti çalışırken "Ses Efekt" butonuna basıldığında sadece ses efekti oynatılacaktır.
- Güç girişi noktasından haricî güç kablosu adaptörü bağlanacaktır.
- Ekran dot matris yerine doğrudan LED kullanılarak da oluşturulabilir.



10. Proje: Bilgisayar Kontrollü Kayan Yazı

Görsel B.10'da CAD ortamında tasarlanan bilgisayar kontrollü kayan yazı projesi görülmektedir.



Görsel B.10: Bilgisayar kontrollü kayan yazı projesi

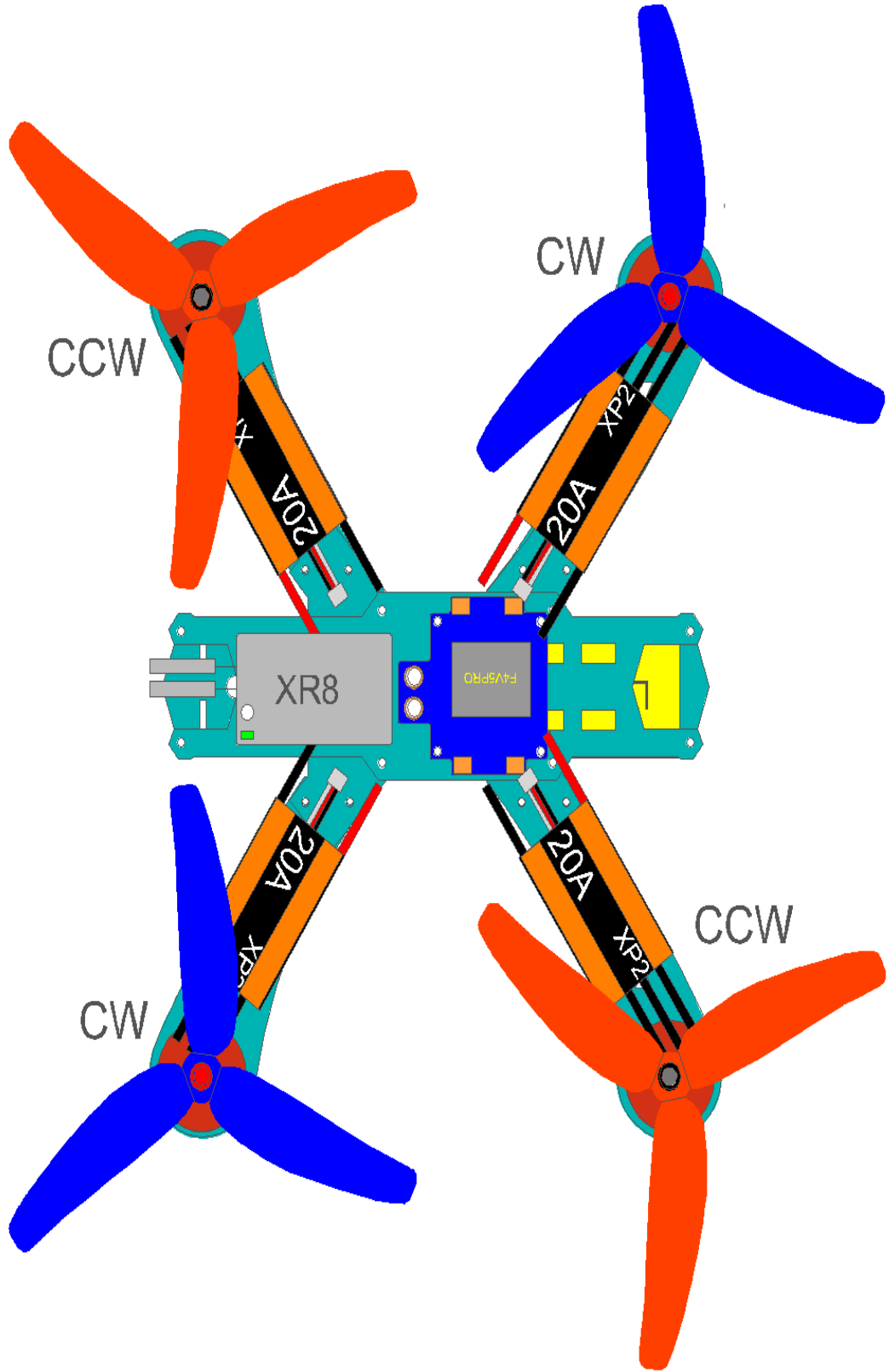
İSTENENLER

- Bilgisayardan kontrollü kayan yazı projesi yapılacaktır (Görsel B.10).
- Panel uygun bir levha sac üzerine monte edilecektir.
- 8x8 dot matris kullanılacaktır. Bu göstergeden 2 satır ve 6 sütunlu display olacak şekilde toplamda 16 satır ve 48 sütunlu bir yazı alanı oluşturulacaktır.
- Her satır ayrı bir yazı geçişini gösterecektir. İstenirse senkronize olacak şekilde bir bütün olarak gösterim yapabilecektir.
- Bilgisayarda oluşturulan yazılım üzerinden ilgili yazılar hazırlanacak ve uygun bağlantı kablosu (COM, USB vb.) ile panele veri aktarım gerçekleştirilecektir.
- Panele yazı aktarımı haricî olarak USB portu üzerinden de gerçekleştirilebilecektir.
- Kullanılan yazılımdan yazılar ile ilgili animasyon ve hız ayarı yapılabilecektir.
- Panel çevresine uygun sayıda LED yerleştirilecektir. Bu LED'lerin görevi panel üzerinde çeşitli animasyon efektleri uygulamaktır. Bunun kontrolü bilgisayardan ya da panel üzerine yerleştirilen "SET" butonu ile otomatik ayarlanabilecektir.
- Kullanılacak LED rengi panelin görünümü etkilemeyecek türden olmalıdır. Ayrıca şerit LED de tercih edilebilir.
- Kullanılacak LED'ler panelin üzerinde ya da yan çerçevelerinde olabilir.
- Kullanılacak metinler grup tarafından belirlenecektir.
- Güç girişi noktasından haricî güç kablosu adaptörü bağlanacaktır.
- Panel dot matris yerine doğrudan LED'ler ile oluşturulabilir.
- Türkçe karakterlerin ve matematiksel sembollerin gösterilmesinde sorun yaşanmaması için gerekli tedbirler alınacaktır.

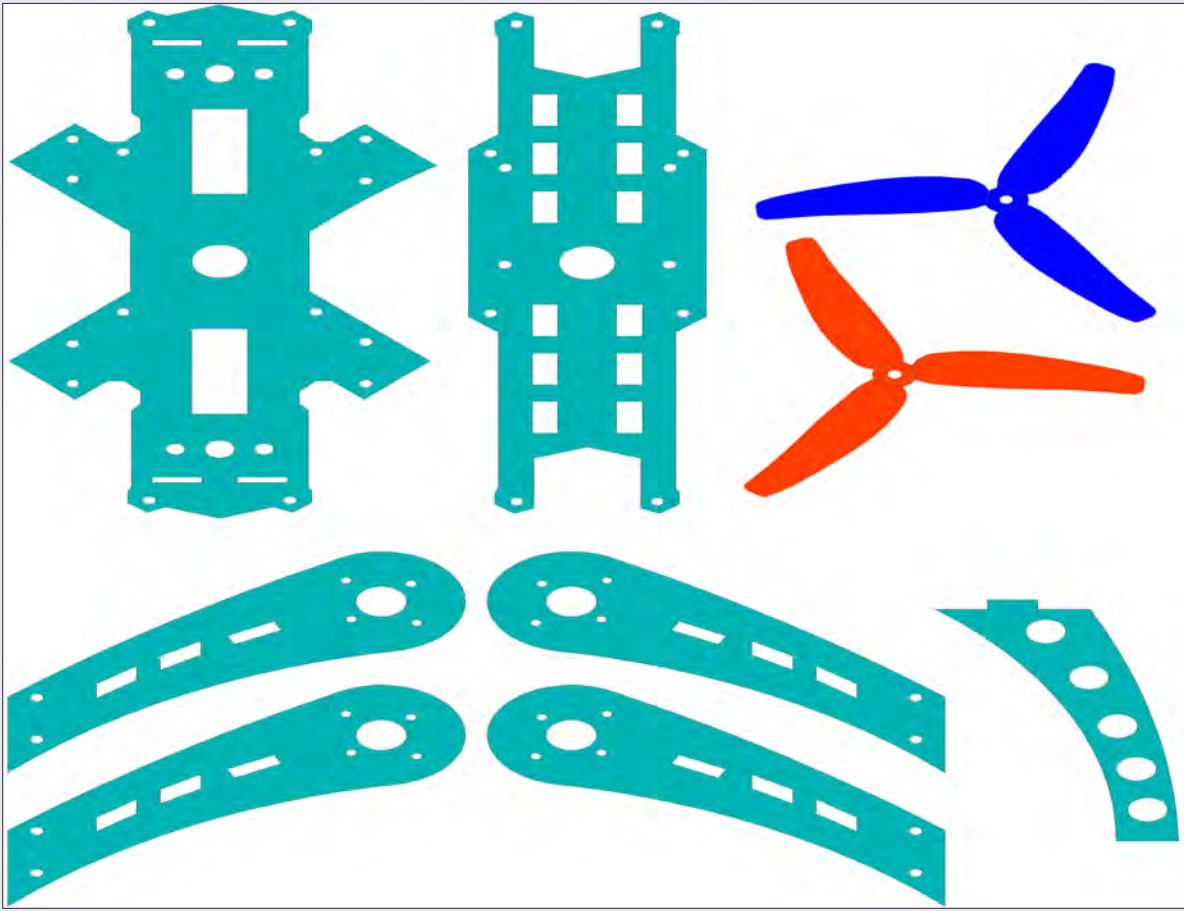


11. Proje: Dron

Görsel B.11'de CAD ortamında tasarlanan dron projesi görülmektedir.

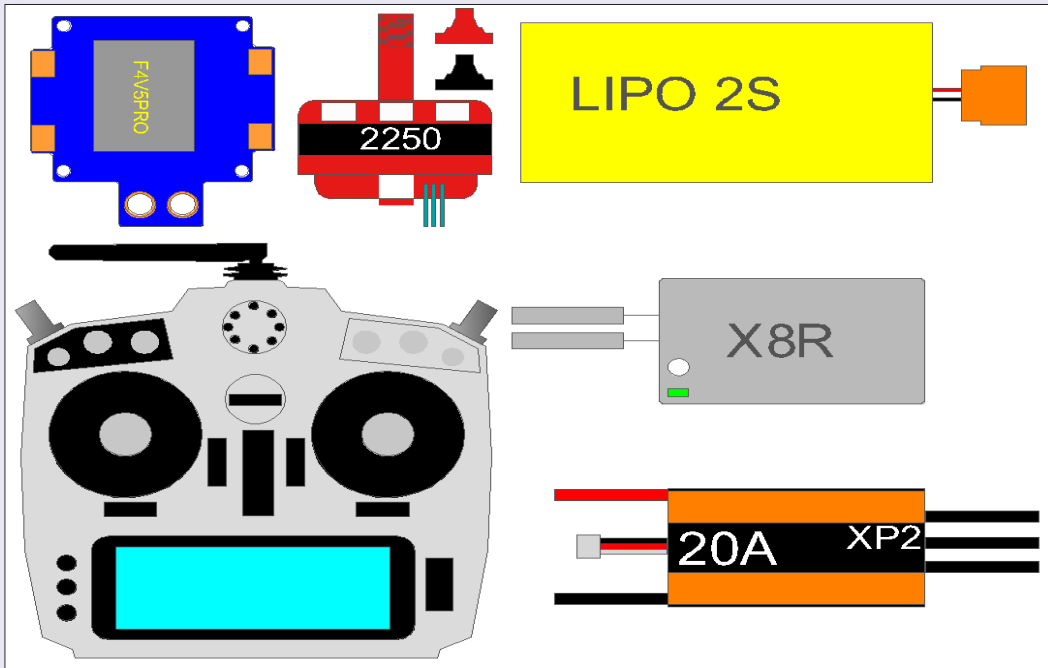


Görsel B.11: Dron projesi



Görsel B.12: Dron gövde parçaları

Görsel B.13'te CAD ortamında tasarlanan dron elektronik parçaları görülmektedir.



Görsel B.13: Dron elektronik parçaları

İSTENENLER

- Dron gövde parçaları hazır alınacak veya verilen CAD dosyasından lazer kesimle elde edilecektir (Görsel B.11).
- Pervaneler 5 inç, üç yapraklıdır. Saatin dönüş yönünde (CW-mavi) ve saat dönüşünün tersi yönde (CCW-turuncu) olmak üzere 2 takım olacaktır.
- Gövde alt parçası, gövde üst parçası ve kolların birleştirilmesinde 2,5 mm plastik vida ve somun kullanılması tercih edilir ancak metal vida ve somun da kullanılabilir.
- Alt ve üst gövdenin birleştirilmesinde plastik aralayıcı (distans) kullanılması önerilir. Plastik aralayıcıların uzunluğu kullanılan LIPO pil yüksekliğine uygun olmalıdır.
- Ayaklar 3 mm kalınlığında fleksi-glass malzemeden kesilecektir. Kol üzerindeki dikdörtgen kısma yerleştirilip yapıştırılacaktır. İsteğe göre hazır ayaklar alınıp gövdede uygun kısma yerleştirilebilir.
- Uzaktan kumanda FrSky Taranis X9D 2,4GHz'dir. İsteğe göre farklı kumanda kullanılabilir.
- Uzaktan kumanda alıcısı FrSky X8R modelidir. Kumandaya uygun farklı alıcı da kullanılabilir.
- Uçuş kontrol kartı (FCC) F4V5PRO modelidir. Farklı uçuş kontrol kartı kullanabilir. Uçuş kontrol kartı plastik vida ve somunla veya kablo bağıyla sabitlenecektir.
- Farklı uçuş kontrol kartı kullanıldığında kartın alıcı modül ile uyumlu olmasına dikkat edilmelidir veya uygun bir alıcı modülü kullanılmalıdır.
- Uzaktan kumanda, alıcı modül ve uçuş kontrol kartı birbirleriyle iletişim sağlayamazsa drone çalışmayacaktır. Bu bileşenlerin birbirlerine uygun modeller olması önemlidir.
- Fırçasız motorlar RS2205 modelidir. Saatin dönüş yönünde (CW) iki adet, saat dönüşünün tersi yönde iki adet olmak üzere toplam dört adet fırçasız motor kullanılacaktır.
- Fırçasız motor sürücüleri (ESC) 20A modelidir. Dört adet kullanılacaktır.
- Fırçasız motor sürücüleri mümkün olduğu kadar gövdeye yakın noktaya ve birbirlerine simetrik şekilde yerleştirilecektir.
- DC gerilim besleme kaynağı olarak 2S 7,4 V 1100mAh LIPO pil kullanılmaktadır. Ancak gövdeye uygun daha yüksek akımlı LIPO pil kullanılabilir.
- LIPO pil, pil kayışı veya kablo bağıyla gövdeye sıkıca tutturulmalıdır.
- Elektronik malzemeler gövde üzerindeki orta noktada bulunan dairesel kesime eşit yük dağılımı yapacak şekilde dengeli dağıtılacaktır.
- Dronun dengeli uçuşurulması ve kolay kontrol edilebilmesi için ağırlık merkezinin orta noktada oluşturulması önemlidir.

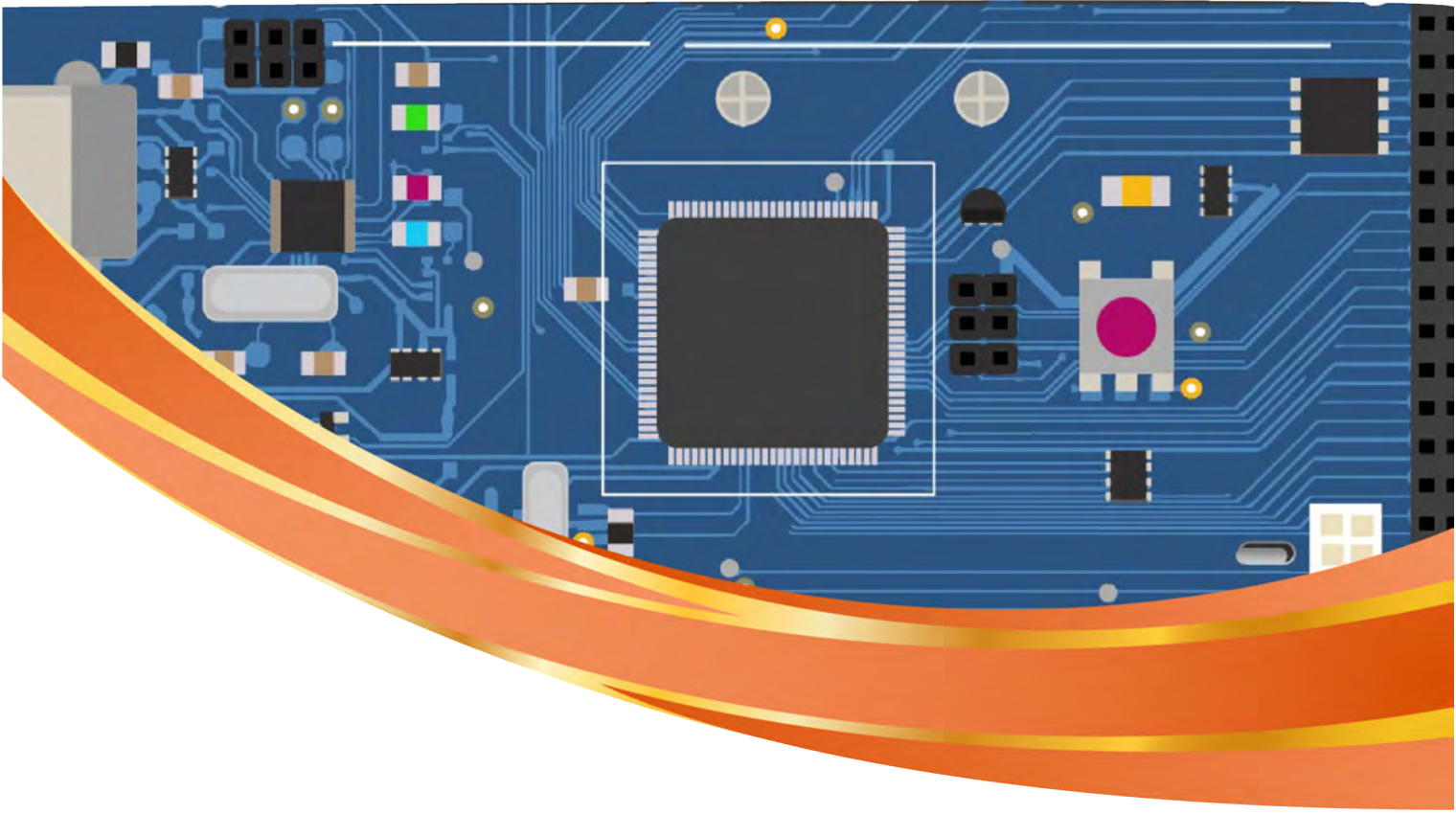


UYGULAMA DEĞERLENDİRME TABLOSU

Aşağıda uygulama değerlendirme ölçütleri (Tablo B.2) görülmektedir. Birinci öğrenme biriminde yapılan uygulama çalışmalarını bu ölçütlere göre değerlendiriniz. Her ölçütün değerlendirme sonucunu karşısındaki puanlama tablosunda işaretleyiniz (v) ve tablodan alınan toplam puanı hesaplayınız.

Tablo B.2: Uygulama Değerlendirme Tablosu

Değerlendirme Ölçütleri		Puanlama Ölçütleri					
		Kötü (0)	Yeterli (1)	Orta (2)	İyi (3)	Çok İyi (4)	Mükemmel (5)
1.	Projede kullanılacak mikrodenetleyiciyi belirlendi.						
2.	Proje kodları uygun yazılım editöründe oluşturuldu.						
3.	Proje kodları derlendi ve kontrol edildi.						
4.	Proje kodları mikrodenetleyiciye yüklendi.						
5.	Proje devre tasarımı şema çizim programında çizildi.						
6.	Proje devresinin simülasyonunu yapıldı.						
7.	Projenin kurulacağı alan (breadboard, PCB vb.) belirlendi.						
8.	Projenin baskı devresinin tasarımı ya da breadboard kurulumu yapıldı.						
9.	Projedeki entegrelerin ve elemanların kullanılacak uçları kataloglar yardımıyla belirlendi.						
10.	Projede kullanılan elemanların sağlamlık kontrolleri yapıldı.						
11.	Proje malzemeleri devreye uygun şekilde yerleştirildi.						
12.	Projeye uygun kablolar belirlendi.						
13.	Proje kablo bağlantıları doğru yapıldı.						
14.	Projede, belirlenen parçalar kullanıldı.						
15.	Çalışmanın gerçekleştirilme aşamaları düzenli olarak kontrol edildi.						
16.	Projede oluşturulan devre istenilen şekilde çalıştı.						
17.	Projeye oluşturulan devrenin çalışması detaylı olarak incelendi.						
18.	Uygulamalar verilen sürede tamamlandı						
19.	Proje, zamanında teslim edildi.						
20.	Proje raporu eksiksiz teslim edildi.						
TOPLAM							
Değerlendirme sonucunda 100 puan üzerinden toplam puan almıştır.							
Öğrencinin Adı-Soyadı:							
Sınıfı-Numarası:							
İmza:							
Öğretmenin Adı-Soyadı:							
İmza:							



EK-C

ARM32 MİKRODENETLEYİCİLER

1. Uygulama: LED Animasyon
2. Uygulama: RGB LED
3. Uygulama: Zamanlayıcı (Timer)
4. Uygulama: Buton LED
5. Uygulama: ADC Sensör
6. Uygulama: PWM
7. Uygulama: NOKIA 5110 LCD Display
8. Uygulama: LCD'de Sensör Verilerini Gösterme
9. Uygulama: LCD'de DHT11 Sıcaklık ve Nem Sensör Bilgilerinin Gösterilmesi
10. Uygulama: LCD'de Dijital Saat



ARM SERİSİ MİKROİŞLEMCİLER

Bu bölümde ARM32 mikrodeneleyici ile ilgili temel bilgiler ve uygulamalar gösterilmektedir.

C.1. ARM (Advanced RISC Machine)

“Acorn” adlı bilgisayar firması tarafından ilk olarak 1981 yılında “Acorn RISC* Machine” adıyla tasarlanmış RISC mimarisidir. Firma adı 1990 yılında “Advanced RISC Machine” olarak değiştirmiştir. Firma son olarak 1998 yılında günümüzdeki ismi olan “ARM LTD.” adını almıştır. ARMv1 ile başlayıp günümüze kadar geliştirilen bu mimari 32 bitlik yapısı sayesinde 8 bitlik işlemcilere göre çok daha hızlıdır. Ayrıca düşük güç tüketimi ve yüksek performansı sayesinde günümüzde yaklaşık %75'lik oranla gömülü sistemler üzerinde en çok kullanılan işlemci mimarisidir. ARM bir mimari türü olduğundan geliştirici firma fiziksel işlemci üretmek yerine bu mimariyi işlemci üreten diğer firmalara lisanslayarak gelir elde etmektedir. ARM mimarisi lisansını elinde bulunduran Apple, Samsung, NVIDIA gibi başlıca firmalar farklı isimlerde birçok işlemci üretmiştir. Nvidia-Tegra, Samsung Hummingbird, Exynos ve Apple-A4, A5, A5X vb. işlemciler buna örnektir. ARM firması verilerine göre sadece 2010 yılında 6,1 milyar adet ARM mimarisi tabanlı işlemci üretilmiştir. Bu işlemcilerin çoğu akıllı telefonlarda, bir kısmı dijital televizyonlarda, geri kalanı ise set üstü kutuları (dijital uydu alıcıları, video oynatıcıları vs.) ve mobil bilgisayarlarda kullanılmıştır.

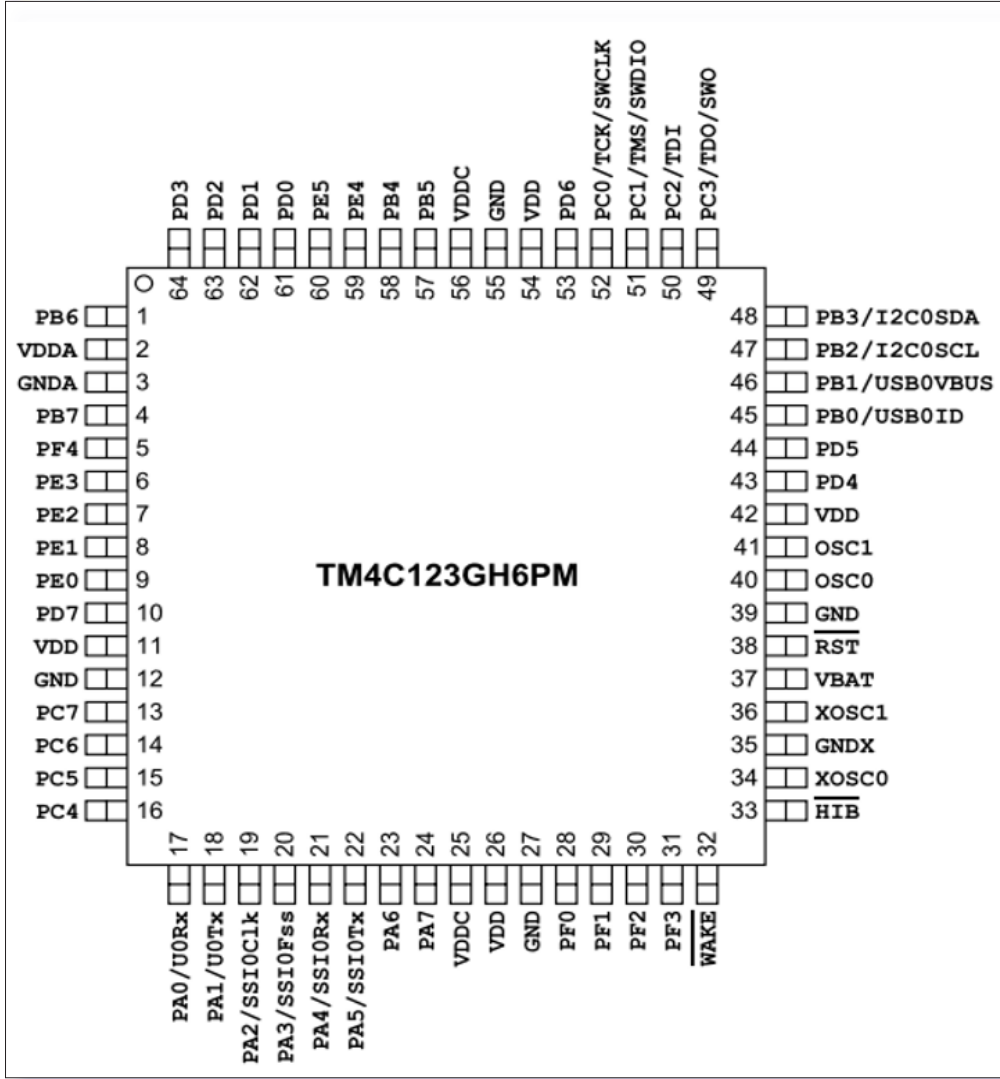
C.2. ARM32 Serisi Mikroişlemcilerin Yapısı ve Çeşitleri

ARM mimarisi; klasik, gömülü (embedded) sistemlere yönelik ve uygulamaya yönelik olmak üzere üç temel gruba ayrılabilir. Klasik ARM işlemci grubunda ARM7, ARM9 ve ARM11 yer alır. Bu mimari yapıları, daha çok motor kontrolü gibi donanımına yakın işlemlerde kullanılır. İkinci grup gömülü işlemci grubudur. Bu grupta Cortex M ve R serileri yer alır. Bunlar matematiksel formüllere dayalı olarak kesin sonuca yönelik çalışması gereken, düşük güç tüketimi isteyen uygulamalarda kullanılır. Son olarak da uygulama işlemcileri Cortex-A5, Cortex-A8, Cortex-A9, Cortex-A15 olarak sınıflandırılabilir. Bunlar da akıllı telefonlar, tablet bilgisayarlar gibi yüksek performans gerektiren, son kullanıcı uygulamalarına yönelik platformlarda kullanılır. ARM tabanlı mikroişlemcilerin yanı sıra aynı tabanlı mikrodeneleyiciler de üretilmektedir. Mikroişlemciler genellikle sadece işlem ünitesi ve temel birimlerden oluşur, harici olarak bellek ve diğer giriş/çıkış birimlerine ihtiyaç duyar. Bu mikroişlemciler daha çok uygulama yazılımlarının çalıştığı sistemlerde görülür. Mikrodeneleyiciler ise kendi içerisinde giriş/çıkış birimleri, çeşitli donanım birimleri (SPI, I2C, PWM), RAM ve kalıcı hafıza (genellikle NOR veya NAND flash) bulunan komple sistemlerdir. Mikrodeneleyiciler daha çok elektronik tasarım alanında kullanılır. Cep telefonları ve tablet bilgisayarlardan çamaşır makinelerine, otomobillere, fotoğraf makinelerine kadar birçok ürünün içinde ARM tabanlı işlemciler/deneleyiciler bulunmaktadır.

C.3. Cortex - M Serisi Mikrodeneleyiciler

ARM mimarisinin mikrodeneleyici ailesinden biri olan Cortex M, piyasada bulunan 8 ve 16 bitlik mikrodeneleyicilere rakip olarak ortaya çıkmış, 32 bitlik mimariye sahip işlemci serisidir. Bu serideki işlemciler düşük enerji tüketimine sahiptir ve düşük maliyetlidir. Serinin mikrodeneleyicileri, endüstriyel kontrol sistemleri, beyaz eşyalar, medikal cihazlar gibi çeşitli alanlarda kullanılmaktadır.

Görsel C.1'de TIVA TM4C123GXL geliştirme kartı donanım yapısı ve Görsel C.2'de TIVA TM4C123GXL geliştirme kartının PIN yapısı görülmektedir.



Görsel C.3: TIVA TM4C123GXL geliştirme kartı PIN yapısı

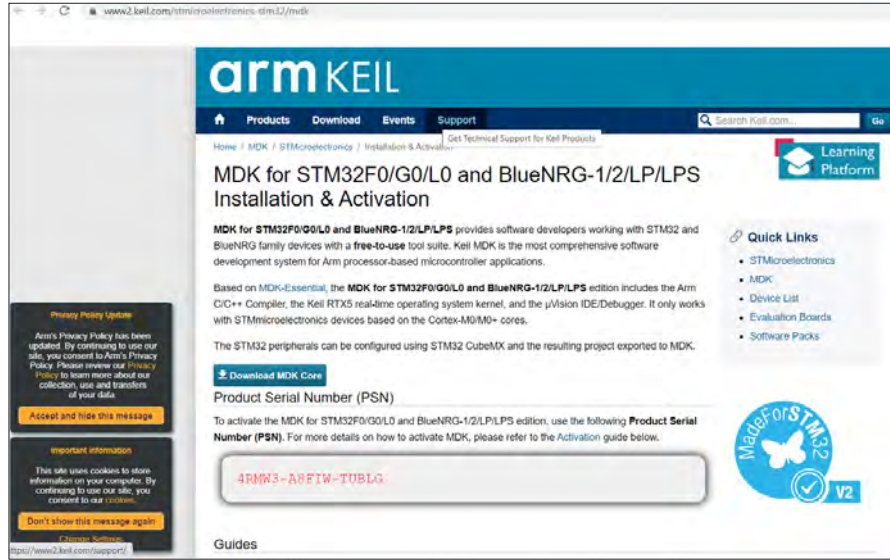
Entegrede A, B, C, D, E, F olmak üzere 6 adet port ve bunlara bağlı 43 adet I/O (giriş/çıkış) pini bulunmaktadır. 5 V tolerans (normalde 3,3 V giriшли iken 5 V giriшли olarak kullanılabilen) giriş olarak kullanılabilen bu pinlerden (PD4, PD5, PB0 ve PB1 hariç) giriş (sink) akımı olarak 18 mA akım geçebilmektedir (Görsel C.3).

C.4. ARMKEIL Yazılımın İndirilmesi ve Kurulması

ARM32 mikroişlemcisi için birçok yazılım kullanılabilir. Bu yazılımlar içinden **ARMKEIL uVision5** yazılımı tercih edilmiştir. Yazılımın 32 kB BIN dosyası, ücretsiz olarak indirilebilir.

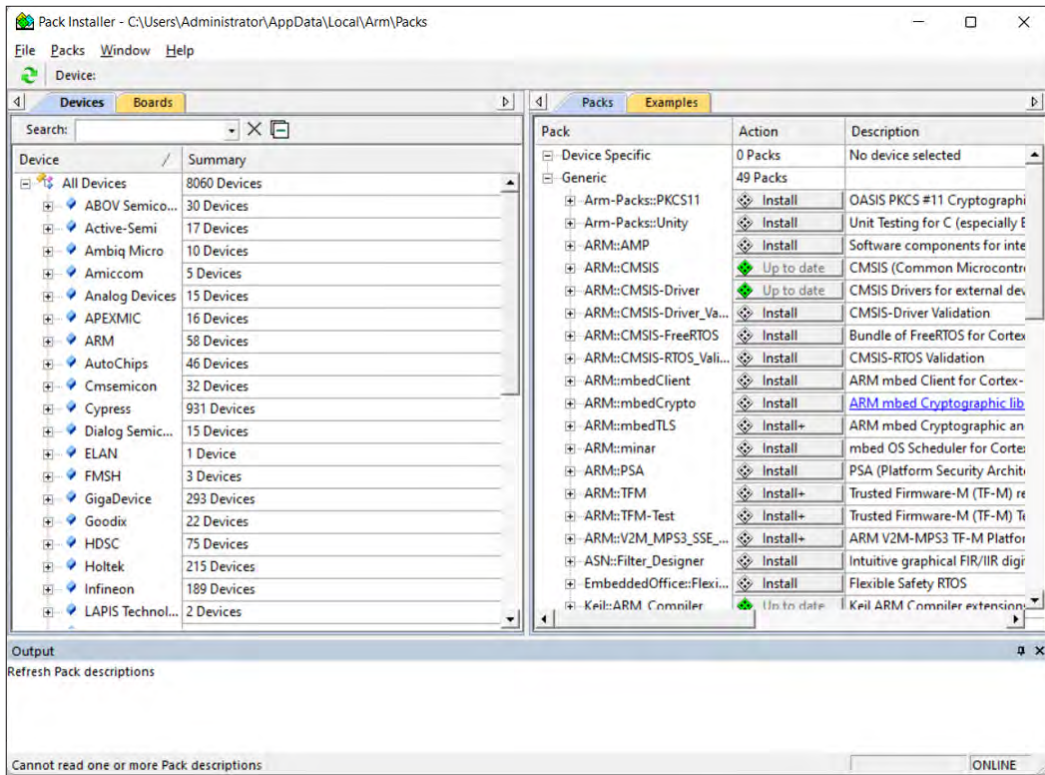
NOT

ARMKEIL uVision5 yazılımı aşağıda verilen linkten kolaylıkla indirilebilir (Görsel C.4).
<https://www2.keil.com/stmicroelectronics-stm32/mdk>

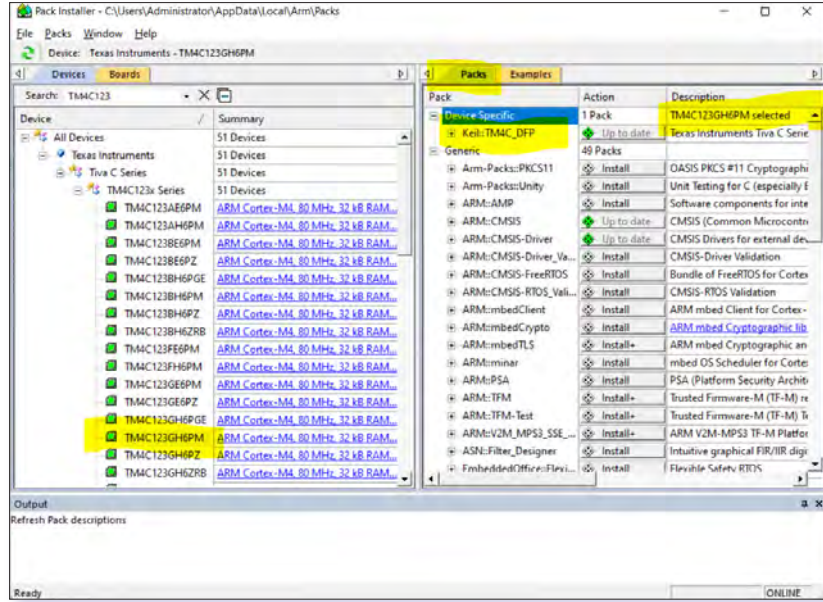


Görsel C.4: ARMKEIL programının indirme web sayfasının ekran görüntüsü

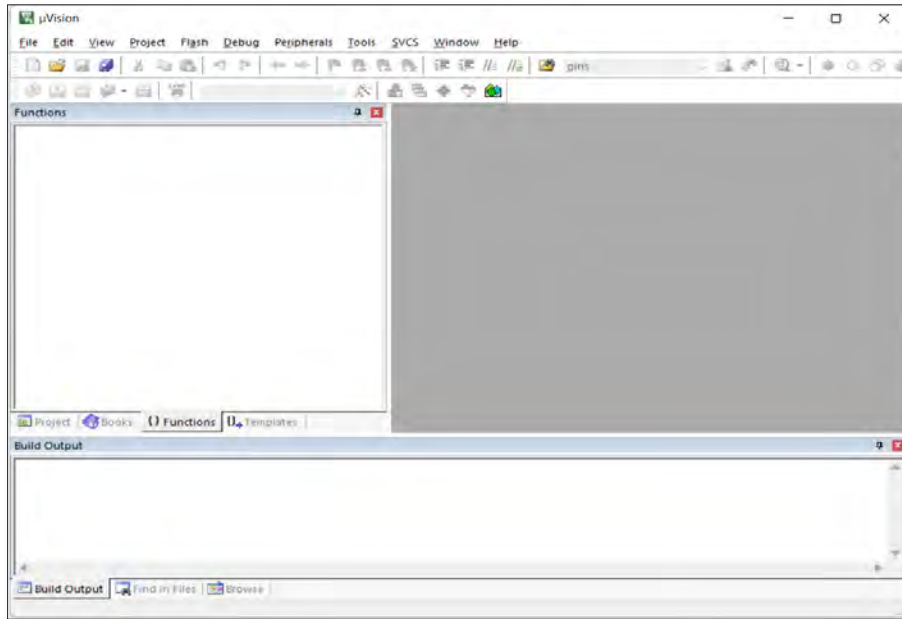
Verilen internet adresindeki **Download MDK Core** kısmından kurulum dosyası indirilir ve alta yer alan **Product Serial Number (PSN)** şifresi kullanılarak program aktif edilir. Kurulum işlemleri **ilerle** (Next) butonları ile gerçekleştirilir. Başlat menüsünden  simgesi tıklanarak çalıştırılır.  **ARMKEIL Pack Installer** (paket yükleyici-arm keyl pak installer) komutu internete bağlı iken çalıştırılır. <https://www2.keil.com> web sitesinden ilgili mikrodnetleyici paketlerinin bir listesi gelir (Görsel C.5). ARMKEIL paket yükleyicisindeki **search** (ara-sörç) kısmından **TM4C123** seri numaralı paket bulunur. Ekrana gelen paket listesinden **Tiva C Serisi TM4C123G6HPM** isimli seri numara bulunur ve paket yüklenir. Yükleme işlemi tamamlanınca **ARMKEIL** paket yükleyici ekranı kapatılır (Görsel C.6-C.7).



Görsel C.5: ARMKEIL paket yükleyicisinin ekran görüntüsü



Görsel C.6: ARMKEIL paket yükleyicisinden TIVA kartının bulunması ve paketin yüklenmesi



Görsel C.7: ARMKEIL programının ana ekranı

C.5. ARM32 Kontrol Kartı Sürücüsünün Yüklenmesi

TM4C123GXX kartı sürücü yazılımı, SW-ICDI-DRIVERS olarak bilinmektedir. Kontrol kartını bilgisayara tanıtmak için sürücüsünün yüklenmesi gerekir. Kontrol kartı bilgisayara takılı olduğunda aygıt yöneticisindeki simge üzerinde ünlem işareti varsa kart sürücülerini yüklenmemiştir.

Kontrol kartı sürücüsü (TM4C123Gxx), aşağıdaki linkten kolaylıkla indirilebilir (Görsel C.8).

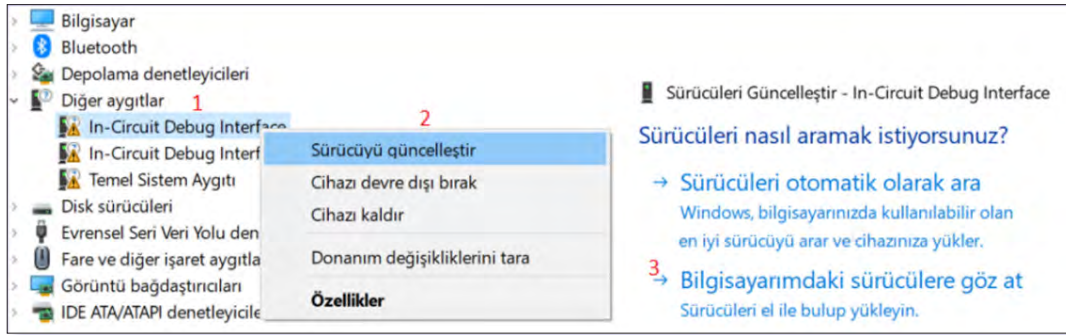
https://www.ti.com/tool/STELLARIS_ICDI_DRIVERS

Part Number	Buy from Texas Instruments or Third Party	Alert Me	Status	Current Version	Version Date
SW-ICDI-DRIVERS: Stellaris® ICDI Drivers - Current	Download 2	Alert Me	ACTIVE	v1.0	21-JAN-2016
SW-ICDI-DRIVERS-AR01: Stellaris® ICDI Drivers - OLD	Download		ACTIVE		

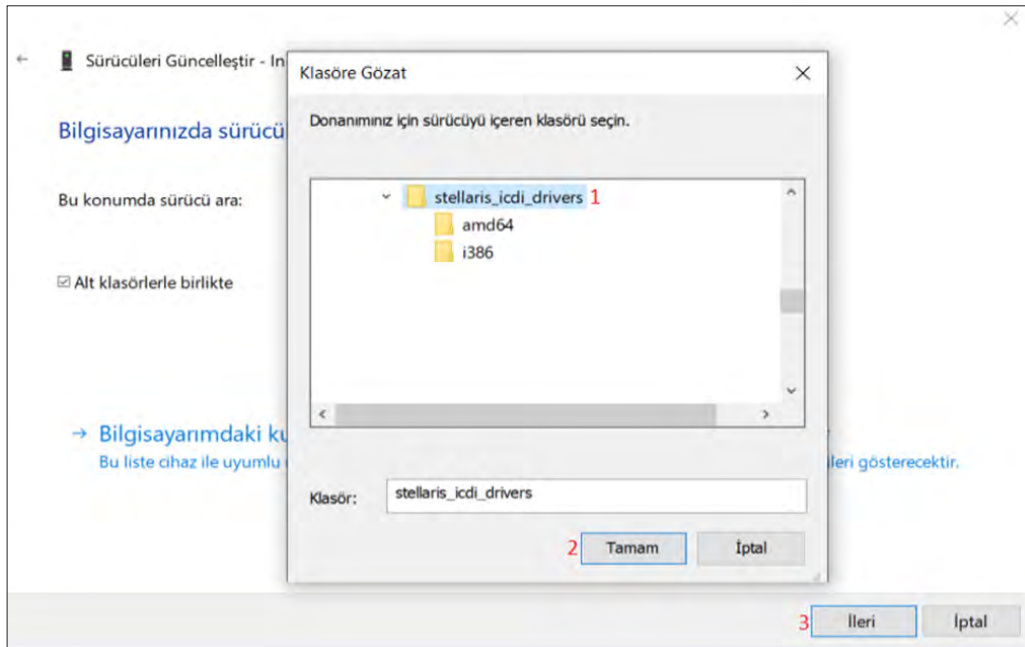
Görsel C.8: ARM mikrodnetleyici Windows sürücüleri indirme linki

Sürücünün yüklenmesi için şu adımlar izlenmelidir:

- Yukarıda belirlenen adresten kurulum dosyası indirilir (Dosyanın konumu **masaüstü** olsun.).
- Kontrol kartı, USB üzerinden bilgisayara bağlanır.
- Denetim Masası ➡ Aygıt Yöneticisi (⏏)
- Açılan pencereden **Diğer Aygıtlar** sekmesi genişletilir. Ünlem işareti bulunan **In-Circuit Debug Interface** kısmına sağ tıklanır ve **Sürücüyü güncelleştir** seçilir (Görsel C.9).



Görsel C.9: Farenin sağ tuşu ile güncellemenin başlatılması



Görsel C.10: Sürücü yazılımını yükleme



- Açılan pencereden **Sürücü Ara** kısmından **Gözet** butonuna basılır, masaüstüne indirilen “stellaris_icdi_drivers” dosyası seçilir ve ardından **Tamam** butonuna tıklanır. İşlem sonunda İleri butonuna basılarak donanım güncellenir. Diğer donanımlar için aynı işlemler tekrarlanır (Görsel C.10). Güncelleme tamamlandığında donanım isimlerinin üzerinde oluşan ünlem ifadeleri ortadan kalkmış olur. Bilgisayar her kontrol kartı için ayrı bir com port numarası (COM5) atamaktadır.

C.6. TIVAWARE Yazılımının Yüklenmesi ve Örnek Dosyanın Açılması

TIVAWARE yazılımı, Tiva C serisi için Texas Instrument firması tarafından geliştirilmiş kütüphaneler topluluğunu içerisinde barındırır. Aşağıdaki adresten kurulum dosyası indirilir ve program kurulumu yapılarak işlem tamamlanır.

TIVAWARE yazılımı, aşağıda verilen linkten kolaylıkla indirilebilir.

<https://www.ti.com/tool/download/SW-TM4C/2.2.0.295>

İndirilecek dosya: “SW-TM4C-2.2.0.295.exe”

Program, bilgisayarda aşağıdaki klasör alanına kurulum işlemini gerçekleştirir:

“C:\ti\TivaWare_C_Series-2.2.0.295\”

Aşağıdaki klasördeki örnek dosya çift tıklanarak ARMKEIL yazılımda çalıştırılır (Görsel C.11).

“C:\ti\TivaWare_C_Series-2.2.0.295\examples\boards\ek-tm4c123gxl\blink\blink.uvproj”

Name	Date modified	Type	Size
ccs	5.09.2020 13:56	File folder	
ewarm	5.09.2020 13:56	File folder	
gcc	5.09.2020 13:56	File folder	
rvmdk	5.09.2020 13:56	File folder	
blink.c	22.11.2021 09:34	C Source	4 KB
blink.ewd	22.11.2021 09:34	EWD File	17 KB
blink.ewp	22.11.2021 09:34	EWP File	21 KB
blink.icf	22.11.2021 09:34	ICF File	3 KB
blink.ld	22.11.2021 09:34	LD File	2 KB
blink.sct	22.11.2021 09:34	SCT File	2 KB
blink.uvopt	22.11.2021 09:34	UVOPT File	9 KB
blink.uvproj	22.11.2021 09:34	uVision4 Project	16 KB
blink_ccs.cmd	22.11.2021 09:34	Windows Comm...	3 KB
Makefile	22.11.2021 09:34	File	3 KB
readme.txt	22.11.2021 09:34	TXT File	2 KB
startup_ccs.c	22.11.2021 09:34	C Source	15 KB
startup_ewarm.c	22.11.2021 09:34	C Source	16 KB
startup_gcc.c	22.11.2021 09:34	C Source	16 KB
startup_rvmdk.S	22.11.2021 09:34	Assembler Source	17 KB

Using an MDK Version 4 Project

This is an MDK Version 4 project requiring Device Support for Cortex-M based devices.

To work with this project in MDK Version 5 you can:

Migrate to Device Pack

Most Cortex-M devices are now supported with Software Packs. You can download such a Software Pack and migrate to the new format.

Install Legacy Support

To continue working with the old project format install the MDK Version 5 - Legacy Support for Cortex-M devices.

Cancel

Missing Device Information

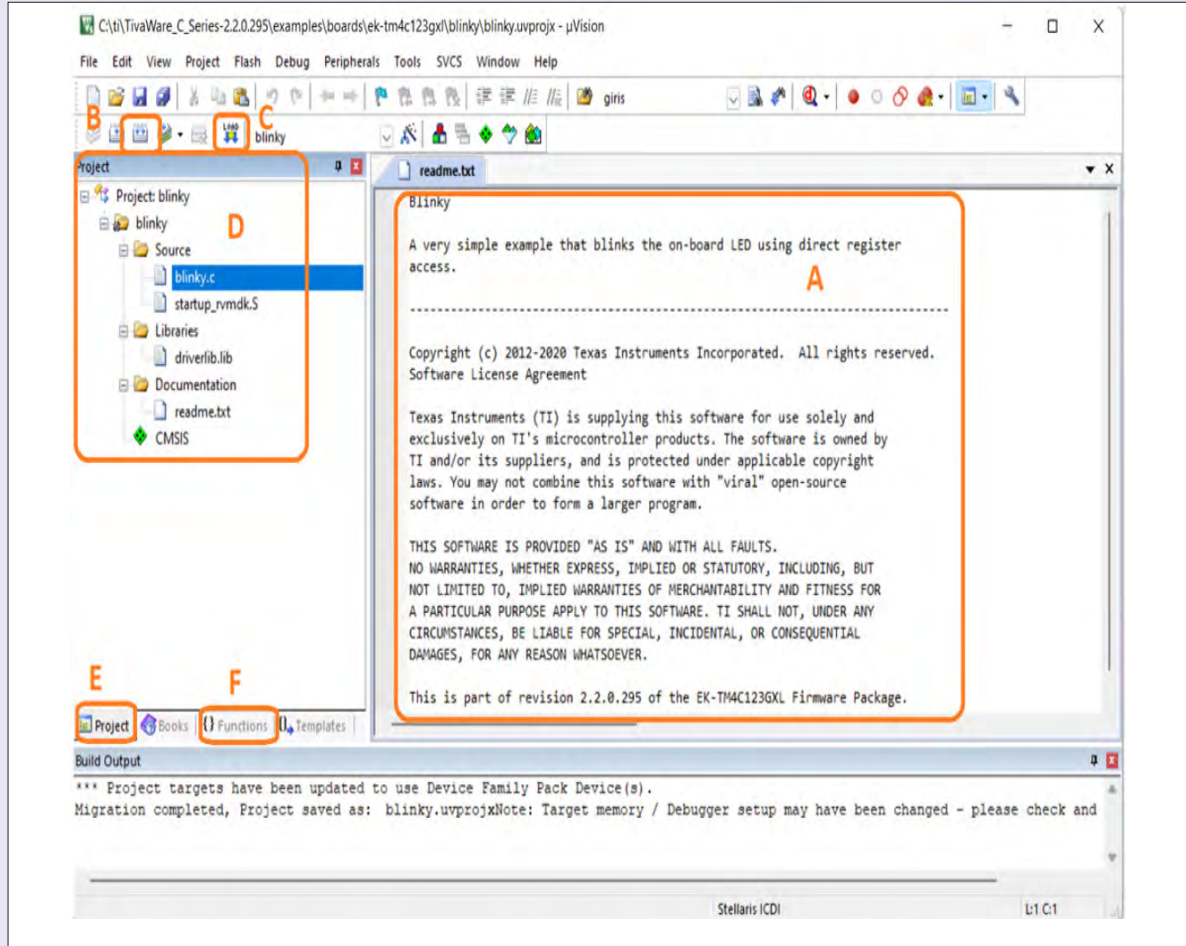
The Device Support for the used devices is found in the following Device Family Pack(s):

Keil:TM4C_DFP:1.1.0

Install Cancel

Görsel C.11: Örnek dosyanın ARMKEIL yazılımda çalıştırılması

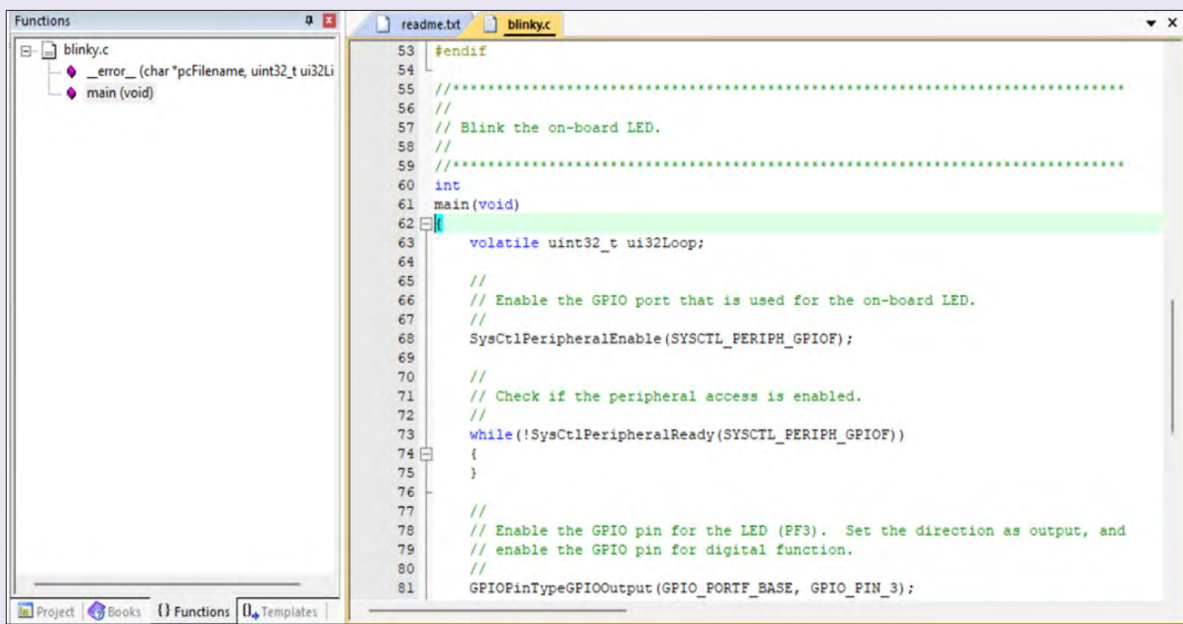
blinky.uvproj dosyası programda açıldığında ilk olarak yeni versiyon ile uyumlu olup olmadığı kontrol eden bir pencere ekrana gelir. Bu pencerede **Migrate to Device Pack** butonuna basılarak sürücü paketlerinin cihaz paketi içine kurulması istenir. Cihazda bununla ilgili hiçbir dosya bulamadığında yazılım TM4C serisinden bir cihaz tespiti yapar ve paketin yüklenmesi için bir kurulum penceresini açar. “Install” butonuna basılarak bu paket dosyası yüklenir. Kurulum işlemi sonunda versiyon uyumu gerçekleştirilmiş olur (Görsel C.11). ARMKEIL yazılımı artık örnek dosyayı rahatlıkla çalıştırmış olur. Görsel C.12’de yazılımın ana ekran görünümü, Görsel C.13’te dosya içindeki fonksiyonların listesi ve Tablo C.1’de ise bu ekrandaki kısımların özellikleri görülmektedir.



Görsel C.12: Örnek dosyanın ARMKEIL yazılımda açılması ve ana ekranda verilerin gösterilmesi

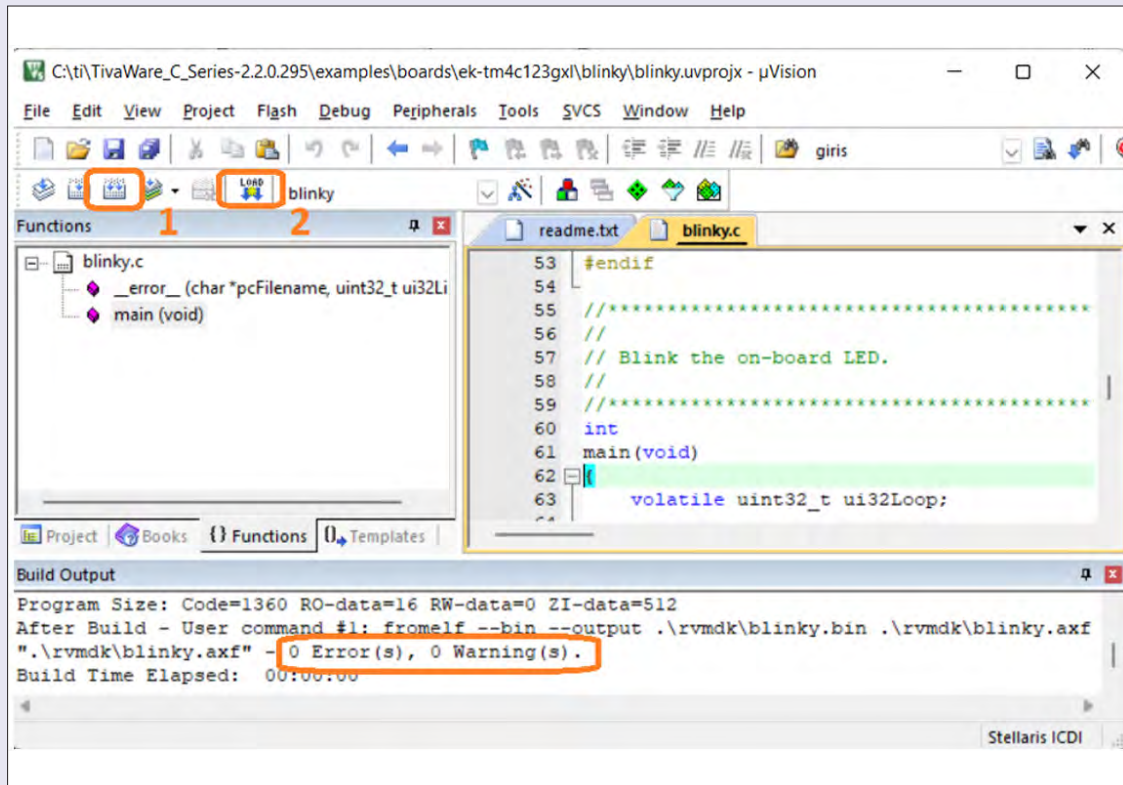
Tablo C.1: Devrelerde Kullanılan Malzemeler

Harf	Özellikler
A	Kod yazma editörü
B	Yazılan projenin derleme (compile-derle-compayl) butonu
C	Derlenen projenin mikrodnetleyiciye yüklenmesini (download-davnload) sağlayan buton
D	Projeyi oluşturan klasörlerin ve dosyaların yapısı
E	Projenin klasör ve dosya yapısını gösteren buton
F	Projede açık olan dosya içinde üretilen fonksiyonların listesini gösteren buton



Görsel C.13: ARMKEIL ekranında açık olan dosya içindeki fonksiyonların listesi

Örnek program açıldıktan sonra proje derlenme işlemi  araç çubuğundan yapılır ve **Build Output** kısmından derlemede hata olup olmadığı kontrol edilir. **0 Error (s), 0 Warning (s)** uyarısı görüldüğünde  araç çubuğuna tıklanarak proje dosyası mikroişlemci kartına yüklenir (Görsel C.14). TIVA kontrol kartı üzerinde bulunan **reset** butonuna basılarak yüklenen program çalıştırılır.



Görsel C.14: ARMKEIL ekranından programın derlenmesi ve karta yüklenmesi

C.7. ARMKEIL Yazılımının Kaynak Kodlarının İncelenmesi



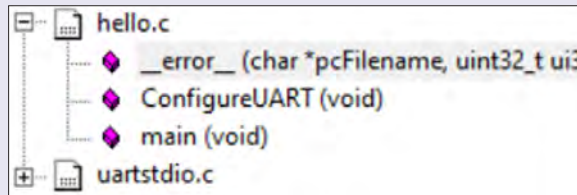
ÖRNEK

Aşağıdaki klasördeki örnek dosya çift tıklanarak ARMKEIL yazılımda çalıştırılır (Görsel C.15).

“C:\ti\TivaWare_C_Series-2.2.0.295\examples\boards\ek-tm4c123gxl\hello\hello.uvproj”

Local Disk (C:) > ti > TivaWare_C_Series-2.2.0.295 > examples > boards > ek-tm4c123gxl > hello >				
Name	Date modified	Type	Size	
ccs	5.09.2020 13:56	File folder		
ewarm	5.09.2020 13:56	File folder		
gcc	5.09.2020 13:56	File folder		
rvmdk	5.09.2020 13:56	File folder		
hello.c	22.11.2021 09:34	C Source	5 KB	
hello.ewd	22.11.2021 09:34	EWD File	17 KB	
hello.ewp	22.11.2021 09:34	EWP File	21 KB	
hello.icf	22.11.2021 09:34	ICF File	3 KB	
hello.ld	22.11.2021 09:34	LD File	2 KB	
hello.sct	22.11.2021 09:34	SCT File	2 KB	
hello.uvopt	22.11.2021 09:34	UVOPT File	9 KB	
hello.uvproj	22.11.2021 09:34	µVision4 Project	16 KB	

Görsel C.15: ARMKEIL yazılımdan hello.uvproj projesi



Görsel C.16: hello.c dosyasına ait fonksiyonlar

Dosya açıldığında bir önceki kısımda anlatılan dosya uyumluluk işlemlerinin aynı şekilde yapılması gerekir. Dosya program ekranında açıldığında Functions sekmesi altında hello.c dosyasının içinde yer alan tüm fonksiyonlar gösterilir (Görsel C.16). Bu dosyada üç adet fonksiyonun olduğu görülür. Her fonksiyon çift tıklanarak imleç, editör penceresinde ilgili kodlara otomatik gider. Dosyanın error fonksiyona ait bölümü Görsel C.17’de verilmiştir. Bu fonksiyonu sürücü kütüphanesi otomatik kullanır. Fonksiyonun kullanıcı ile ilgisi yoktur.

```

// *****
//
// The error routine that is called if the driver library encounters an error.
// *****
#ifdef DEBUG
void
__error__(char *pcFilename, uint32_t ui32Line)
{
}
#endif

```

Görsel C.17: hello.c dosyasına ait error bölümü



Dosyanın ilk çalıştırılacak main fonksiyona ait bölümü Görsel C.18'de verilmiştir. Mikrodenetleyici (MCU) ilk olarak bu kısımdan satır satır çalıştırılır. Seri port-0'ın nasıl ayarlandığını gösteren kod yazılımı Görsel C.19'da verilmiştir.

```
//*****  
// Print "Hello World!" to the UART on the evaluation board.  
//*****  
int main(void)  
{  
    // Enable lazy stacking for interrupt handlers. This allows floating-point  
    // instructions to be used within interrupt handlers, but at the expense of  
    // extra stack usage.  
    MAP_FPU_LazyStackingEnable(); // Ondalikli sayi matematigi aktif ediyor.  
    // Set the clocking to run directly from the crystal. PLL=400/DIV4=50MHz kristal ayarlaniyor.  
    MAP_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);  
    // Enable the GPIO port that is used for the on-board LED.  
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF); // F portu AKTIF ediyor.  
    // Enable the GPIO pins for the LED (PF2 & PF3).  
    MAP_GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_2); // F2 ve F3 pinleri ÇIKIS olara ayarlaniyor.  
    // Initialize the UART. Seri portu ayarlayan fonksiyon çağırılıyor.  
    ConfigureUART();  
    // Hello!  
    UARTprintf("Hello, world!\n"); // Seri porta (bilgisayara) "Hello, world" yazısı gönderiliyor.  
    // We are finished. Hang around doing nothing.  
    while(1) // Küme parantez içindeki kodlar sonsuza kadar çalıştırılıyor.  
    {  
        // Turn on the BLUE LED. Mavi LED yakılıyor.  
        GPIOWrite(GPIO_PORTF_BASE, GPIO_PIN_2, GPIO_PIN_2);  
        // Delay for a bit. Biraz bekliyor.  
        SysCtlDelay(SysCtlClockGet() / 10 / 3);  
        // Turn off the BLUE LED. Mavi LED söndürülüyor.  
        GPIOWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0);  
        // Delay for a bit. Biraz bekliyor  
        SysCtlDelay(SysCtlClockGet() / 10 / 3);  
    }  
}
```

Görsel C.18: hello.c dosyasına ait main bölümü

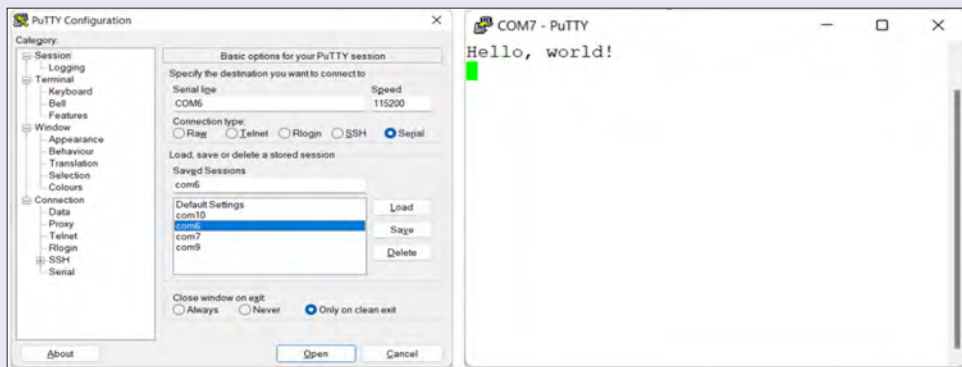
```
void ConfigureUART(void)  
{  
    // Enable the GPIO Peripheral used by the UART.  
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA); // Seri portu kullanan A portu AKTIF ediyor.  
    // Enable UART0  
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0); // Sekiz adet seri porttan SIFIR numaralı seri port seçiliyor.  
    // Configure GPIO Pins for UART mode.  
    MAP_GPIOPinConfigure(GPIO_PA0_U0RX); // A0 pini RX olarak ayarlanıyor.  
    MAP_GPIOPinConfigure(GPIO_PA1_U0TX); // A1 pini TX olarak ayarlanıyor.  
    MAP_GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1); // Seriport pinleri AKTIF ediyor.  
    // Use the internal 16MHz oscillator as the UART clock source.  
    UARTClockSourceSet(UART0_BASE, UART_CLOCK_PIOSC); // Seri port osilatör dahili olarak 16MHz seçiliyor.  
    // Initialize the UART for console I/O.  
    UARTStdioConfig(0, 115200, 16000000); // Seri port 0 16MHz ve 115200 BAUD olarak AKTIF ediyor.  
}
```

Görsel C.19: hello.c dosyasında seri port-0'ın ayarlanması



NOT

Mikrodenetleyicinin bilgisayara gönderdiği bilgiyi görmek için ücretsiz bir yazılım olan **putty** programı <https://www.putty.org/> web sitesinden indirilerek kurulur. Görsel C.20'de putty programının ayarları ve terminal ekran görüntüsü verilmiştir.



Görsel C.20: Putty programının ayarları ve terminal ekran görüntüsü



C.8. ARM32 İLE İLGİLİ
ÖRNEK UYGULAMALAR



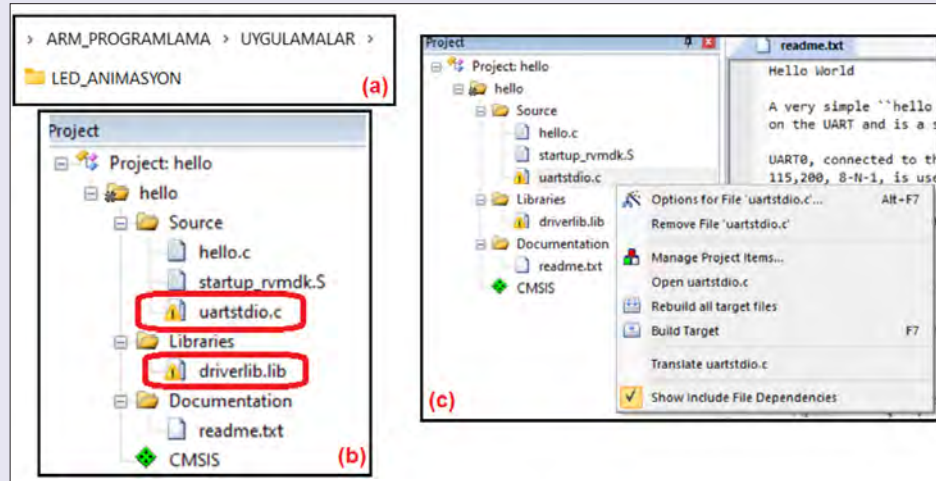
Bu bölümde 11 farklı uygulama ARM32 TIVA kullanılarak gerçekleştirilecektir. Bu uygulamalar aşağıda listelenmiştir.

1. Uygulama:	LED Animasyon
2. Uygulama:	RGB LED
3. Uygulama:	Zamanlayıcı (Timer)
4. Uygulama:	Buton LED
5. Uygulama:	ADC Sensör
6. Uygulama:	PWM
7. Uygulama:	NOKIA 5110 LCD Display
8. Uygulama:	LCD'de Sensör Verilerini Gösterme
9. Uygulama:	LCD'de DHT11 Sıcaklık ve Nem Sensör Bilgilerinin Gösterilmesi
10. Uygulama:	LCD'de Dijital Saat
11. Uygulama:	Kronometre



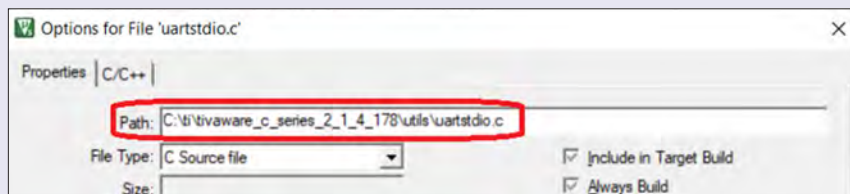
1. Uygulama: LED Animasyon

- TIVAWARE klasöründe yer alan hello adlı klasör, uygulamalar klasörü içine kopyalanır. **Hello** adlı klasör ismi, örnek olarak **led_animasyon** ismi ile değiştirilir ve **hello.uvproj** dosya ismi çift tıklanarak dosya, **ARMKEIL** yazılımında açılır. Pack installer yazılımı ile versiyon güncellemesi gerçekleştirilir (Görsel C.21).



Görsel C.21: Dosya klasörü, dosya yapısında görülen hatalar ve bu hataların seçilmesi

- Görsel C.21'de görülen hatalı dosya, farenin sağ tuşu ile tıklanır. Hatalı dosya olan **uartstdio.c** dosyası sağ fare tuşu ile tıklanınca gelen menüden **Options for File** (dosya için seçenekler-opşins for fayl) komutu seçilir. **Path** (yol) yazı kutusuna **TIVAWARE** klasör yapısı kopyalanır. Görsel C.22'deki **TIVAWARE** klasör yolu kopyalanıp **OK** butonu tıklanınca **uartstdio.c** dosyasında görülen ünlem işareti görülmez.



Görsel C.22: TIVAWARE dosya klasörü adresinin tanımlanması



NOT

LED_ANIMASYON olarak değiştirilen klasör bundan sonraki tüm uygulamalarda proje adına uygun olarak kopyalanır, yeni proje üretilir, klasör ismi proje adına göre değiştirilir ve **ARMKEIL** yazılımı ile açılıp gerekli kod değişikliği yapılarak karta kodlar yüklenir. Bu nedenle öğrenme birimlerindeki uygulama kısımlarında bu işlemler tekrarlanır.



2. Uygulama: RGB LED

Uygulamalar klasöründeki **LED_UYGULAMASI** klasöründen kopya üretilir ve klasör ismi **RGB_LED** olarak değiştirilir. **ARMKEIL** yazılımı ile **hello.uvproj** dosyası çift tıklanarak açılır. TIVA kartının üzerinde F1, F2 ve F3 pinlerine bağlı olan RGB LED'i vardır. Aynı anda bu üç pinden hangisine veya hangilerine çıkış gönderilirse o pindeki RGB ışıkları aktif edilir (Görsel C.26).

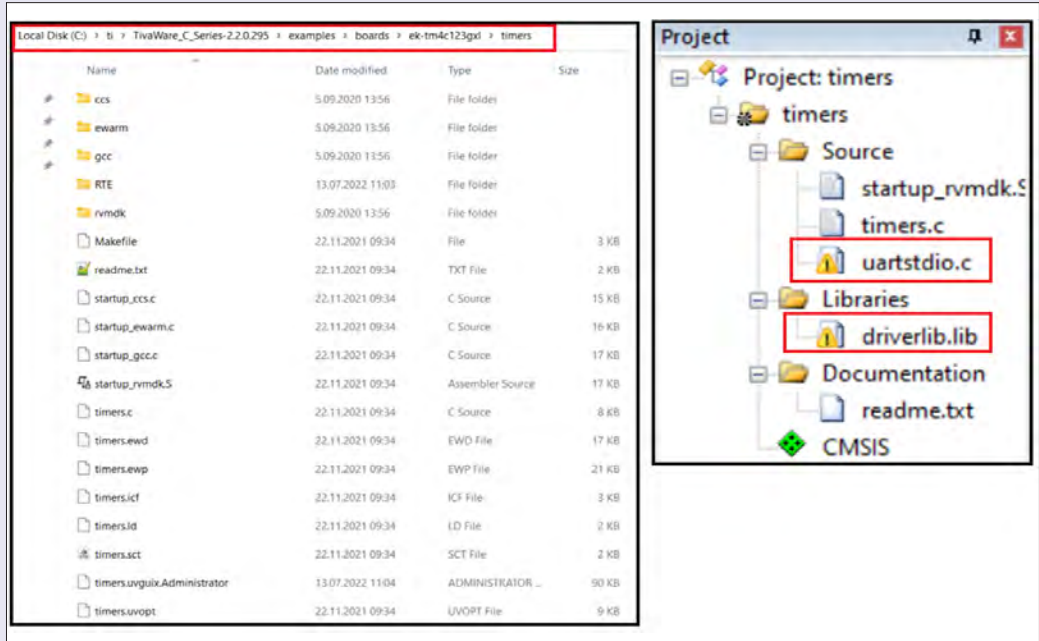
```
int main(void)
{
    ROM_FPU_LazyStackingEnable(); // Ondalıkli sayi matematik islemcisi AKTIF edilir.
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN); // PLL=400MHz / DIV4 = 50MHz ayarlanır.
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF); // F portu seçilerek AKTIF edilir.
    ROM_GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3); // F1, F2 ve F3 pinleri ÇIKIŞ olarak ayarlanır.
    ConfigureUART(); // Seri port 0 ayarlayan fonksiyon çağırılır.
    UARTprintf("RGB LED Uygulaması!\n"); // Seri porttan bilgisayara "RGB LED Uygulaması" yazısı gönderilir.
    while(1) // Kume parantez içlerindeki kodlar sonsuza kadar işlemci (50MHz) hızında tekrar tekrar çalıştırılır.
    {
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, GPIO_PIN_1); // Bekleme olmadan, Kırmızı led yanar.
        SysCtlDelay(SysCtlClockGet() / 10 / 1); // Bir sn bekler.
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, GPIO_PIN_2); // Bekleme olmadan, Mavi led yanar. Kırmızı led yanmaya devam eder.
        SysCtlDelay(SysCtlClockGet() / 10 / 1); // Bir sn bekler.
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0); // Mavi led soner. Kırmızı led yanmaya devam eder.
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_3, GPIO_PIN_3); // Bekleme olmadan, Yeşil led yanar. Kırmızı led yanmaya devam eder.
        SysCtlDelay(SysCtlClockGet() / 10 / 1); // Bir sn bekler.
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_3, 0); // Yeşil led soner.
    }
}
```

Görsel C.26: RGB LED'lerin kontrol edildiği kodlar



3. Uygulama: Zamanlayıcı (Timer)

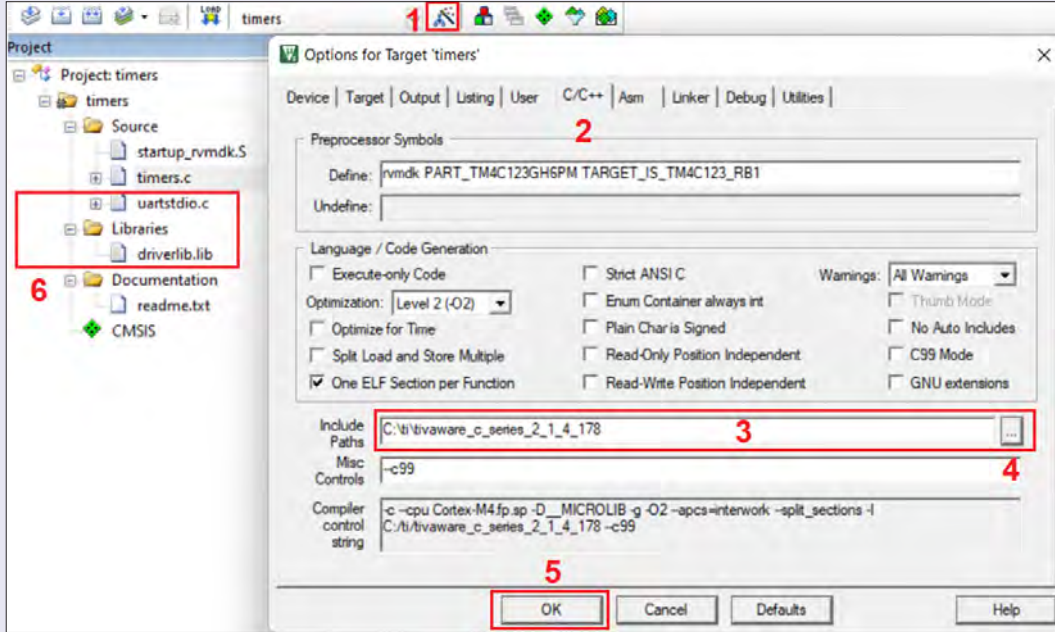
TIVAWARE klasöründen timers klasörü, çalışma klasörü olan **uygulamalar** klasörüne kopyalanır. **timers** klasör ismi, **BUTON_LED_KONTROL** ismi ile değiştirilir ve **timers.uvproj** dosya ismi çift tıklanarak dosya **ARMKEIL** yazılımında açılır. Pack installer yazılımında **Migrate to Device Pack** ile versiyon güncellemesi gerçekleştirilir (Görsel C.27).



Görsel C.27: TIVAWARE klasöründeki timers uygulama klasörü ve bu projedeki hatalı yollar

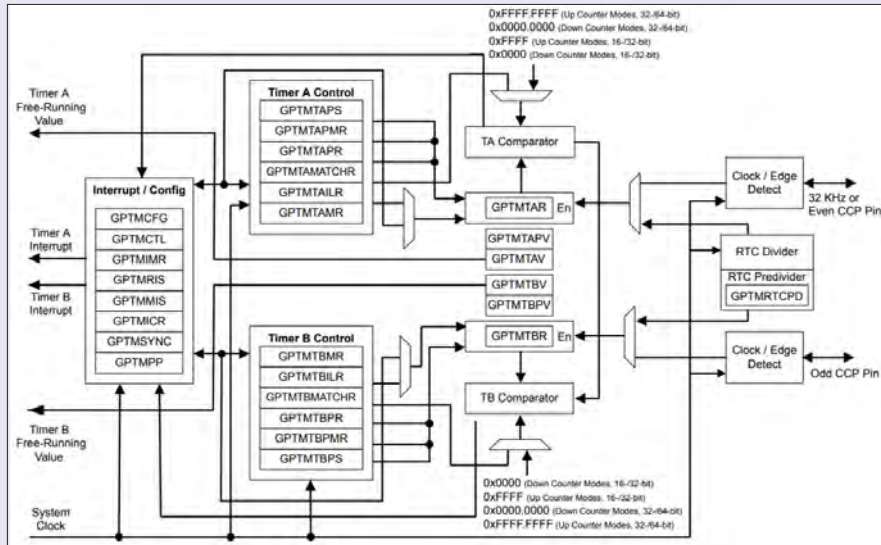


Daha önce görülen dosya yolu hata düzeltme işlemi timers projesine de uygulanarak çözülür (Görsel C.28).



Görsel C.28: TIVAWARE klasöründeki timers projesindeki hatalı yolların düzeltilmesi

Timer kavramı kısaca, belli zaman aralıklarında sürekli istenen kodları çalıştıran anlamına gelir. Ana (main) fonksiyondan çağrılmaz, otomatik ve bağımsız çalışır. TIVA **TM4C123GH6PM** entegresinde altı adet 16 veya 32 bitlik blok ile yine altı adet 32 veya 64 bitlik blok vardır. Her iki altı adetlik bloкта da 32,768 kHz'lik real time clock (gerçek zaman saati) sayacını bünyesinde barındırır. Bu zaman saati, 32/64 bitlik olan bloкта 64 bit olarak çalışır. Her iki timer bloğunda; bir kere çalışabilecek, periyodik çalışabilecek veya PWM sinyali olabilecek şekilde de yazılımla ayarlanarak çalışır. Yukarı veya aşağı sayar. İki adet bloktaki PWM'ler ayrı ayrı ayarlanarak 12 adet PWM kullanılabilir. ADC analog sinyalleri tetikleme olaylarında kullanılır. Timer tetiklemeleri kesmelerle (interrupt) üretilir. Görsel C.29'da GPTM modülünün timer blok diyagramı görülmektedir.



Görsel C.29: TIVA entegresine ait timer blok diyagramı



Hangi pinlerin Timer pinleri olduğu bilgisi, 706-707'nci sayfalarda [veri sayfası (Data Sheet)] bulunur. Ek 1'deki (tm4c123gh6pm_DATASHEET.pdf) sayfa 704'te timerlar hakkında daha fazla teknik bilgi bulunmaktadır (Görsel C.29). Firmanın sayfasından ilgili dosyaya ulaşılabilir.

Görsel C.30'da seri porta bilgi gönderme ayarlarına sahip C kodları görülmektedir.

```
void ConfigureUART(void)
{
    // Enable the GPIO Peripheral used by the UART.
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
    // Enable UART0
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
    // Configure GPIO Pins for UART mode.
    MAP_GPIOPinConfigure(GPIO_PA0_U0RX);
    MAP_GPIOPinConfigure(GPIO_PA1_U0TX);
    MAP_GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);
    // Use the internal 16MHz oscillator as the UART clock source.
    UARTClockSourceSet(UART0_BASE, UART_CLOCK_PIOSC);
    // Initialize the UART for console I/O.
    UARTStdioConfig(0, 115200, 16000000);
}
```

Görsel C.30: Seri porta bilgi gönderme ayarları

Görsel C.31'de timers projesinin C kodları görülmektedir.

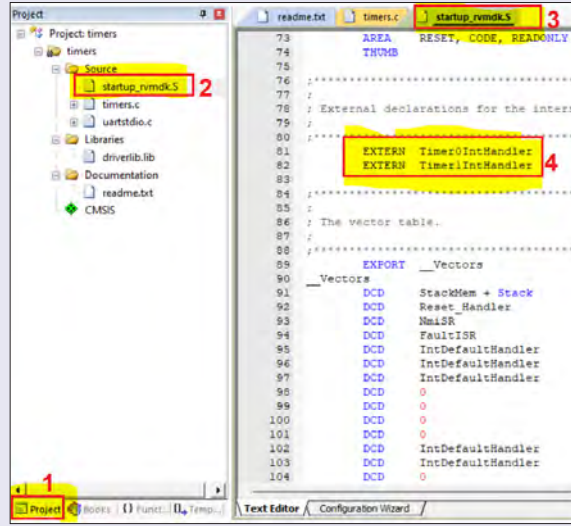
```
int main(void)
{
    // Enable lazy stacking for interrupt handlers. This allows floating-point
    // instructions to be used within interrupt handlers, but at the expense of
    // extra stack usage.
    MAP_FFULazyStackingEnable(); // Ondalıkli sayı matematik işlemcisini aktif et.
    // Set the clocking to run directly from the crystal.
    // PLL kullanmadan, direk osilatör ile 16MHz de çalıştır.
    MAP_SysCtlClockSet(SYSCTL_SYSDIV_1 | SYSCTL_USE_OSC | SYSCTL_OSC_MAIN | SYSCTL_XTAL_16MHZ);
    // Initialize the UART and write status.
    ConfigureUART(); // Seri portu ayarlayan fonksiyonu çağır.
    UARTprintf("\033[2JTIMER Örnek\n"); // Seri porta "TIMER Örnek" yazısını gönder.
    // \033[2J Karakterleri terminal ekranını temizle ve her zaman ilk satıra yaz
    UARTprintf("T1: 0 T2: 0"); // İki adet zamanlayıcının ilk verisi 0'dır.
    // Enable the GPIO port that is used for the on-board LED.
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF); // F portunu AKTIF et.
    // Enable the GPIO pins for the LED (PF1 & PF2).
    MAP_GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_2 | GPIO_PIN_1); // F1 ve F2 pinleri ÇIKIS olarak ayarla.
    // Enable the peripherals used by this example.
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0); // TIMER 0'ı kullanmak için AKTIF et.
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER1); // TIMER 1'i kullanmak için AKTIF et.
    // Enable processor interrupts.
    MAP_IntMasterEnable(); // Ana kesmeleri AKTIF et.
    // Configure the two 32-bit periodic timers.
    MAP_TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC); // Timer 0'i periyodik olarak ayarla.
    MAP_TimerConfigure(TIMER1_BASE, TIMER_CFG_PERIODIC); // Timer 1'i periyodik olarak ayarla.
    MAP_TimerLoadSet(TIMER0_BASE, TIMER_A, MAP_SysCtlClockGet()); // Timer0'i periyot süresini MCU saatine esitle (1sn)
    MAP_TimerLoadSet(TIMER1_BASE, TIMER_A, MAP_SysCtlClockGet() / 2); // Timer0'i periyot süresini MCU saatinin 2 kat hızına esitle (0.5sn)
    // Setup the interrupts for the timer timeouts.
    MAP_IntEnable(INT_TIMER0A); // Timer0'in kesmesini AKTIF et.
    MAP_IntEnable(INT_TIMER1A); // Timer1'in kesmesini AKTIF et.
    MAP_TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT); // Timer0 için Zaman gecikmesini AKTIF et.
    MAP_TimerIntEnable(TIMER1_BASE, TIMER_TIMA_TIMEOUT); // Timer0 için Zaman gecikmesini AKTIF et.
    // Enable the timers.
    MAP_TimerEnable(TIMER0_BASE, TIMER_A); // Timer0'ı çalıştır.
    MAP_TimerEnable(TIMER1_BASE, TIMER_A); // Timer1'i çalıştır.

    // Loop forever while the timers run.
    while(1) // Sonsuz döngü içinde çalışan kod yoktur. Tüm kodlar Timer0 ve Timer1 fonksiyonları içinde çalışırlar.
    {
    }
}
```

Görsel C.31: Timers projesinin ana fonksiyonunda çalışan kodlar

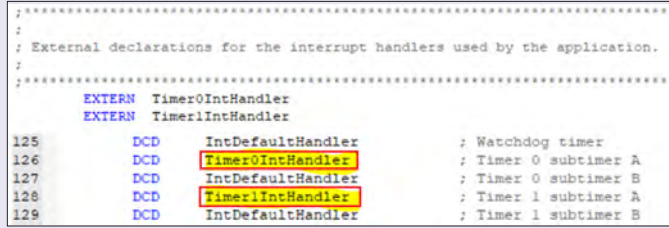


Görsel C.32’de timers projesindeki başlangıçta otomatik çalıştıran dosya olan **startup_rvmdk.s** dosyasının yapısı görülmektedir.



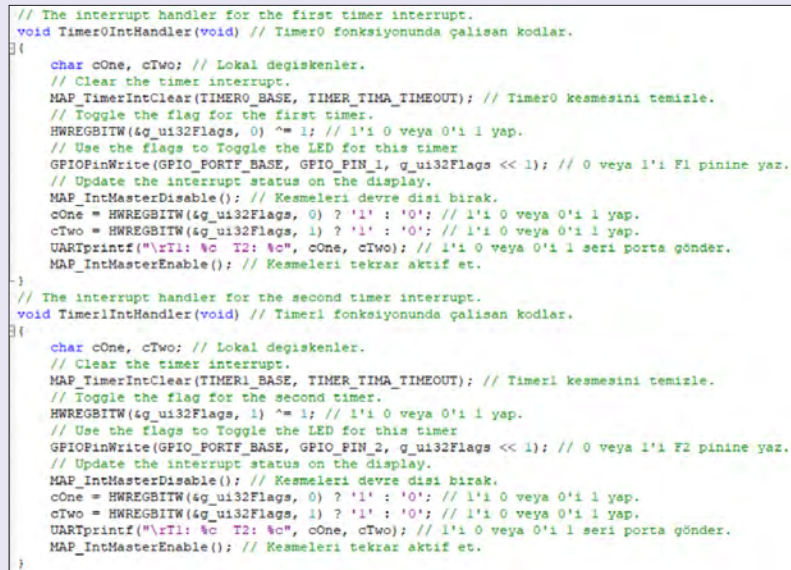
Görsel C.32: Timers projesindeki başlangıçta otomatik çalıştıran dosya olan **startup_rvmdk.s** dosyası

Görsel C.33’te timers projesinde yer alan **startup_rvmdk.s** dosyasındaki haricî fonksiyonların nasıl çalıştırılacağını gösteren C kodları görülmektedir.



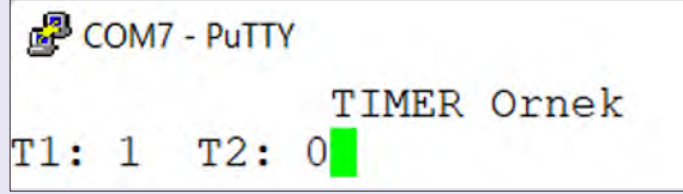
Görsel C.33: Timers projesindeki startup_rvmdk.s dosyasındaki haricî fonksiyonların nasıl çalıştırılacağını gösteren kodlar

Görsel C.34’te timers projesindeki Timer0 ve Timer1 fonksiyonlarında yer alan C kodları görülmektedir.



Görsel C.34: Timers projesindeki Timer0 ve Timer1 fonksiyonlarındaki kodlar

Görsel C.35'te timers projesinin derlenmesi ve işlem sonunda karta yüklenmesi sonucunda **putty** yazılımındaki çalışması görülmektedir.



Görsel C.35: Timers projesi derlenip yüklendikten sonra putty programındaki ekran görüntüsü



4. Uygulama: Buton LED

- Uygulamalar klasöründeki **TIMER_UYGULAMASI** klasöründen kopya üretilir ve klasör ismi **BUTON_LED_UYGULAMASI** olarak değiştirilir **ARMKEIL** yazılımı ile **timers.uvproj** dosyası çift tıklanarak açılır.
- Görsel C.36-C.37'deki kullanıcı butonu 1'e basıldığında F1 LED'i yanacak, kullanıcı butonu 2'ye basıldığında ise F1 LED'i sönecek şekilde bir uygulama yapılacaktır.

```
// F0 ve F4 pinlerine bagli kullanıcı butonlarına degisken isimleri ataniyor.
#define SOL_BUTON        GPIO_PIN_0
#define SAG_BUTON        GPIO_PIN_4
#define BUTONLAR          (SOL_BUTON | SAG_BUTON)
static unsigned char BUTON_DURUMLARI = BUTONLAR;
```

Görsel C.36: Kullanıcı butonlarının değişken isimlerine atama işlemlerinin kodları

```
MAP_TimerLoadSet(TIMER1_BASE, TIMER_A, MAP_SysCtlClockGet() / 100); // Timer0'i periyot süresini MCU saatinin 100 kat hızına esitle (0.01sn)
// Setup the interrupts for the timer timeouts.
MAP_IntEnable(INT_TIMER0A); // Timer0'in kesmesini AKTIF et.
MAP_IntEnable(INT_TIMER1A); // Timer1'in kesmesini AKTIF et.
MAP_TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT); // Timer0 için Zaman gecikmesini AKTIF et.
MAP_TimerIntEnable(TIMER1_BASE, TIMER_TIMA_TIMEOUT); // Timer1 için Zaman gecikmesini AKTIF et.
// Enable the timers.
MAP_TimerEnable(TIMER0_BASE, TIMER_A); // Timer0'i çalıştır.
MAP_TimerEnable(TIMER1_BASE, TIMER_A); // Timer1'i çalıştır.
HWREG(GPIO_PORTF_BASE + GPIO_O_LOCK) = GPIO_LOCK_KEY;
HWREG(GPIO_PORTF_BASE + GPIO_O_CR) |= 0x01;
HWREG(GPIO_PORTF_BASE + GPIO_O_LOCK) = 0;
ROM_GPIODirModeSet(GPIO_PORTF_BASE, BUTONLAR, GPIO_DIR_MODE_IN); // BUTONLAR giriş için SET edildi.
// BUTONLAR 2mA akım çekecek şekilde ayarlandı.
ROM_GPIOPadConfigSet(GPIO_PORTF_BASE, BUTONLAR, GPIO_STRENGTH_2MA, GPIO_PIN_TYPE_STD_WFU);
```

Görsel C.37: Timer1 periyot süresinin saniyenin 100 katı hızla tekrarlanacağı şekilde ve butonların giriş olarak ayarlandığı kodlar

Timer1 zamanlayıcısı ile saniyenin 100 katı hızla tekrarlama gerçekleştirilir. Bu süre içinde mikrodnetleyici, kullanıcı tarafından butona basılıp basılmadığını kontrol eder. Kullanıcı tarafından butona basıldığı anda **BUTONLAR** isimli değişkene buton durumlarının değerleri atanır. Bu işlemler Timer1 fonksiyonu içinde gerçekleştirilir (Görsel C.38-C.39).

```
//*****
//
// The interrupt handler for the first timer interrupt.
void Timer0IntHandler(void) // Timer0 fonksiyonunda çalışan kodlar.
{
    return; // RETURN komutu ile, asagidaki kodlar devre disi birakiliyor.
    char cOne, cTwo; // Lokal degiskenler.
    // Clear the timer interrupt.
    MAP_TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT); // Timer0 kesmesini temizle.
    // Toggle the flag for the first timer.
    HWREGBITW(&g_ui32Flags, 0) ^= 1; // 1'i 0 veya 0'i 1 yap.
    // Use the flags to Toggle the LED for this timer
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, g_ui32Flags << 1); // 0 veya 1'i F1 pinine yaz.
    // Update the interrupt status on the display.
    MAP_IntMasterDisable(); // Kesmeleri devre disi birak.
    cOne = HWREGBITW(&g_ui32Flags, 0) ? '1' : '0'; // 1'i 0 veya 0'i 1 yap.
    cTwo = HWREGBITW(&g_ui32Flags, 1) ? '1' : '0'; // 1'i 0 veya 0'i 1 yap.
    UARTprintf("\rT1: %c T2: %c", cOne, cTwo); // 1'i 0 veya 0'i 1 seri porta gönder.
    MAP_IntMasterEnable(); // Kesmeleri tekrar aktif et.
}
```

Görsel C.38: Timer0 fonksiyonu içindeki kodların devre dışı bırakılmasını gösteren kodlar



```
// The interrupt handler for the second timer interrupt.
void Timer1IntHandler(void) // Timer1 fonksiyonunda çalışan kodlar.
{
    MAP_TimerIntClear(TIMER1_BASE, TIMER_TIMA_TIMEOUT); // Timer1 kesmesini temizle.
    MAP_IntMasterDisable(); // Kesmeleri devre dışı bırak.
    BUTON_DURUMLARI = GPIOPinRead(GPIO_PORTF_BASE, BUTONLAR);
    if(BUTON_DURUMLARI == SOL_BUTON) {
        UARTprintf("\r(F4) SOL BUTONA BASILDI");
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, GPIO_PIN_2);
        // Butona basma tekrarlanmaması için biraz bekle.
        SysCtlDelay((SysCtlClockGet() * 0.3) / 3);
    } else if(BUTON_DURUMLARI == SAG_BUTON) {
        UARTprintf("\r(F0) SAG BUTONA BASILDI");
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0);
        // Butona basma tekrarlanmaması için biraz bekle.
        SysCtlDelay((SysCtlClockGet() * 0.3) / 3);
    }
    //UARTprintf("\r(F0) STEP MOTOR DURDURULDU");
    MAP_IntMasterEnable(); // Kesmeleri tekrar aktif et.
}
```

Görsel C.39: Timer1 fonksiyonu içinde butona basılarak LED'i yakma ve söndürme işlemi



5. Uygulama: ADC Sensör

Uygulamalar klasöründeki **BUTON_LED_UYGULAMASI** klasöründen kopya üretilir ve klasör ismi **ADC_SENSOR_UYGULAMASI** olarak değiştirilir. ARMKEIL yazılımı ile timers. uvproj dosyası çift tıklanarak açılır.

```
unsigned long ADC0DATA[2]; // İki adet sensör bilgisi için degisken.
int sensor1, sensor2; // İki adet sensör bilgisinin atanacağı degiikenler..
```

Görsel C.40: ADC okuma işlemleri için tanımlanan değişkenler

- ADC (analog-dijital çevirici), sensör okuma işlemleri için kullanılır. ADC'den okuma işlemleri için tanımlanan değişken kodlar Görsel C.40'da görülmektedir. B4 ve B5 pinlerine 3,2 Volt'a kadar analog çıkışı olan iki adet herhangi sensor bağlanır. Seri porta okunan sensor1 ve sensor2 ADC değerleri gönderilir.
- Main fonksiyonu içinde Görsel C.41'deki kodlar yazılarak ADC ayarlanır. Görsel C.42'deki kodlarla ise Timer0 fonksiyonu içinde sürekli olarak ADC okuma işlemi yapılır ve sonuçlar seri porta gönderilir.

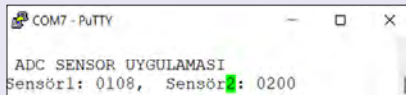
```
// ADC AYARLA
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);
GPIOPinTypeADC(GPIO_PORTB_BASE, GPIO_PIN_4 | GPIO_PIN_5);
ADCSequenceConfigure(ADC0_BASE, 0, ADC_TRIGGER_PROCESSOR, 0);
ADCSequenceStepConfigure(ADC0_BASE, 0, 0, ADC_CTL_CH10);
ADCSequenceStepConfigure(ADC0_BASE, 0, 1, ADC_CTL_CH11 | ADC_CTL_END | ADC_CTL_IE );
ADCSequenceEnable(ADC0_BASE, 0);
ADCIntClear(ADC0_BASE, 0);
```

Görsel C.41: main fonksiyon içerisinde ADC işlemi için B4 ve B5 pinlerinin ayarlanması için kodlar

```
void Timer0IntHandler(void)
{
    ADCProcessorTrigger(ADC0_BASE, 0);
    while(!ADCIntStatus(ADC0_BASE, 0, false)) { }
    ADCIntClear(ADC0_BASE, 0);
    ADCSequenceDataGet(ADC0_BASE, 0, &ADC0DATA[0]);
    sensor1 = ADC0DATA[0]; sensor2 = ADC0DATA[1];
    UARTprintf("\rSensör1: %04d, Sensör2: %04d", sensor1, sensor2);
}
```

Görsel C.42: Timer0 fonksiyonu içinde sürekli olarak ADC okuma işlemi yapan ve sonuçları seri porta gönderen kodlar

- Görsel C.43'te seri porta gönderilen ADC sensör verilerinin putty programındaki görüntüsü görülmektedir.



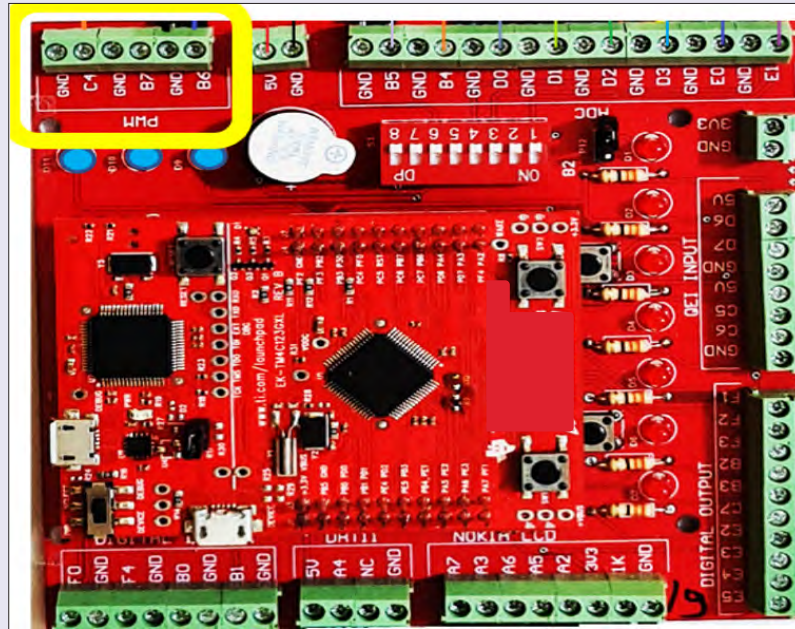
Görsel C.43: Seri porta gönderilen ADC sensör verilerinin putty programındaki görüntüsü

6. Uygulama: PWM

- Uygulamalar klasöründeki **ADC_SENSOR_UYGULAMASI** klasöründen kopya üretilir ve klasör ismi **PWM_UYGULAMASI** olarak değiştirilir. ARMKEIL yazılımı ile **timers.uvproj** dosyası çift tıklanarak açılır. PWM uygulamasının main fonksiyonun içinde PWM'nin pinlerini, jeneratörlerini, periyotlarını ve sayısal değerlerini ayarlayan kodlar bulunmaktadır (Görsel C.44).

```
//***** PWM AYARLA *****
// PWM osilatör sistem osilatörüne göre ayarlanıyor
MAP_SysCtlPWMClockSet(SYSCTL_PWMDIV_1);
// Çevre birimleri AKTİF ediliyor.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);
//M0 ve M1 iki modul LED pinleri için ayarlanıyor.
MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM0);
MAP_GPIOPinConfigure(GPIO_PB6_M0PWM0); // 1.PWM B6 pinde tanımlandı.
MAP_GPIOPinConfigure(GPIO_PB7_M0PWM1); // 2.PWM B7 pinde tanımlandı.
MAP_GPIOPinConfigure(GPIO_PC4_M0PWM6); // 3.PWM C4 pinde tanımlandı.
MAP_GPIOPinTypePWM(GPIO_PORTB_BASE, GPIO_PIN_6 | GPIO_PIN_7); // B6 ve B7 PWM tipinde.
MAP_GPIOPinTypePWM(GPIO_PORTC_BASE, GPIO_PIN_4); // C4 PWM tipinde.
// PWM seçenekleri ayarlanıyor. GEN0 ve GEN3 iki jeneratör AKTİF edildi.
MAP_PWMGenConfigure(PWM0_BASE, PWM_GEN_0, PWM_GEN_MODE_DOWN | PWM_GEN_MODE_NO_SYNC);
MAP_PWMGenConfigure(PWM0_BASE, PWM_GEN_3, PWM_GEN_MODE_DOWN | PWM_GEN_MODE_NO_SYNC);
// GEN0 için Peryot maksimum 10000 ayarlandı.
MAP_PWMGenPeriodSet(PWM0_BASE, PWM_GEN_0, 10000); // Peryot %100 = 10000 demektir.
// GEN3 için Peryot maksimum 10000 ayarlandı.
MAP_PWMGenPeriodSet(PWM0_BASE, PWM_GEN_3, 10000); // Peryot %100 = 10000 demektir.
// PWM DUTY yüzde karşılıkları ayarlandı.
MAP_PWMPulseWidthSet(PWM0_BASE, PWM_OUT_0, 10000); // %100 PWM
MAP_PWMPulseWidthSet(PWM0_BASE, PWM_OUT_1, 5000); // %50 PWM
MAP_PWMPulseWidthSet(PWM0_BASE, PWM_OUT_6, 500); // %10 PWM
// GEN0 ve GEN1 AKTİF edildi.
MAP_PWMGenEnable(PWM0_BASE, PWM_GEN_0);
MAP_PWMGenEnable(PWM0_BASE, PWM_GEN_3);
// PWM çıkış PIN lerİ AÇILDI.
MAP_PWMOutputState(PWM0_BASE, PWM_OUT_0_BIT | PWM_OUT_1_BIT | PWM_OUT_6_BIT, true);
//*****
```

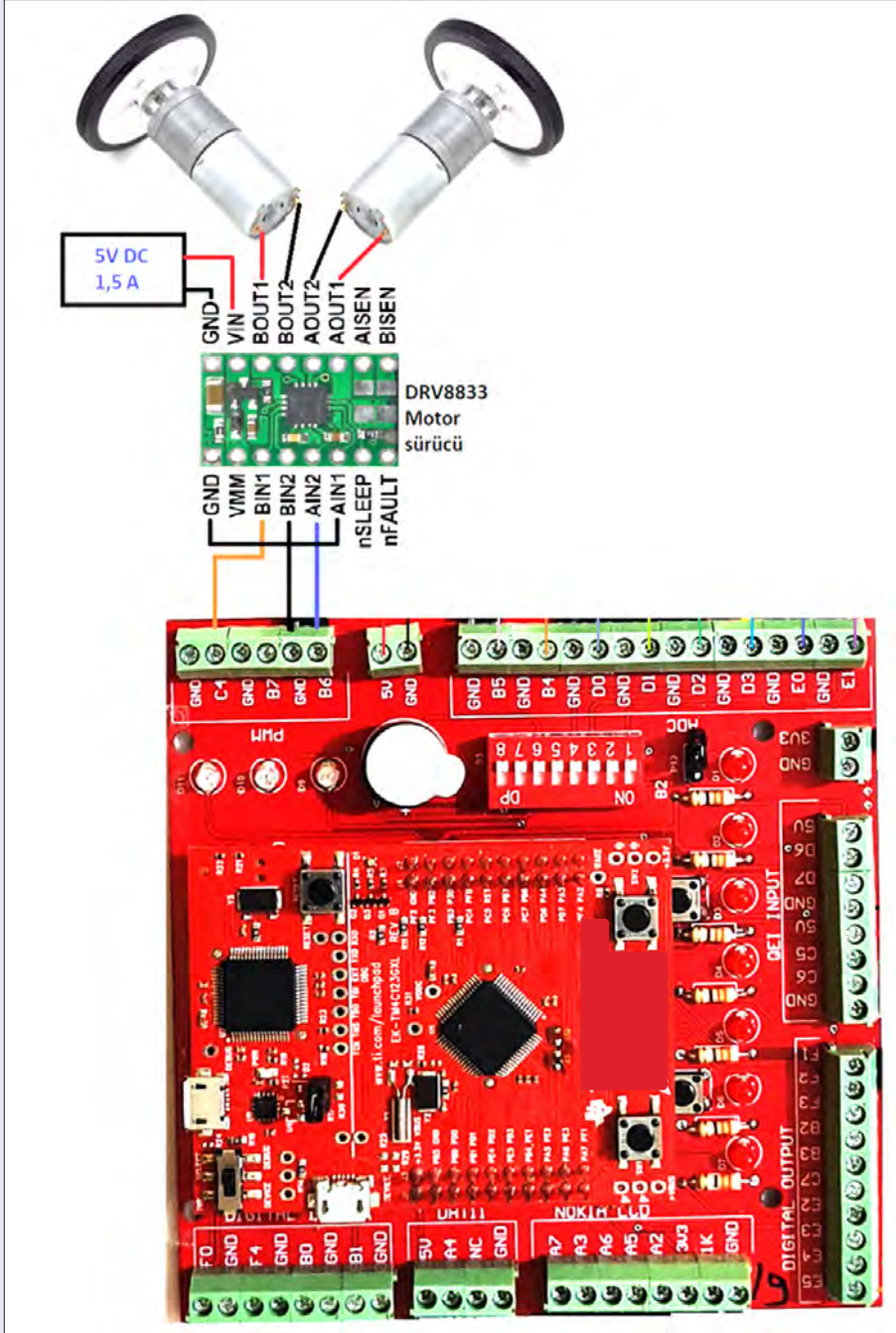
Görsel C.44: PWM işlemlerini ayarlayan kodlar



Görsel C.45: PWM pinlerinin TIVA kartında gösterimi

- Görsel C.45'te PWM pinlerinin TIVA kartındaki gösterimi görülmektedir. B6, B7 ve C4 pinlerine birer LED bağlanır. PWM projesi derlenip mikrodnetleyiciye yüklenince B6, B7 ve C4 pinlerine bağlı LED'ler de sırasıyla tam parlak, yarım parlak ve yüzde on parlak olacak şekilde yanar.

- LED bağlantısı haricinde DC motor da PWM uçlarına bağlanabilir. Görsel C.46'da DC motor sürücüsü ve bu sürücüye bağlı DC motorların TIVA kartına bağlantısı görülmektedir.



Görsel C.46: PWM pinlerine DC motor sürücüsü ve DC motor bağlantı şeması



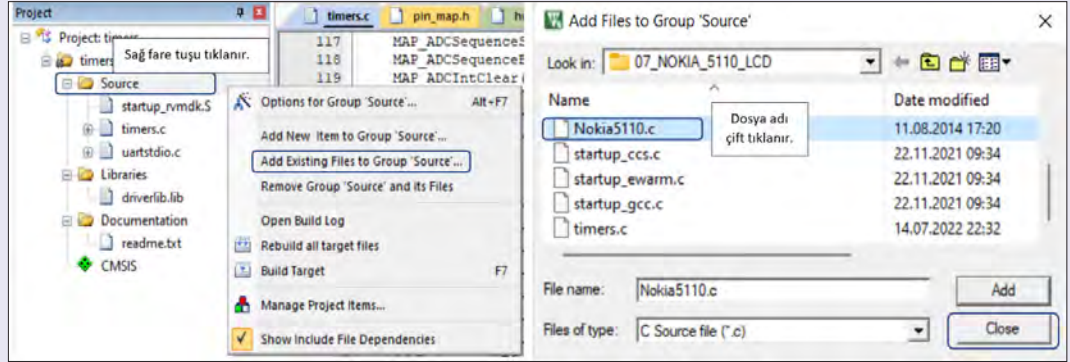
NOT

Görsel C.44'te verilen kodlar ışığında Görsel C.46'daki devre bağlantısına uygun kodlar oluşturunuz, yazılımı derleyip mikrodeneleyici kart üzerinde çalıştırınız.



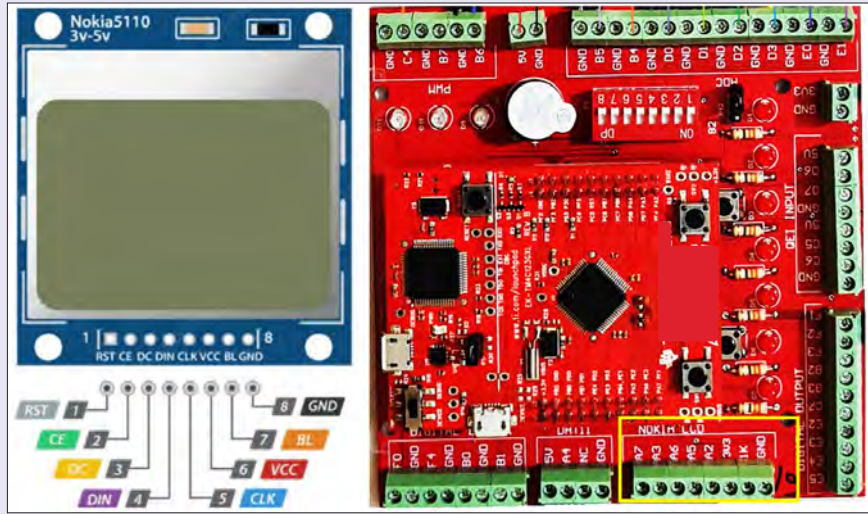
7. Uygulama: Nokia 5110 LCD Display

- Uygulamalar klasöründeki **PWM_UYGULAMASI** klasöründen kopya üretilir ve klasör ismi **NOKIA_5110_LCD** olarak değiştirilir. ARMKEIL yazılımı ile **timers.uvproj** dosyası çift tıklanarak açılır. Uygulama NOKIA 5110 cep telefonlarında da kullanılan LCD displayin kütüphanesi projeye dâhil edilir (Görsel C.47).



Görsel C.47: NOKIA 5110 LCD kütüphanesinin projeye dâhil edilmesi

- Görsel C.48’de NOKIA 5110 LCD ile TIVA kartının PIN bağlantı şemaları ve Tablo C.2’de bu pinlerin özellikleri görülmektedir.



Görsel C.48: Nokia 5110 LCD ile TIVA kontrol kartının PIN bağlantı şeması

Tablo C.2: NOKIA 5110 LCD Pin Bağlantıları

Pin No	Özellği	TIVA Pin
1	Reset (RST)	A7
2	SSIOFss (CE)	A3
3	Data/Command (DC)	A6
4	SSIOtx (Din)	A5
5	SSIOClk (Clk)	A2
6	3.3V (Vcc)	3,3 V
7	Back Light (BL)	1k 3,3V
8	Ground (Gnd)	GND



- Görsel C.49'da NOKIA 5110 kütüphanesinin ve header dosyasının projeye dâhil edilmesini gösteren kodlar görülmektedir.

```
22 #include "Nokia5110.h"
23 #include "inc/hw_nvic.h"
```

Görsel C.49: NOKIA 5110 kütüphanesinin ve header dosyasının projeye dâhil edilmesi

- Tablo C.3'te mavi ve kırmızı Nokia 5110 LCD pinlerinin uçlarının özellikleri görülmektedir.

Tablo C.3: Mavi ve Kırmızı Nokia 5110 LCD Pinleri

Pin	Mavi Nokia 5110 LCD			Pin	Kırmızı Nokia 5110 LCD		
-	Signal	5110	LaunchPad pin	-	Signal	5110	LaunchPad pin
1	Reset	RST	Connected to PA7	1	3.3V	Vcc	Power
2	SSIOFss	CE	Connected to PA3	2	Ground	Gnd	Ground
3	Data/ Command	DC	Connected to PA6	3	SSIOFss	CE	Connected to PA3
4	SSIOTx	Din	Connected to PA5	4	Reset	RST	Connected to PA7
5	SSIOClk	Clk	Connected to PA2	5	Data/ Command	DC	Connected to PA6
6	3.3V	Vcc	Power	6	SSIOTx	Din	Connected to PA5
7	Back Light	BL	Not Connected	7	SSIOClk	Clk	Connected to PA2
8	Ground	Gnd	Ground	8	Back Light	BL	Not Connected

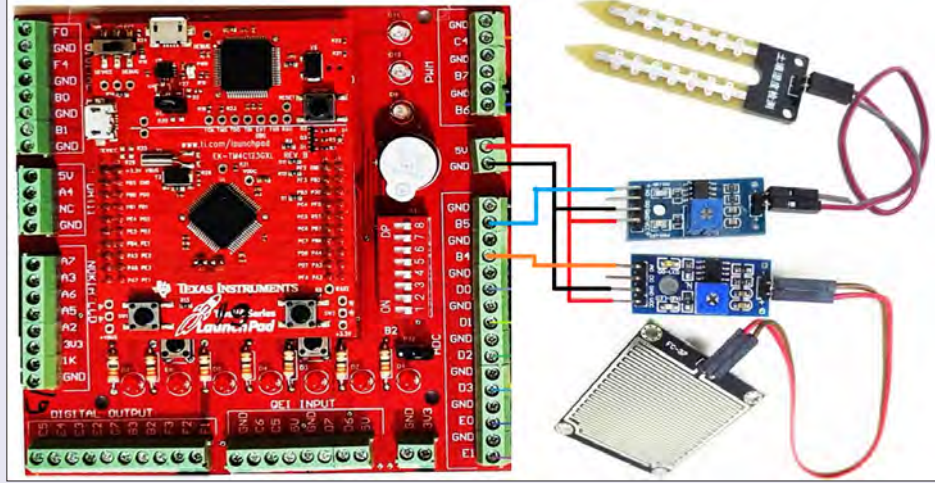
- Görsel C.50'de main fonksiyonu içindeki NOKIA 5110 LCD displayin ayarının yapıldığı ve NOKIA 5110 LCD'ye yazı yazdırma işleminin yapıldığı kodlar görülmektedir.

```
// NOKIA LCD DISPLAY AYARLA
Output_Init(); // NOKIA 5110 DISPLAY ÇIKIS cihaz olarak ayarlandı.
Output_Clear(); // Displayi temizle.
Nokia5110_SetCursor(0,0); Nokia5110_OutString(" MESLEKI "); // 1.sütun 1.satıra "MESLEKI" bilgisini yazdır.
Nokia5110_SetCursor(0,1); Nokia5110_OutString(" TEKNİK "); // 1.sütun 2.satıra "TEKNİK" bilgisini yazdır.
Nokia5110_SetCursor(0,2); Nokia5110_OutString(" EGİTİM "); // 1.sütun 3.satıra "EGİTİM" bilgisini yazdır.
```

Görsel C.50: Main fonksiyonunun içinde NOKIA 5110 LCD display ayarları ve temel yazdırma işlemleri

8. Uygulama: LCD'de Sensör Verilerini Gösterme

- Uygulamalar klasöründeki **NOKIA_5110_LCD** klasöründen kopya üretilir ve klasör ismi **LCD_SENSOR_DISPLAY** olarak değiştirilir. ARMKEIL yazılımı ile **timers.uvproj** dosyası çift tıklanarak açılır. Bu uygulamada LCD displayde iki adet sensör bilgisi gösterilir.
- Global değişken, **unsigned char NOKIA_LCD_BILGI[20];** olarak tanımlanır. Nokia 5110 LCD displayin bağlantı şeması Görsel C.51'de, kodları ise Görsel C.52'de verilmiştir. İki adet sensör bilgisi NOKIA_LCD_BILGI değişkeninde birleşir, bu bilgiler hem seri porta hem LCD ekrana gönderilir.



Görsel C.51: Toprak nem ve yağmur sensörlerinin TIVA kartına bağlantısı

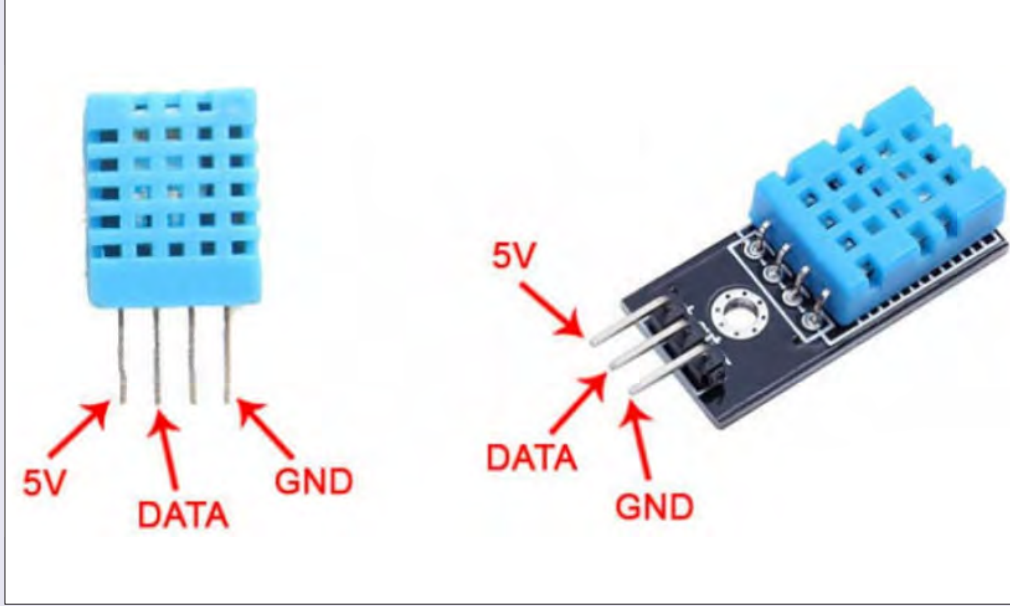
```
void Timer0IntHandler(void)
{
    ADCProcessorTrigger(ADC0_BASE, 0);
    while(!ADCIntStatus(ADC0_BASE, 0, false)) { }
    ADCIntClear(ADC0_BASE, 0);
    ADCSequenceDataGet(ADC0_BASE, 0, &ADC0DATA[0]);
    sensor1 = ADC0DATA[0]; sensor2 = ADC0DATA[1];
    sprintf((char*)NOKIA_LCD_BILGI, "%04d, %04d", sensor1, sensor2);
    UARTprintf("%s", NOKIA_LCD_BILGI); // İki sensör verisini seri porta gönder
    UARTprintf("\rSensör1: %04d, Sensör2: %04d", sensor1, sensor2);
    Nokia5110_SetCursor(1,3); Nokia5110_OutString(NOKIA_LCD_BILGI);
    // İki sensör verisini NOKIA LCD Displaye gönder.
    SysCtlDelay((SysCtlClockGet() * 1.5) / 10);
}
```

Görsel C.52: LCD'de sensör bilgilerinin gösterildiği yazılım kodları



9. Uygulama: LCD'de Dht11 Sıcaklık ve Nem Sensör Bilgilerinin Gösterilmesi

- Uygulamalar klasöründeki **LCD_SENSOR_DISPLAY** klasöründen kopya üretilir ve klasör ismi **LCD_DHT11_SENSOR** olarak değiştirilir. ARMKEIL yazılımı ile **timers.uvproj** dosyası çift tıklanarak açılır. Bu uygulamada LCD displayde DHT11 sensöründen elde edilen sıcaklık ve nem bilgisi alınarak hem seri portta hem de LCD displayde gösterilir.
- Görsel C.53'te DHT11 sensörünün pin bağlantıları görülmektedir.



Görsel C.53: DHT11 sensörünün PIN bağlantı durumları

- Görsel C.54'te DHT11 sensör kütüphanesindeki global fonksiyon, sıcaklık ve nem bilgilerini taşıyan değişkenlerin tanımlanmasını sağlayan kodlar görülmektedir.

```
int ReadDHT(); // ReadDHT fonksiyonuna dosya içerisinde global erişim için tanımlandı.
int sıcaklik=0, nem=0; // DHT11 sensöründen elde edilen verilerin atanacağı değişkenler.
```

Görsel C.54: DHT11 sensör kütüphanesinde global fonksiyon, sıcaklık ve nem bilgilerini taşıyacak değişkenlerin tanımlanması ile ilgili kodlar

- Tablo C.3'te DHT11 sensöründen elde edilen sıcaklık ve nem bilgileri seri port ve LCD'ye gönderen kodlar görülmektedir. DHT11 sensörü için kütüphane kodları, proje dosyasının içine fonksiyon olarak kodlanır.

Tablo C.3: DHT11 Sensöründen Elde Edilen Sıcaklık ve Nem Bilgilerini Seri Port ve Lcd'ye Gönderen Kodlar

No.	Kodlar (ARDUINO)
1	int ReadDHT()
2	{ // bit buffers & timeout
3	char bitcount;
4	char byte;
5	char bits[5] = {0,0,0,0,0};
6	unsigned int bitints[6];
7	long timerval;
8	int clockMhz, dongu = 100;

```

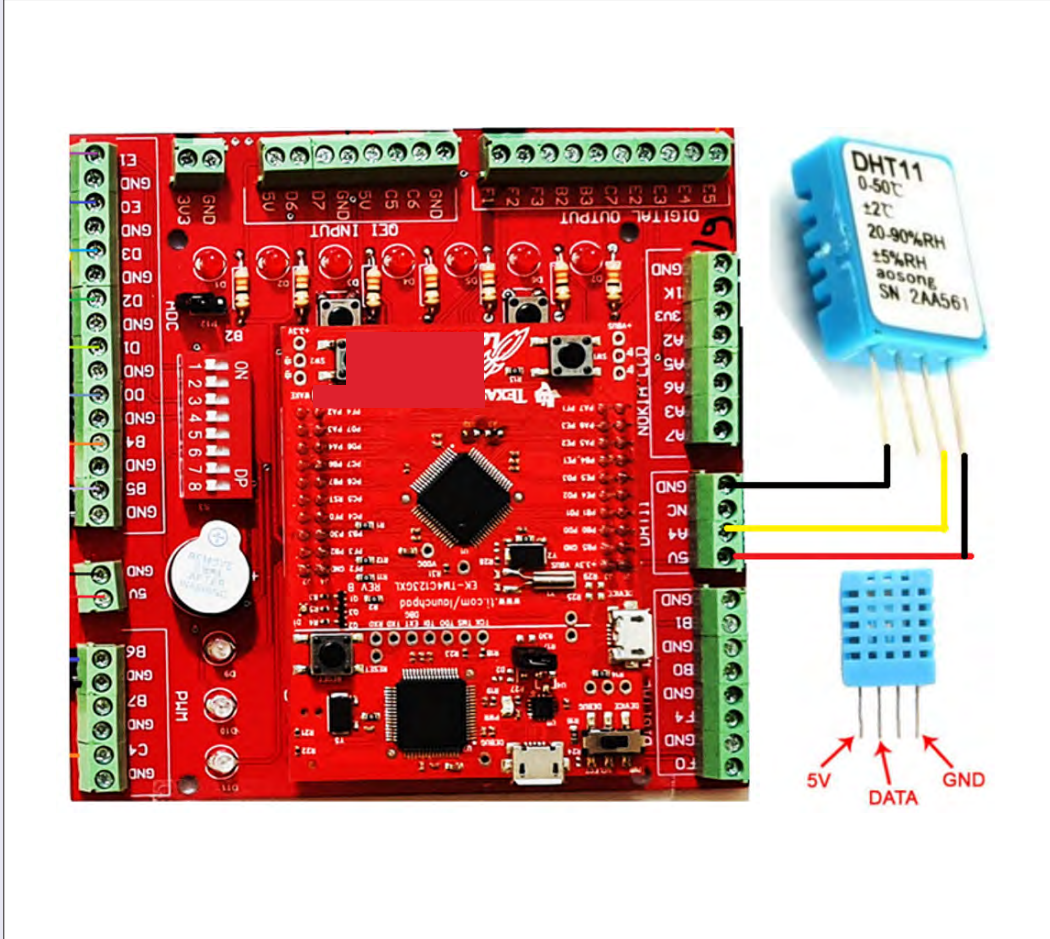
9   unsigned int loopCnt = 10000;
10  #define MY_PIN_PERIPH   SYSCTL_PERIPH_GPIOA
11  #define MY_PIN_PORTBASE  GPIO_PORTA_BASE
12  #define MY_PINNR        GPIO_PIN_4
13  #define MY_TIMER_PERIPH  SYSCTL_PERIPH_TIMER2 // SYSCTL_PERIPH_TIMER0
14  #define MY_TIMERBASE     TIMER2_BASE
15  #define MY_TIMER         TIMER_B
16
17  SysCtlPeripheralEnable(MY_TIMER_PERIPH);
18  TimerConfigure(MY_TIMERBASE, TIMER_CFG_SPLIT_PAIR | TIMER_CFG_B_ONE_SHOT);
19  SysCtlPeripheralEnable(MY_PIN_PERIPH);
20  GPIOPinTypeGPIOOutput(MY_PIN_PORTBASE, MY_PINNR);
21  GPIOPinWrite(MY_PIN_PORTBASE, MY_PINNR, 0x0);
22  //SysCtlDelay((SysCtlClockGet() /3)/30 ); // -> 30 ms
23  SysCtlDelay((SysCtlClockGet() /3)/60 ); // -> 30 ms
24  GPIOPinTypeGPIOInput(MY_PIN_PORTBASE, MY_PINNR);
25  loopCnt = SysCtlClockGet() /dongu;
26  while(GPIOPinRead(MY_PIN_PORTBASE, MY_PINNR))
27  {loopCnt = loopCnt -1; if (loopCnt == 0) return 0;}
28  loopCnt = SysCtlClockGet() /dongu;
29  while(!(GPIOPinRead(MY_PIN_PORTBASE, MY_PINNR)))
30  {loopCnt = loopCnt -1; if (loopCnt == 0) return 0;}
31  loopCnt = SysCtlClockGet() /dongu;
32  while(GPIOPinRead(MY_PIN_PORTBASE, MY_PINNR))
33  {loopCnt = loopCnt -1; if (loopCnt == 0) return 0;}
34  // start receiving 40 bits
35  char i;
36  bitcount = 7;
37  byte = 0;
38  int conta = 0;
39  int timerval2=0;
40  int largest = 0;
41  int lowest  = 99999;
42  for (i=0; i < 40; i++)
43  { loopCnt = SysCtlClockGet() /dongu;
44  while(!(GPIOPinRead(MY_PIN_PORTBASE, MY_PINNR)))
45  { loopCnt = loopCnt -1; if (loopCnt == 0) return 0; }
46  TimerDisable(MY_TIMERBASE, MY_TIMER);

```

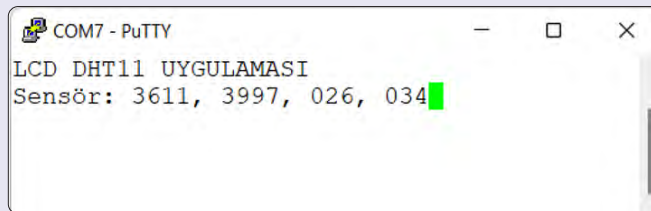


```
47 TimerConfigure(MY_TIMERBASE, TIMER_CFG_SPLIT_PAIR | TIMER_CFG_B_ONE_SHOT);
48 TimerLoadSet(MY_TIMERBASE, MY_TIMER, SysCtlClockGet() /4000);
49 TimerEnable(MY_TIMERBASE, MY_TIMER);
50 timerval = TimerValueGet(MY_TIMERBASE, MY_TIMER);
51 loopCnt = SysCtlClockGet() /dongu;
52 while(GPIOPinRead(MY_PIN_PORTBASE, MY_PINNR))
53 {loopCnt = loopCnt -1; if (loopCnt == 0) return 0;}
54 timerval2 = TimerValueGet(MY_TIMERBASE, MY_TIMER);
55 conta = timerval - timerval2;
56 if (conta > largest) largest = conta;
57 if (conta < lowest) lowest = conta;
58 if ((conta) > (1000)) bits[byte] |= (1 << bitcount);
59 if (bitcount == 0)
60 {bitcount = 7;
61 byte++;
62 } else {bitcount--;}
63 bitints[0] = ((unsigned int) bits[0] & (0x000000FF));
64 bitints[1] = ((unsigned int) bits[1] & (0x000000FF));
65 bitints[2] = ((unsigned int) bits[2] & (0x000000FF));
66 bitints[3] = ((unsigned int) bits[3] & (0x000000FF));
67 bitints[4] = ((unsigned int) bits[4] & (0x000000FF));
68 if(((bitints[0] + bitints[1] + bitints[2] + bitints[3]) & (0x000000FF)) == bitints[4])
69 {sicaklik = bitints[2];
70 nem = bitints[0];
71 return 1;
72 } else {
73 return 0;}}
74 void Timer0IntHandler (void)
75 { ADCProcessorTrigger (ADC0_BASE, 0);
76 while(!ADCIntStatus(ADC0_BASE, 0, false)) { }
77 ADCIntClear (ADC0_BASE, 0);
78 ADCSequenceDataGet(ADC0_BASE, 0, &ADC0DATA[0]);
79 sensor1 = ADC0DATA[0]; sensor2 = ADC0DATA[1];
80 ReadDHT();
81 sprintf ((char*)NOKIA_LCD_BILGI, "%04d, %04d, %03d, \"%03d\", sensor1, sensor2, sicaklik, nem);
82 UARTprintf("\rSensör: %s",NOKIA_LCD_BILGI); // İki sensör verisini seri porta gönder.
83 // UARTprintf("\rSensör1: %04d, Sensör2: %04d", sensor1, sensor2);
84 Nokia5110_SetCursor(1,3) ; Nokia 5110_OutString(NOKIA_LCD_BILGI);
85 // İki sensör verisini NOKIA LCD displaye gönder.
86 SysCtlDelay((SysCtlClockGet() * 1.5) / 10) ;
87 }
```

- Görsel C.55'te DHT11 sensörünün TIVA kontrol kartına bağlantı şeması, Görsel C.56'da ise DHT11 sensör bilgilerinin seri porta giden verilerin ekran görüntüsü görülmektedir. Görsel C.56'da "sensör: 3611 ve 3997 12 bitlik" verinin değerleri ile "26 derece" ve "%34 nem" bilgileri Putty yazılımı üzerinden görülmektedir.



Görsel C.55: DHT11 sensörünün TIVA kontrol kartına bağlantı yapısı



Görsel C.56: DHT11 sensör bilgilerinin seri porttan okunması



10. Uygulama: LCD'de Dijital Saat

- Uygulamalar klasöründeki **LCD_SENSOR_DISPLAY** klasöründen kopya üretilir ve klasör ismi **LCD_DIGITAL_SAAAT** olarak değiştirilir. ARMKEIL yazılımı ile **timers.uvproj** dosyası çift tıklanarak açılır. Uygulama ile LCD displayde **Real Time Clock** fonksiyonundaki saniye bilgisinden dijital saat yapılır. Saat bilgisi hem seri porta hem de LCD displaye gönderilir.
- Görsel C.57'de ilgili kütüphanelerin kodları görülmektedir. Projeye hibernate ve math kütüphanesi dâhil edilir. Real Time Clock fonksiyonları da bu kütüphane içindedir.

```
#include "inc/hw_hibernate.h"
#include "driverlib/hibernate.h"
#include "math.h"
```

Görsel C.57: İlgili kütüphanelerin kodları

- Görsel C.58'de projede saat bilgilerinin ve voltaj değerinin atandığı global değişkenler ile saati hesaplayan fonksiyonların tanımlandığı kod yazılımları görülmektedir.

```
void SaatGoster (void);
int32_t saat=17, dakika=18, saniye=0; // Real Time Clock için global degiskenler.
int voltaj=0, voltaj1=0; // tam sayi ve ondalikli sayilar için degisken.
float volt; // Sensörl degerinden GERILIM elde etmek için degisken.
```

Görsel C.58: Global değişkenler ile saatin hesaplandığı fonksiyon kodları

- Görsel C.59'da saati RTC'den gelen saniyelik bilgiye göre hesaplayan fonksiyon kodları ve Görsel C.60'da fonksiyondaki saat, sıcaklık ve voltaj bilgilerinin LCD ekrana gönderildiği kodlar görülmektedir.

```
void SaatGoster (void) {
    saniye = HibernateRTCGet(); // RTC'den her bir saniye bilgisi atanır.
    if (saniye > 59) {
        HibernateRTCSet(0); saniye = 1;
        if (++dakika > 59) {
            dakika = 0;
            if (++saat > 23) {
                saat = 0;
            }
        }
    }
}
```

Görsel C.59: Saati RTC'den gelen saniyelik bilgiye göre hesaplayan fonksiyon kodları

```
void Timer0IntHandler(void)
{
    ADCProcessorTrigger(ADC0_BASE, 0);
    while(!ADCIntStatus(ADC0_BASE, 0, false)) { }
    ADCIntClear(ADC0_BASE, 0);
    ADCSequenceDataGet(ADC0_BASE, 0, &ADC0DATA[0]);
    sensor1 = ADC0DATA[0]; sensor2 = ADC0DATA[1];
    ReadDHT();
    SaatGoster();
    // Sensor1 bilgisi 12 bit olarak gerilime dönüştürüldü
    volt = sensor1 * 3.3 / 4095;
    voltaj = floor(volt); // TAM SAYI kısmı.
    voltaj1 = (volt - voltaj) * 100; // ONDALIKLI SAYI kısmı.
    sprintf(BILGI, "%d.%dV", voltaj, voltaj1);
    sprintf((char*)NOKIA_LCD_BILGI, "T%02d,H%02d,%s %02d:%02d:%02d",
    sicaklik, nem, BILGI, saat, dakika, saniye);
    UARTprintf("\r%s",NOKIA_LCD_BILGI);
    Nokia5110_SetCursor(0,3); Nokia5110_OutString(NOKIA_LCD_BILGI);
    SysCtlDelay((SysCtlClockGet() * 1.5) / 10);
}
```

Görsel C.60: Timer1 fonksiyonu içinde saat, sıcaklık ve voltaj bilgilerinin LCD'ye gönderildiği kodlar

- Görsel C.61'de seri porta gönderilen saat, derece ve sensör bilgilerinin ekran görüntü bilgileri görülmektedir.



Görsel C.61: Seri porta gelen saat, derece ve sensör bilgilerinin ekran görüntüsü



11. Uygulama: Kronometre

- Uygulamalar klasöründeki **LCD_DIGITAL_SAAT** klasöründen kopya üretilir ve klasör ismi **KRONOMETRE** olarak değiştirilir. ARMKEIL yazılımı ile **timers.uvproj** dosyası çift tıklanarak açılır. Uygulama ile seri porttan kronometre bilgileri alınır. Main fonksiyonu bir önceki uygulamadaki kodlarla aynıdır.

```
39 // Kronometre Global Degiskenleri
40 int32_t ILK_DEGER = 0;
41 bool durum = false;
42 int MiliSaniye=0, Saniye=0, Dakika=0;
```

Görsel C.62: Kronometre uygulamasında kullanılacak olan global değişken kodları

```
52 void Timer0IntHandler(void) // Her 1 ms'de tetiklenecek kodlar.
53 {
54     if (durum) {
55         ILK_DEGER++;
56         ILK_DEGER = ILK_DEGER % 6;
57         MiliSaniye = ILK_DEGER % 1000;
58         if (ILK_DEGER > 1000) {
59             Saniye = (ILK_DEGER - MiliSaniye) / 1000;
60             if (Saniye > 59) {Saniye = Saniye % 60; }
61         } else {Saniye = 0;}
62         if (ILK_DEGER > 60000) {
63             Dakika = (ILK_DEGER - ILK_DEGER % 1000) / 60000;
64         } else {Dakika = 0;}
65         sprintf((char*)BILGI, "%02d.%02d.%03d", ILK_DEGER, Dakika, Saniye, MiliSaniye);
66         UARTprintf("\r%s", BILGI);
67         SysCtlDelay(80000); // TIMER0A Kesme için 1 ms ayarlama. 1000 ms = 1 sn.
68     }
69 }
```

Görsel C.63: Timer0A kesmesinde çalışan kodlar

```
71 void Timer1IntHandler(void)
72 {
73     MAP_TimerIntClear(TIMER1_BASE, TIMER_TIMA_TIMEOUT);
74     MAP_IntMasterDisable();
75     BUTON_DURUMLARI = GPIOPinRead(GPIO_PORTF_BASE, BUTONLAR);
76     if (BUTON_DURUMLARI == SOL_BUTON) { // Kronometre Durdur.
77         UARTprintf("\r%08d", ILK_DEGER);
78         if (durum) {durum = false;}
79         SysCtlDelay((SysCtlClockGet() * 0.3) / 5); // Butona basma süresi.
80     } else if (BUTON_DURUMLARI == SAG_BUTON) { // Kronometre Baslat.
81         UARTprintf("\r%08d", ILK_DEGER);
82         GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_3, 0);
83         durum = true; ILK_DEGER = 0;
84         SysCtlDelay((SysCtlClockGet() * 0.3) / 5); // Butona Basma Süresi.
85     }
86     MAP_IntMasterEnable();
87 }
88 }
```

Görsel C.64: Timer1A kesmesinde çalışan kodlar

- Görsel C.62'de kronometre uygulamasında kullanılacak global değişken kodları, Görsel C.63'te Timer0A kesmesinde çalışan kodlar ve Görsel C.64'te Timer1A kesmesinde çalışan kodlar görülmektedir. Bu kodların çalışmasının seri port ekran üzerindeki gösterimi Görsel C.65'teki gibi olmaktadır.

```
COM10 - PuTTY
KRONOMETRE. MESLEKİ TEKNİK EĞİTİM
-----
006454, 00:06.454
```

Görsel C.65: Kronometre uygulamasının seri port ekran görüntüsü



UYGULAMA ETKİNLİĞİ



Aşağıda verilenleri sırasıyla uygulayınız.

- 1. öğrenme biriminde yer alan 16 farklı uygulamadan birini seçiniz.
- Verilen devre şemasından ve kodlardan yola çıkarak devreyi ARM32 TIVA mikrodeneleyici ile tasarlayınız ve kodları derleyerek tekrardan oluşturunuz.
- Devreyi breadboard üzerinde kurunuz ve kodları yükleyerek çalıştırınız.
- 2. öğrenme biriminde yer alan 5 farklı uygulamadan birini seçiniz.
- Verilen devre şemasından ve kodlardan yola çıkarak devreyi ARM32 TIVA mikrodeneleyici ile tasarlayınız ve kodları derleyerek tekrardan oluşturunuz.
- Devreyi breadboard üzerinde kurunuz ve kodları yükleyerek çalıştırınız.
- Her iki devreyi simülasyon ortamında kurarak devrelerin çalışmasını test ediniz.
- Her iki devrede mikrodeneleyiciler arasındaki etkileri değerlendiriniz ve sonuçları raporlayarak teslim ediniz.

Değerlendirme: Çalışma aşağıda verilen ölçütler doğrultusunda değerlendirilecektir.

No.	Ölçütler	PUANLAR	
		Verilen	Alınan
1	1. öğrenme biriminden uygulama seçilmesi ve kodlarının incelenmesi	10	
2	Seçilen ilk uygulamanın devre tasarımının yapılması	10	
3	Seçilen ilk uygulamanın ARM32 bağlantılarının oluşturulması	10	
4	Seçilen ilk uygulamanın yazılım kodlarının oluşturulması ve çalıştırılması	10	
5	2. öğrenme biriminden uygulama seçilmesi ve kodlarının incelenmesi	10	
6	Seçilen ikinci uygulamanın devre tasarımının yapılması	10	
7	Seçilen ikinci uygulamanın ARM32 bağlantılarının oluşturulması	10	
8	Seçilen ikinci uygulamanın yazılım kodlarının oluşturulması ve çalıştırılması	10	
9	Sonuçlarının değerlendirilmesi ve raporlaştırılması	10	
10	Sonuç raporunun simülasyon dosyası ile teslim edilmesi	10	
TOPLAM PUAN		100	

CEVAP ANAHTARI

1. ÖĞRENME BİRİMİ: MİKRODENETLEYİCİ İLE DİJİTAL UYGULAMALAR

1	2	3	4	5
Y	D	D	Y	Y

6	7	8	9	10
4	2	#include	grafik	optokuplör

11	12	13	14	15
E	D	B	A	B

2. ÖĞRENME BİRİMİ: MİKRODENETLEYİCİ İLE ANALOG UYGULAMALAR

1	2	3	4	5
Y	D	D	Y	D

6	7	8	9	10
DS18B20	340	PWM	10	komütatör

11	12	13	14	15
E	B	A	C	D

KAYNAKÇA



Materyalin genel ađ kaynakçası ve grsel kaynakçasına buradan ulařılır. Karekoda ulařılmaması durumunda ařađıdaki linki kullanabilirsiniz.

<http://kitap.eba.gov.tr/karekod/Kaynak.php?KOD=2495>



NOTLAR

A series of horizontal dotted lines for taking notes, spanning the width of the page below the header.



NOTLAR