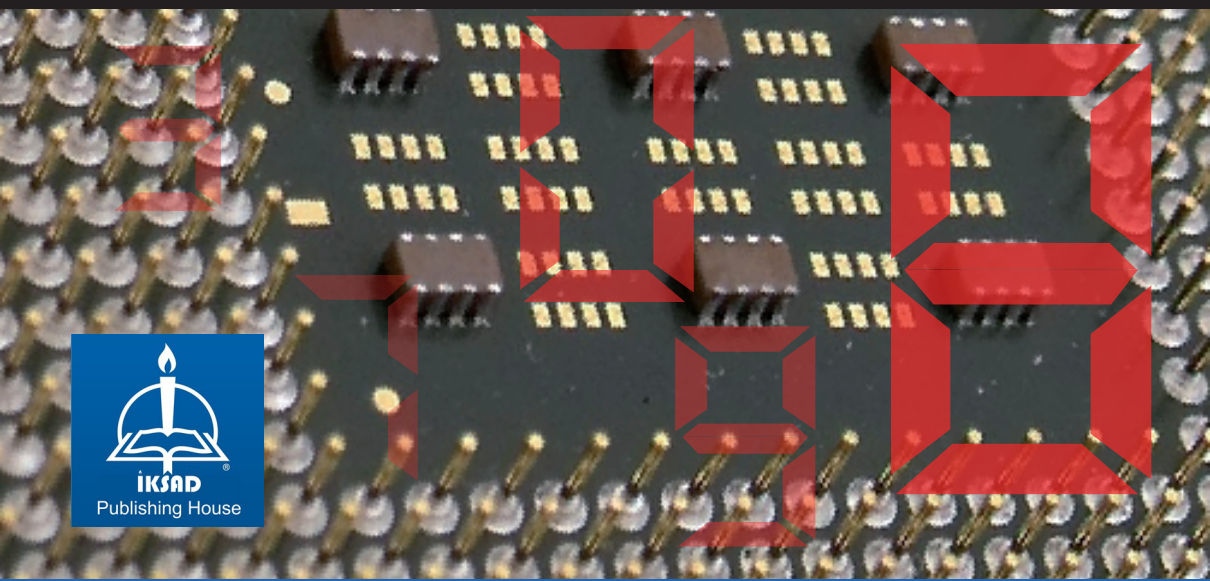
MİKRODENETLEYİCİLER VE PROGRAMLAMA

Dr. Koray ÖZSOY
Dr. Bekir AKSOY
Seyit Ahmet İNAN



İKSAD
Publishing House

MİKRODENETLEYİCİLER VE PROGRAMLAMA

Dr. Koray ÖZSOY[1]
Dr. Bekir AKSOY[2]
Seyit Ahmet İNAN[3]

İKSAD Publishing House

[1] Isparta Uygulamalı Bilimler Üniversitesi, Senirkent MYO, ISPARTA, korayozsoy@isparta.edu.tr
[2] Isparta Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi, ISPARTA, bekiraksoy@isparta.edu.tr
[3] Süleyman Demirel Üniversitesi, ISPARTA, ahmetinan@sdu.edu.tr,

(The Licence Number of Publicator: 2014/31220)
TURKEY TR: +90 342 606 06 75
USA: +1 631 685 0 853
E mail: iksadyayinevi@gmail.com
www.iksad.net
It is responsibility of the author to abide by the publishing ethics rules.
Iksad Publications – 2019©
ISBN: 978-625-7029-34-6
Cover Design: İbrahim KAYA
November / 2019
Ankara / Turkey
Size = 14,8 x 21 cm

İÇİNDEKİLER

ÖNSÖZ.....1

BÖLÜM 1

1. GİRİŞ VE MİKRODENETLEYİCİLERİN TEMELLERİ.....3

1.1. Mikrodenetleyici.....4

1.2. Mikrodenetleyicilerin Sağladığı Üstünlükler.....8

1.3. Mikrodenetleyicilerin Dezavantajları.....8

Kaynaklar.....9

BÖLÜM 2

2. ALGORİTMA VE PROGRAMLAMAYA GİRİŞ.....11

Kaynaklar.....19

BÖLÜM 3

3. 18F45K22 MİKRODENETLEYİCİ VE ÖZELLİKLERİ.....21

3.1. CCP modülü.....28

3.2. Timer modülü.....29

3.3. Kesme (interrupt) modülü.....30

3.4. Anolog Dijital Dönüştürücü (ADC) modülü 31

3.5. Osilatör Modülü.....33

3.6. Güç Yönetim Modları.....34

Kaynaklar.....35

BÖLÜM 4

4. MİKRO C YAZIM KURALLARI, DEĞİŞKEN TİPLERİ VE UYGULAMALAR……39

4.1. Noktalama İşaretleri……..39

4.1.1. Ayraçlar……..39

4.1.2. Parantez……..39

4.1.3. Küme Ayraçları……40

4.1.4 Virgül……40

4.1.5 Noktalı Virgül……41

4.1.6. İki Nokta Üst-Üste……41

4.1.7. Yıldız İşareti……41

4.1.8. Eşittir İşareti……41

4.1.9. Diyez İşareti……42

4.2. Değişken Tipleri……42

4.3.Temel Tipler……42

4.3.1. Aritmetik Tipler……43

4.3.2.Tümleşik Tipler……43

4.3.3.Kayan Noktalı Tipler……43

4.4. Ek Yazım Kuralları……44

4.5. Sabitler……45

4.5.1.Tamsayı Sabitler……46

4.5.2. Karakter Sabitleri……46

4.5.3. String Sabitler……46

4.6. PIC 18F45K22 Mikrodenetleyici İle Trafik Işığı Uygulaması……47

Kaynaklar……51

BÖLÜM 5

5. MİCRO C DİLİNDE ÖNİŞLEMCİ DİREKTİFLERİ.....53

5.1.1 # Operatörü.....53

5.1.2 ## Operatörü.....53

5.2 Önişlemci Direktifleri.....55

5.2.1 #define komutu.....55

5.2.2 # include komutu.....56

Kaynaklar.....58

BÖLÜM 6

6. DİZİLER 59

6.1. Giriş.....59

6.1. Tek Boyutlu Diziler.....59

6.1.1. Tek Boyutlu Dizilerde Atama.....60

6.2. Çok boyutlu diziler.....62

6.3. Karakter dizileri.....65

6.3.1. Karakterlerin hafızada saklanması.....66

Kaynaklar.....67

BÖLÜM 7

7. YEDİ PARÇALI GÖSTERGE (SEVEN SEGMENT DİSPLAY) İLE 0-99 SAYICI UYGULAMASI.....69

7.1. GİRİŞ.....69

7.2. Yedi Parçalı Gösterge (Seven Segment Display).....69

7.3. Sayıcı Devreleri.....75

7.3.1. Asenkron Sayıcılar.....75

7.3.2. Senkron Sayıcılar……74

7.4. MikroC ile Sayıcı Devresi Uygulaması……75

Kaynaklar……79

BÖLÜM 8

8. MİKRODENETLEYİ TABANLI ANALOG SAYISAL DÖNÜŞTÜRME (ADC) VE SICAKLIK ÖLÇME VE KONTROL UYGULAMASI……81

8.1. GİRİŞ……81

8.2. Analog Sayısal Dönüştürme (ADC)……81

8.4. PIC 18F45K22 MİKRODENETLEYİ İLE SICAKLIK UYGULAMASI……85

Kaynaklar……89

BÖLÜM 9

9. MİKRODENETLEYİCİ İLE LCD TEXT 2x16 YAZI YAZDIRMA VE DİJİTAL SAAT……91

9.1. Sıvı Kristal Ekran (Liquid Crystal Display/LCD)……91

9.2. PIC Mikrodenetleyici ile LCD Kullanımı……91

9.2.1. LCD Panel Bacakları ve Bağlantı Şekilleri……92

9.3. Mikrodenetleyici ve LCD Bağlantısı……93

9.4. Mikro C LCD Kütüphanesi……94

9.4.1. Kütüphane İşlevleri……94

9.4.2. LCD Sabit Fonksiyonları……96

Kaynaklar……100

BÖLÜM 10

10. MİKRODENETLEYİCİ İLE SERİ PORT İLE HABERLEŞME……101

10.1. Seri Haberleşme……101

10.2. RS-232 Seri İletişim Protokolü……102

10.3. Seri Haberleşmede UART Kullanımı……104

Kaynaklar……108

BÖLÜM 11

11. MİKRODENETLEYİCİ İLE PWM KONTROLÜ……111

11.1. GİRİŞ……111

11.2. Analog ve Sayısal Sinyaller……111

11.3. Darbe Genişliği Modülasyonu (PWM)……113

11.4. MikroC Programlama PWM Kütüphanesi……114

11.6. PWM DC Motor (Fan) Kontrolü Uygulaması……117

Kaynaklar……120

BÖLÜM 12

12. MİKRODENETLEYİCİ İLE ADIM (STEP) MOTOR KONTROLÜ……123

12.1. Adım (Step) Motorlar……123

12.2. Adım (Step) Motor Çeşitleri……124

12.2.1 Bipolar Step Motorlar……124

12.2.2 Unipolar Step Motorlar……125

12.3 Step Motorlarda Uç Tespitinin Yapılması……125

12.4 Step Motorun Çalışması……126

12.5 Step Motor Sürücü (ULN2003)……126

12.6 Mikrodenetleyici ile Fonksiyon Oluşturma……..127

12.7 Mikrodenetleyici ile Step Motor Kontrol Uygulaması……..127

Kaynaklar……130

BÖLÜM 13

13. MİKRODENETLEYİCİ İLE GERÇEK ZAMAN SAATİ (RTC) UYGULAMASI.133

13.1. Giriş……133

13.2. Gerçek Zaman Saati (Real Time Clock/RTC)……133

13.3. PCF8583P Entegre Gerçek Zaman Saati Özellikleri……134

Kaynaklar……142

BÖLÜM 14

14. MİKRODENETLEYİCİ İLE KABLOSUZ HABERLEŞME……..143

14.1. Kablosuz Haberleşme……143

14.2. HC-06 Bluetooth Modülü……145

14.3. HC-05 Bluetooth Modülü……146

14.4. HC-05 Ve HC-06 Ortak Özellikleri……147

Kaynaklar……150

Özgeçmişler……152

ÖNSÖZ

Dijital dönüşüm, hızla gelişen bilgi ve iletişim teknolojilerinin sunduğu imkânlar ve değişen toplumsal ihtiyaçlar doğrultusunda iş süreçleri ve teknoloji unsurlarında gerçekleştirilen bütüncül bir dönüşümdür. Teknoloji hızlı bir şekilde değişmektedir. Böylece tüm dünyanın açık pazar haline geldiği müşteri beklentileri ile ekonomik, hızlı, standart, güvenli ve kaliteli ürünler istenmektedir. Bu durum endüstriyel otomasyon kavramını ortaya çıkarmıştır. Endüstriyel otomasyon; bir fabrikada veya endüstriyel çalışma alanında, insanlar ile makineler arasında gerçekleşen iş dağılımıdır. Otomasyon endüstride, yönetimde ve bilişim alanında gerçekleştirilen işlemlerde insan vasıtasıyla gerçekleştirilen işlerin otomatik olarak yapılabilmesi için programlama tekniği oldukça önemlidir.

Programlama tekniği, problemin çözümü için bilgisayarlara komut vererek birtakım işlemlerin yapılmasını sağlayan bağımsız düşünce sanatına verilen isimdir. Programlama mantığının temelini algoritmalar ve akış diyagramları oluşturmaktadır. Programlama dillerinde olduğu gibi mikrodenetleyiciler kullanarak yapılan programlamalarda da iyi bir algoritma bilgisine sahip olmak oldukça önemlidir. Bu nedenle diğer programlama dillerinde olduğu gibi mikrodenetleyicilerde herhangi bir sorun çözülmek istendiğinde, sorunun iyi bir şekilde anlaşılması gereklidir. Mikrodenetleyici, bir bilgisayarın temel bölümleri olan hafıza ve giriş/çıkış ünitelerinin bir yonga içine gömülü olarak üretilmiş şeklidir. Mikrodenetleyici birçok farklı firma tarafından üretilmektedir. Bunlar içerisinde fiyat düşüklüğü ve ulaşılabilirliği kolay olması nedeniyle Microchip firması tarafından üretilen PIC ailesi (Peripheral Interface Controller/ çevresel ünite denetleme arabirimi) en çok tercih edilen mikrodenetleyicilerdir. Günümüzde mikrodenetleyiciler otomobillerde, görüntü sistemlerinde, biyomedikal uygulamalarda, klima ve ısıtıcılarda, kameralarda, cep telefonlarında, baskı makinelerinde, ses ve görüntü sistemlerinde vb. birçok alanda kullanılmaktadır.

Kitap, mesleki ve teknik eğitim veren orta ve yükseköğretim kurumlarındaki öğrenciler için kaynak olması düşünülerek hazırlanmıştır. Kitapta Mikrodenetleyicilerin temelleri ve uygulama örnekleri akademik bir dille ele alınmıştır. Kitap, mikrodenetleyici ve programlama konusunda daha önceden hiçbir bilgisi olmayan bireyleri başlangıç düzeyinden alarak ileri düzeye kadar getirecek biçimde

hazırlanmıştır. Kitap da her bölüm doyurucu örnekler ile işlenmiştir. Önemli bilgiler, çizelgeler ve şekiller bir araya getirilerek teknik elemanların yararlanabileceği bir kaynak olarak hazırlanmıştır. Kitap elektrik-elektronik, mekatronik ve bilgisayar teknolojilerinde eğitim gören öğrencilere ve bu alanlarda çalışan mühendisler için hazırlanmıştır. Kitabın okuyucularına, akademiye ve bilime yararlı olacağını umuyoruz.

Saygılarımızla…

Dr. Koray ÖZSOY, Dr. Bekir AKSOY, Öğr. Gör. Seyit Ahmet İNAN

Aralık, 2019

BÖLÜM 1

1. GİRİŞ VE MİKRODENETLEYİCİLERİN TEMELLERİ

Teknolojinin hızla gelişmesi ile birlikte üretimde otomasyon önem kazanmıştır. Hızlı bir biçimde gelişmesi sürdürülmektedir. Maliyeti düşürmek ve kaliteli ürün imal edebilmek için endüstriyel otomasyon kontrol sistemleri şirketlerde gitgide önemi artmıştır. Hızlı üretimin sonucu olarak kontrol teknolojileri sürekli değişim halinededir. Bu nedenle işlem ve kontrol birimlerinin tümünün bir arada bulunduğu kompakt sistemler (mikroişlemci sistemleri vb.) geliştirilmiştir.

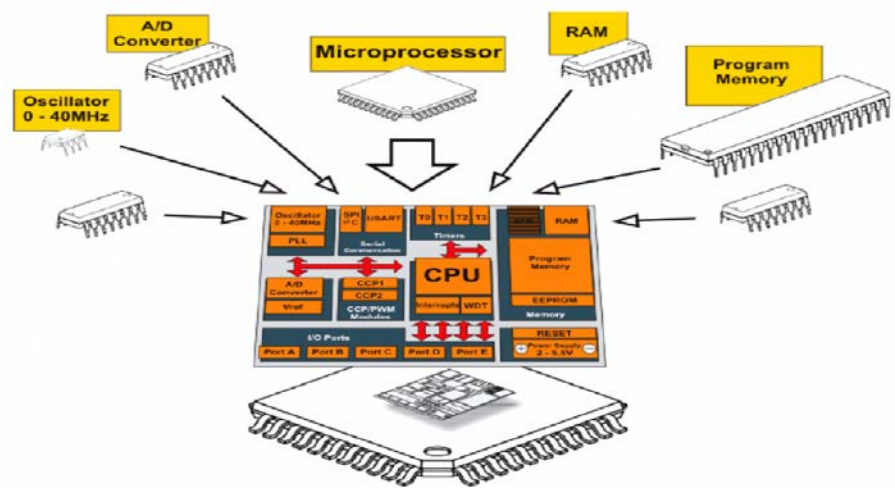
Mikroişlemcinin ilk yıllarında endüstride ve bilgisayarda aynı mikroişlemciler kullanılmıştır. Daha sonraki yıllarda bilgisayar daha hızlı işlem yapan işlemcilere gereksinim duyulmuştur. Endüstrinin ise yavaş fakat içerisinde sıkça kullanılan yardımcı birimleri içeren mikroişlemciler istemesi nedeniyle endüstrinin gereksinimi için yeni arayışlara girilmiştir. Mikroişlemciler, bilgisayar programlarının yaptığı basit kontrol işlemlerini (giriş-çıkış, ADC, PWM vb.) kolayca yerine getirmektedir. Başka bir ifadeyle CPU (Central Processing Unit-CPU) olarak adlandırılır. Bir mikroişlemcinin işlevini yerine getirmek için yardımcı elemanlara ihtiyaç vardır [1]. Mikroişlemciden farklı olarak, giriş ve çıkış birimleri içerisinde bulunduran ve tek bir çipte barındıran bütünleşik yapıya mikrodenetleyiciler denir [2]. Mikroişlemciler temel matematiksel ve mantıksal işlemleri yapmak üzere tasarlanmıştır. Mikrodenetleyiciler ise girişden almış oldukları bilgileri içersindeki yazılım doğrultusunda işleyerek çıkış sinyal üreten tümleşik devredir. Ayrıca mikrodenetleyici uygulamalarında, baskı devre maliyetleri düşük, güç tüketimi azdır. Bu nedenle endüstriyel uygulamalarda çoğu kez tercih edilmektedir [3]. General Instruments firması tarafından 1970 yılında Harvard mimarisine sahip ilk mikrodenetleyici birimi, Signetics 8X300 modeli üretilmiştir. Peripheral Interface Controller (Çevrebirim Arayüz Denetleyicisi-PIC) giriş/çıkış portu olmak üzere 16 bitlik mikrodenetleyici birimi (MPU) programlanabilmek üzere geliştirilmiştir [4]. Daha sonra General Instruments firması elektronik bölümünü satarak, 1988 yılında Microchip Technology adında yeni bir şirket kurularak PIC üretimi hızlanmıştır. 1989'da ilk piyasaya sürülen aile PIC16C5X serisi olmuştur. Endüstriyel ihtiyaçlar doğrultusunda 14-bitlik işlemciye sahip PIC16CXXX ailesi, 1992 yılında üretilmiştir. Bu mikrodeneyleyicide diğer üretilen PIC ailesinden farklı olarak daha büyük

bellek, ADC çevirici, timer(zamanlayıcı) ve interrupt (kesme) ve sayıcılar gibi işlemeri yapabilme özelliğine sahiptir. Daha sonra, 16-bit işlemciye sahip PIC18CXXX ailesi 1999 yılında da komut seti geliştirilerek endüstriyel uygulamalarında kullanılmak üzere piyasaya sunulmuştur.

Günümüzde IC kompakt imal eden başlıca firmalar Atmel, Texas Instruments, Michrochip, Intel, National Semiconductror, NEC vb. firmalardır. Endüstride yapılan işler büyük, küçük ve orta ölçekli olarak sınıflandırılır. Bu işlere uygun mikrodenetleyici seçilir. Bu seçim sırasında ihtiyaç duyulan gereksinimlere göre mikrodenetleyiciler üretilmektedir. Bu gereksinimler, programın büyüklüğü, I/O sayısı, veri işleme hızları, seri iletişim (UART) ihtiyaçlarına göre değişmektedir. Son yıllarda Mikrodenetleyicilerin (PIC) kullanımının yaygınlaşması ile birlikte açık döngülü endüstriyle otomasyon sistemlerinde çoğu kez kullanılmaktadır. Ayrıca uygun fiyatı nedeniyle popülerliği giderek artmaktadır [5]. Günümüzde mikrodenetleyiciler otomobillerde, görüntü sistemlerinde, biyomedikal uygulamalarda, klima ve ısıtıcılarda, kameralarda, cep telefonlarında, baskı makinesi, ses ve görüntü sistemlerinde gibi sayılmayacak birçok alanda kullanılmaktadır [6].

1.1. Mikrodenetleyici

Mikrodenetleyiceler endüstriyel sistem tasarımında otomatik kontrol amacıyla maliyeti ve güç tüketimi azlığı nedeniyle sıkça tercih edilmektedir. Mikrodenetleyici, bir bilgisayarın temel bölümleri olan hafıza ve giriş/çıkış ünitelerinin bir yonga içine gömülü olarak üretilmiş şeklidir. Mikrodenetleyici, dışarıdan gelen bir veriyi (programı) hafızasına alan, derleyen ve sonucunda da çıktı elde eden bir bilgisayardır. Mikrodenetleyici yapısında CPU, RAM, ROM, giriş çıkış pinleri, seri ve paralel portlar ve bazı modellerde olmak üzere A/D (Analog to Digital) çeviriciler barındırır (Şekil 1.1).



Şekil 1.1. Mikrodenetleyicinin blok şeması [7]

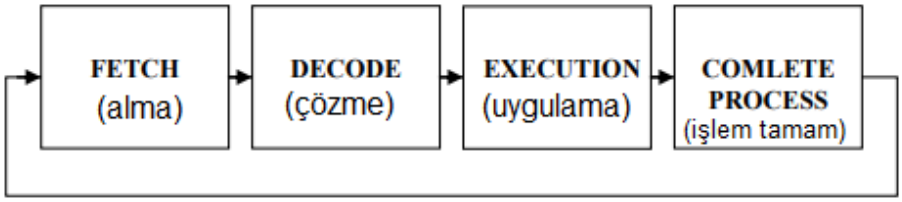
Mikrodenetleyici, hafızasına programlanan yazılımı işleyebileceği şekilde derler ve komutlara göre bir çıkış sinyali gönderir. Örneğin mikrodenetleyicinin çıkış portuna bağlı olan bir piston gelen sinyale göre hareket etmeye başlayacaktır. Özetle mikrodenetleyiciler eletkronik devrelerde beyin işlevi görür ve elektronik sistemleri kontrol etmektedir. Mikrodenetleyici birçok farklı firma tarafından üretilmektedir. Bunlar içerisinde fiyat düşüklüğü ve ulaşılabilirliği kolay olması nedeniyle Microchip firmasının tarafından üretilen PIC ailesi (Peripheral Interface Controller/ çevresel ünite denetleme arabirimi) en çok tercih edilen mikrodenetleyicidir. 12F, 16F, 18F, vb. şeklinde kodlanan PIC ailesi, hızlı, basit bir yapıya sahiptir, kolay programlanır, ucuzdur ve az sayıda çevre birimi gerektirir [8].

Mikroişlemci mimarisi iki mimarisi vardır. Bunlar RISC (Azaltılmış komut seti/ Reduced Instruction Set Computer) ve CISC (Karmaşık komut seti/ Complex Instruction Set Computer) tabanlı işlemcilere göre tipleri vardır. PIC mikrodenetleyici ailesi genellikle piyasada bu iki işlemci sınıfından birisine sahiptir. PIC mikrodenetleyiciler hızlı çalışmaları amacıyla RISC işlemci olarak tasarlanmışlardır [9,10].

Mikrodenetleyici üretici firmalarının kullandıkları mimari türleri aşağıdaki gibidir:

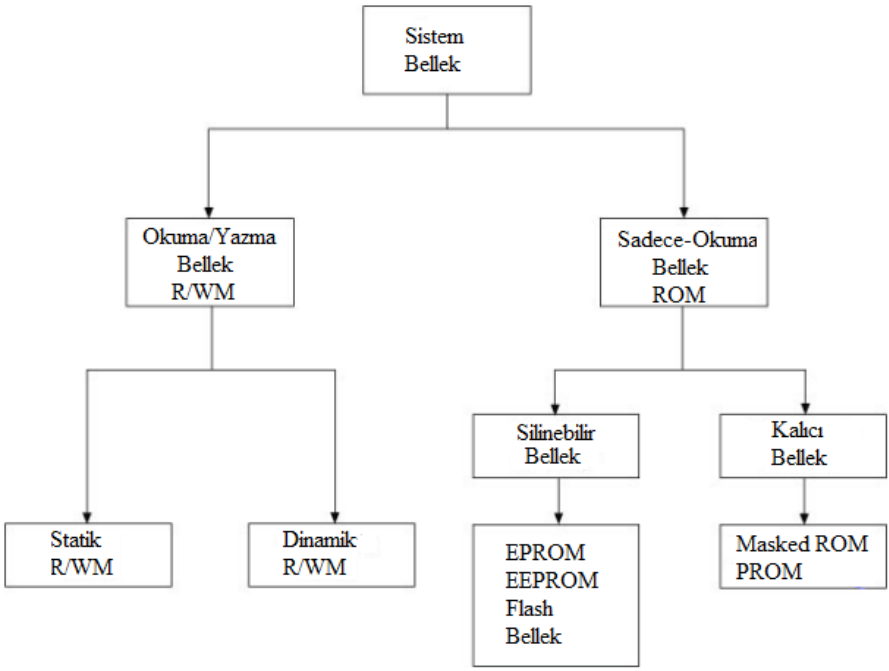
- Microchip firması RISC,
- Intel firmasının CISC,
- Atmel firmasını RISC,

Mikrodenetleyici bir komutun işlenme süresi dört (4) aşamada gerçekleşmektedir. Bunlar alma(fetch), kod çözme(decode), uygulama (execution), işlem tamamlama (complete process) süreçleridir (Şekil 1.2).



Şekil 1.2. Mikrodenetleyici komut işleme süreci [11]

Mikrodenetleyiciler farklı amaçta kullanılan program bellekleri içerisinde bulunmaktadır. Mikro denetleyicinin iki adet bellek vardır. Bunlar program belleği, flash bellek (okunur yazılabilir bellek) ve RAM(rastgele erişim bellek) bellektir. Program bellekleri EPROM (Erasable Programmable Read Only Memory), EEPROM (Electrically Alterable Read-only Memory), ROM (Read Only Memory) şeklindedir. EPROM salt okunur bellek sıkça kullanılan bellek tipidir. Bu tip belleği silebilmek için ultraviole ışık kaynağı gerekmektedir. Beş (5) dakika ile yarım saat arasında bir süre zarfında gerçekleştirilir. EPROM bellek hücrelerine mikrodenetleyici programlayıcı vasıtasıyla elektriksel sinyal uygulayarak kayıt yapılır. Daha sonra elektrik kesilse bile bellek içersindeki veri silinmez. EEPROM bellek üzerindeki program silinip başka bir program yazılacağı zaman mikrodenetleyici programlayıcısı ile elektriksel sinyal gönderilerek silme ve programlama işlemi gerçekleştirilmektedir. ROM bellek ise de salt okunur bellek çerisinde verilerin okunamasına imkân tanıyan ancak değiştirilemeyen bellektir. Üretildikleri fabrikada bir defaya özel yazılmaktadır. Bu nedenel program geliştirmeye müsait değildir. Bu bellekler araçlarda, fırınlarda, bulaşık makinesinde vb. mikrodenetleyici kontrol sistemlerinde tercih edilmektedir. Bellek sınıflandırılması Şekil 1.3'de detaylı gösterilmiştir.



Şekil 1.3. Bellek çeşitlerinin sınıflandırılması

Mikrodenetleyici ile uygulama geliştirilirken kullanılacak modüller, hafıza büyüklüğü, dahili zamanlayıcı ve harici kesme vb. özellikler listelenmelidir [12]. Mikrodenetleyici seçiminde özellikler ve dikkat edilmesi gereken birimler aşağıda verilmiştir:

- Çalışma palsi,
- Osilatör tipleri (XT, RC vb.)
- I/O sink-source akımları,
- Analog-Dijital Dönüştürücü (ADC) birimi
- Anolog ve dijital ayarlanan I/O pin sayısı,
- Seri haberleşme (UART) birimi,
- Haberleşme protokolü (SPI, I^2C USART vb.)
- Darbe genişlik modülasyonu (PWM) birimi,
- Harici ve dahili kesme (interrupt) sayısı
- Zamanlayıcı (timer) birimi,
- Flash bellek kapasitesi,
- Bellek tipi (RAM, ROM, EPROM, ve EEPROM)

- Dâhili rastgele erişebilir bellek (RAM) büyüklüğü
- USB arabirim,
- CAN bus, ethernet bağlantı [13]

1.2. Mikrodenetleyicilerin Sağladığı Üstünlükler

Mikrodenetleyicilerin bazı avantajları aşağıda verilmiştir:

- Mikrodenetleyicilerin sistemlerin tasarımı ve kullanımı mikroişlemcili sistemlere göre daha sade ve ucuzdur.
- Mikrodenetleyicili sistemlerde sistemin çalışması için elemanın kendisi ve bir osilatör kaynağının olması yeterlidir.
- Mikrodenetleyicilerin küçük ve ucuz olması elektronik kontrol devrelerinde sıkça kullanılmasını sağlamaktadır [14].
- Mikrodeneyleyicilerde işlem süresi gerçekleştirmek için düşük zaman gerekir.
- Mikrodenetleyici işlemci yongaları çok küçük ve esnektir.
- Mikrodenetleyicilere ek RAM, ROM ve I/O portlarını arayüz eklemek oldukça kolaydır.
- Herhangi bir dijital parça olmadan mikrobilgisayar görevi görebilir [15]

1.3. Mikrodenetleyicilerin Dezavantajları

Mikrodenetleyicilerin bazı dezavantajları aşağıda verilmiştir:

- Mikrodenetleyici, yüksek güçlü cihazlara doğrudan arayüz olamaz.
- Mikroişlemciye göre daha karmaşık bir yapıya sahiptir.
- Aynı anda sınırlı sayıda işlem gerçekleştirmektedir.
- Genellikle mikro ekipmanlarda kullanılmaktadır.

Kaynaklar

[1] Saraoğlu, H.M. "Seramik Ve Çini Fırın Sıcaklığının Mikrokontrolcü (Hpc) İle Denetimi." Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü Dergisi 002: 166-171, 2001.

[2] Bayram, U., & Çetinkaya, V. (2007). Kütüphane Otomasyonu. *IV. Otomasyon Sempozyumu, Syf*, 69-71.

[3] Çotuk, H., "PIC mikro denetleyicileri için gerçek zamanlı işletim sistemi", TOBB Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 6-7 (2008).

[4] F. A. Kazan, A. C. Aaçayak, C. Arslan, and M. Selek, MikroC le PIC18F4550 Uygulamalar, 1th ed. Konya: Mesleki Akademi, 2014.

[5] Aydoğdu, Ö., & Bayer, M. PIC Tabanlı Fırçasız DC Motor Sürücüsü Tasarımı. *Elektrik-Elektronik Mühendisliği Bölümü Selçuk Üniversitesi, Konya*, 2008.

[6] Altınbaşak, O. *Mikrodenetleyiciler ve PIC programlama*. Altaş, 2001.

[7] Mikrodenetleyici Avantaj ve Dezavantaj, Erişim Tarihi: 19/10/2019. https://www.polytechnichub.com/advantages-disadvantages-microcontroller/

[8] Öner, Y., Gürdal, O., Çetin, E., & Çetin, M. Küresel motor tabanlı güvenlik otomasyonu, 2007.

[9] Robert A. Ravenscroft, Jr., Using a PIC32 microcontroller and simulator to teach computer organization, Journal of Computing Sciences in Colleges, v.27 n.6, p.135-141, June 2012

[10] Bakırcılar, S., & Özcerit, A. Programlanabilir CPLD tabanlı akıllı mikrodenetleyici eğitim seti tasarımı ve uygulaması. *Sakarya University Journal of Science*, *19*(2), 123-133. 2015.

[11] Mikrodenetleyiciler MEGEP Ders notları, MEB MTE Program Modülleri, 2013.

[12] Okyay, M.O., A Portable Real Time Operating System for Embedded Platforms, Y.Lisans Tezi, İzmir İleri Teknoloji Enstitüsü, İzmir, 2004.

[13] Çotuk, Hüseyin. PIC mikrodenetleyiciler için gerçek zamanlı işletim sistemi. MS thesis. TOBB Ekonomi ve Teknoloji Üniversitesi-Fen Bilimleri Enstitüsü-Bilgisayar Mühendisliği Anabilim Dalı, 2008.

[14] Milli Eğitim Bakanlığı, Elektrik Elektronik Teknolojisi, Mikroişlemci ve Mikrodenetleyiciler, 523EO0019, Ankara, 67s. 2012.

[15] Mikroe, Erişim Tarihi: 19/10/2019 https://www.mikroe.com/ebooks/PIC-microcontrollers-programming-in-assembly/PIC16f887-microcontroller-device-overview

BÖLÜM 2

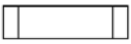
2. ALGORİTMA VE PROGRAMLAMAYA GİRİŞ

Programlama tekniği, problemin çözümü için bilgisayarlara komut vererek birtakım işlemlerin yapılmasını sağlayan bağımsız düşünce sanatına verilen isimdir [1]. Programlama, yalnızca belirli bir programlama dilini bilmek değil, aynı zamanda içinde tasarlama, problem çözme, düzenleme ve hata ayıklama gibi boyutların bulunduğu bir süreçtir [2]. Programlama dili ile makine dilinde birden çok komutla yapılabilecek işlemler tek komutla daha kolay ve hızlı bir biçimde yapılabilir. Fakat yazılacak algoritmaya göre programlama dilinde hazırlanan komutların sırası veya biçiminin bilinmesi oldukça önemlidir [3].

Programlama mantığının temeli algoritmalar ve akış diyagramları oluşturmaktadır [4]. Algoritma sonlu bir işi tanımlamada kullanılan, açık bir şekilde tanımlanıp yürütülebilen ardışık işlemlerin bir bütünü olarak tanımlanmaktadır [5]. Gökoğlu (2017), algoritmayı bir programın yapacağı işlemin tanımlanmasında kullanılan, açık bir şekilde ifade edilebilen, birbiri ardına dizilmiş adımlarla yazılması olarak tanımlanmıştır [6]. Programlama dillerinde olduğu gibi mikrodenetleyiciler kullanarak yapılan programlamalarda da iyi bir algoritma bilgisine sahip olmak oldukça önemlidir. Bu nedenle diğer programlama dillerinde olduğu gibi mikrodenetleyicilerde de herhangi bir sorunu çözmek istediğimizde sorunun iyi bir şekilde anlaşılması gereklidir [7].

Programlamanın ilk aşaması programın algoritmasını hazırlamaktır. Algoritmalar programlamanın tasarımında kullanılan ve programın geliştirilmesi için gereken aşamaların her programlama dili için uygun bir biçimde anlatıldığı ortak yapılardır [8]. Algoritmaların hazırlanması programların daha basit bir şekilde yazılmasına sağlayan ve programlama aşamaları içerisinde en önemli adımlarından bir tanesidir [9]. Algoritma hazırlama sırasında kullanılan adımlar kısa, öz ve anlaşılır olmalıdır [8]. Algoritmalar hazırlanırken her bir adım belirleyici olmalı ve tüm ihtimaller göz önünde bulundurulmalıdır [10].

Algoritma hazırlandıktan sonra; akış diyagramı ile programın yapması gereken işlemler akış diyagramı ile gösterilir [11]. Akış diyagramı çeşitli anlamlar ifade eden ve birbirlerine oklarla bağlanan dikdörtgen, baklava, elips ve daire vb. gibi görsel sembollerle algoritmanın adımlarını ifade etmektedir [12]. Akış diyagramları oluşturulurken kullanılan bazı semboller ve anlamları Şekil 2.1'de verilmiştir.

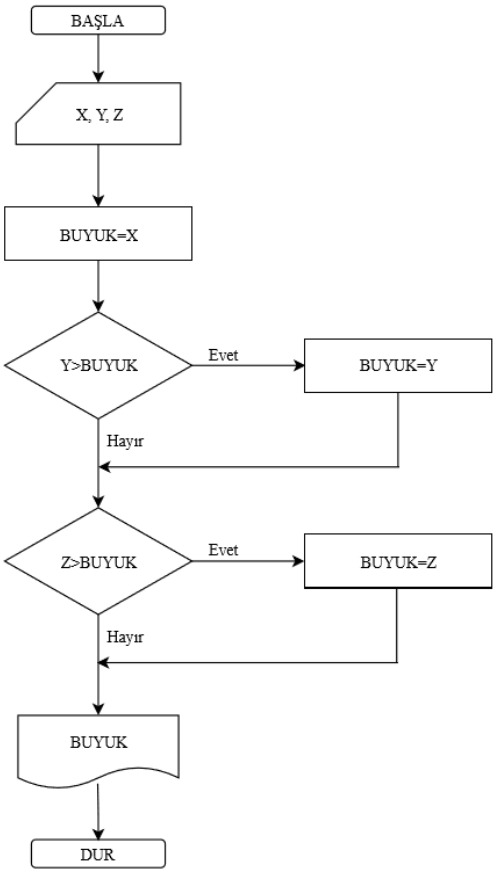
SEMBOL	ANLAMI
	Başla / Dur
	Veri / bilgi girişi
	İşlem
	Döngü
	Karar / Karşılaştırma
	Veri / bilgi yazma
	Önceden tanımlı işlem (alt program, fonksiyon..)
	El ile veri girişi
	Bağlantı
	İşlem akış yönü

Şekil 2.1. Akış diyagramlarında kullanılan bazı semboller ve anlamları

Aşağıda verilen algoritmada girilen 3 tamsayıdan en büyüğünü bulan algoritma hazırlanmıştır. Algoritmaya ait akış diyagramı da Şekil 2.2'de verilmiştir.

Algoritma

A1. Başla
A2. X, Y, Z sayılarını gir.
A3. BUYUK = X
A4. Eğer Y>BUYUK ise BUYUK = Y
A5. Eğer Z> BUYUK ise BUYUK = Z
A6. BUYUK yaz.
A7. DUR

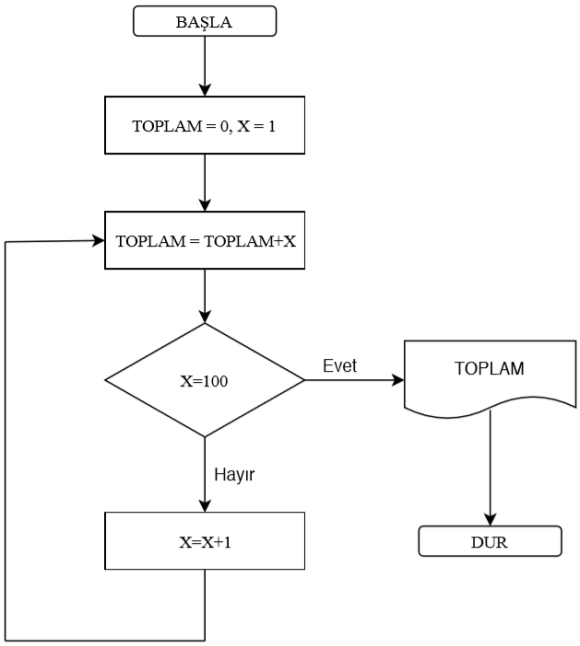


Şekil 2.2 Girilen üç tane sayıdan büyüğünü bulan akış diyagramı

1'den 100'e kadar olan tamsayıların toplamını bulan algoritma ve akış diyagramı Şekil 2.3'de verilmiştir.

Algoritma

A1. Başla
A2. TOPLAM = 0, X = 1
A3. TOPLAM = TOPLAM + I
A4. Eğer I = 100 ise A6'ya git.
A5. I=I+1 al ve A3'e git.
A6. TOPLAM yaz.
A7. DUR.

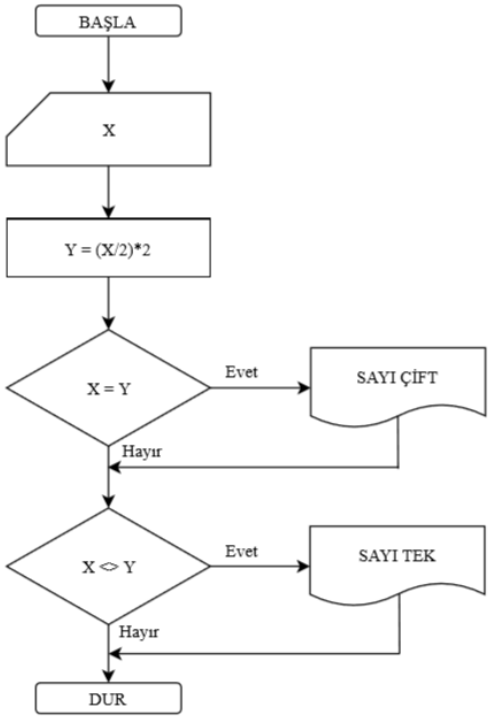


Şekil 2.3. 1'den 100'e kadar olan sayıların toplamını bulan akış diyagramı

Aşağıda hazırlanan algoritmada klavyeden girilen bir tamsayının tek sayı mı yoksa çift sayı mı olduğunu bulan algoritma hazırlanmış ve akış diyagramı Şekil 2.4'de verilmiştir.

Algoritma

A1. Başla
A2. X sayısını gir.
A3. Y = TAM (X/2)*2
A4. Eğer A = B ise "SAYI ÇİFT" yaz.
A5. Eğer A<>B ise "SAYI TEK" yaz.
A6. DUR



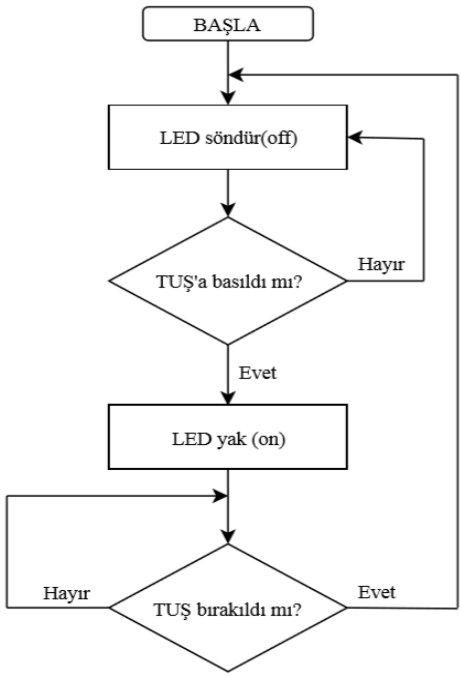
Şekil 2.4. Klavyeden girilen bir tamsayının tek/çift olduğu bulan akış diyagramı.

Mikrodenetleyici kullanılarak led yakıp söndürme işlemine ait algoritmanın adımları Şekil 2.5'de verilmiştir. Bu algoritmaya ait akış diyagramı ise Şekil 2.1'de verilen semboller ve anlamlarından yararlanılarak oluşturulmuştur. Algoritma ve akış diyagramlarının kullanımı ile temel örnekler aşağıda verilmiştir.

Algoritma

A1. Başlangıç ayarlamalarını yap.
A2. Ledi söndür.
A3. Butonun basılı olup olmadığını kontrol et.
A4. Buton basılı değilse 3. adıma dön.
A5. Buton basılı ise ledi yak.
A6. Butonun basılı olup olmadığını kontrol et.
A7. Buton basılı ise 6.adıma dön.
A8. Buton basılı değilse 2.adıma dön.

Yukarıda verilen algoritmaya uygun olarak akış diyagramının çizimi aşağıda verilmiştir.



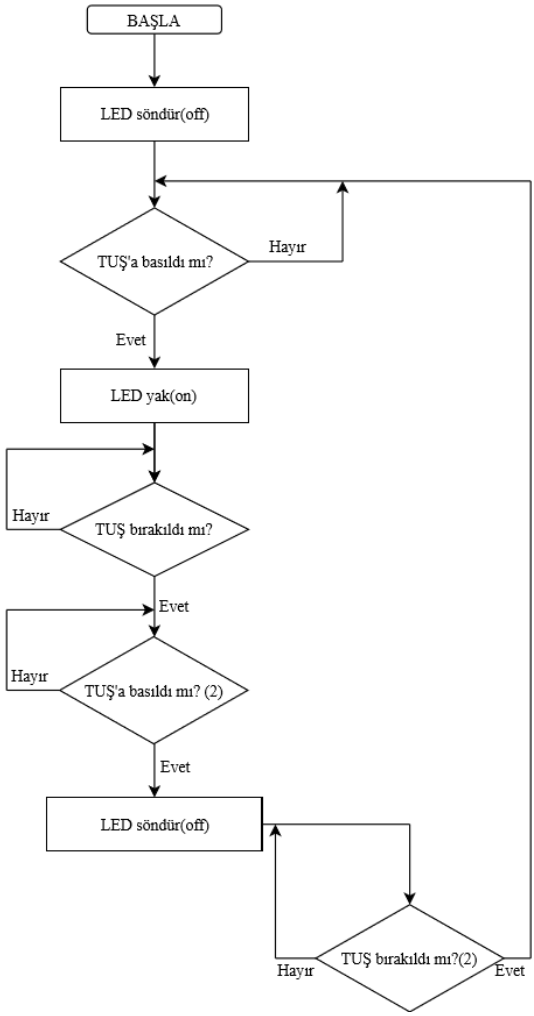
Şekil 2.5. Mikrodenetleyici ile led yanma/sönme akış diyagramı

Mikrodenetleyici kullanılarak led yakıp söndürme işlemi için tuş kontrolü ile led yakıp söndürme işlemine ait algoritmanın adımları şekil 2.6'de verilmiştir. Bu algoritmaya ait akış diyagramı ise şekil 2.1'de verilen semboller ve anlamlarından yararlanılarak oluşturulmuştur.

Algoritma

A1. Başlangıç ayarlamalarını yap.
A2. Ledi söndür.
A3. Butonun basılı olup olmadığını kontrol et.
A4. Buton basılı değilse 3. adıma dön.
A5. Buton basılı ise ledi yak.
A6. Butonun basılı olup olmadığını kontrol et.
A7. Buton basılı ise 6.adıma dön.
A8. Butonun basılı olup olmadığını kontrol et.
A9. Buton basılı değilse 8. adıma dön.
A10. Buton basılı ise ledi söndür.

A11. Butonun basılı olup olmadığını kontrol et.
A12. Buton basılı ise 11.adıma dön.
A13. Buton basılı değilse 3.adıma dön.



Şekil 2.3. Mikrodenetleyici ile tuş kontrollü led yakma/söndürme akış diyagramı

Mikrodenetleyiciler programlamasında algoritma ve akış diyagramlarının kullanılması ile ilgili akademik literatür incelendiğinde;

De la Hoz vd. (2015), yapmış olduğu çalışmalarında mikrodenetleyici bir sistem kullanılarak kompleks tabanlı bir şifreleme modülü için algoritma ve akış diyagramı kullanarak bir yazılım gerçekleştirilmiştir [13]. Diğer bir çalışmada, PIC 18F4560 mikro denetleyici kullanılarak güneş takibi için mekanik bir tasarım gerçekleştirilirken izleme algoritması ve akış diyagramı oluşturulmuştur [14]. Islam ve Sharif (2009) çalışmalarında, fotovoltaik uygulama için mikrodenetleyiciler tabanlı sinüzoidal PWM invertör kullanımında tek fazlı PWM sinyali için bir algoritma ve akış diyagramı kullanmışlardır [15]. Diğer başka bir çalışmada ise mikro denetleyici kontrollü mobil robotların hareket kontrolünde master verici programın hazırlanmasında algoritma ve akış diyagramı kullanılmıştır [16]. Banerji çalışmasında mikrodenetleyicili GSM şebekesi kullanarak insansız araç yapımında algoritma ve akış diyagramı kullanmıştır [17].

Kaynaklar

[1] Saygıner, Ş., & Tüzün, H. (2017). Programlama eğitiminde yaşanan zorluklar ve çözüm önerileri.

[2] Lane, C. H. (2004). Natural Language Tutoring and The Novice Programmer. Doktora Tezi, University of Pittsburg, USA.

[3] Ders, C. P. Notları.

[4] Durak, G. (2009). Algoritma konusunda geliştirilen" programlama mantığı öğretici-PM Ö." yazılımının öğrenci başarısına etkisi (Master's thesis, Balıkesir Üniversitesi Fen Bilimleri Enstitüsü).

[5] Özkan, Y., Programlama Dilleri: C ile Programlama, Alfa yayınları, İstanbul, (2003).

[6] Gökoğlu, S. (2017). Programlama Eğitiminde Algoritma Algısı: Bir Metafor Analizi. *Cumhuriyet International Journal of Education*, *6*(1), 1-14.

[7] Köse, U., Tüfekçi, A. (2015). "Al12goritma ve akış şeması kavramlarının öğretiminde akıllı bir yazılım sistemi kullanımı", Pegem Eğitim ve Öğretim Dergisi, Cilt:5, Sayı:5, Sayfa:569-586.

[8] Arabacıoğlu, T., Bülbül, H. İbrahim., Filiz, A. (2007). "Bilgisayar Programlama Öğretiminde Yeni Bir Yaklaşım" IX. Akademik Bilişim Konferansı, Sayfa:193-197.

[9] Aytekin, A., Çakır, F. S., Yücel, Y. B., Kulaözü, İ. (2018). Algoritmaların Hayatımızdaki Yeri ve Önemi. Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi, *5*(7), 143-150.

[10] 'C' ile Programlamaya Giriş", A. Tekin, A. Akbal, B. Sevinç, F. Ertam, H. H. Tuzsuzoğlu, İ. Serhatlıoğlu, K. Balıkçı, M. F. Talu, M. Çıbuk, O. Özdemir, R. Daş ,Y. Akbulut Z. Genç, 200,

[11] Bacaksız, G. (1998). Tasarlama Sürecinde İletişim ve Bilgi.

[12] Özcan, Ö. Ü. C. Hafta 1 Programlamaya Giriş.

[13] Zapateiro De la Hoz, M., Acho, L., & Vidal, Y. (2015). An experimental realization of a chaos-based secure communication using arduino microcontrollers. The Scientific World Journal, 2015.

[14] Abas, M. A., Kadir, S. A., & Azim, A. K. (2010, December). Improved structure of solar tracker with microcontroller based control. In 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies (pp. 55-59). IEEE.

[15] Islam, S. M., & Sharif, G. M. (2009, December). Microcontroller based sinusoidal PWM inverter for photovoltaic application. In 2009 1st International Conference on the Developements in Renewable Energy Technology (ICDRET) (pp. 1-4). IEEE.

[16] Yasuda, G. I. (2000, March). Microcontroller implementation for distributed motion control of mobile robots. In 6th International Workshop on Advanced Motion Control. Proceedings (Cat. No. 00TH8494) (pp. 114-119). IEEE.

[17] Banerji, S. (2013). Design and Implementation of an Unmanned Vehicle using a GSM Network with Microcontrollers. arXiv preprint arXiv:1306.6125.

BÖLÜM 3

3. 18F45K22 MİKRODENETLEYİCİ VE ÖZELLİKLERİ

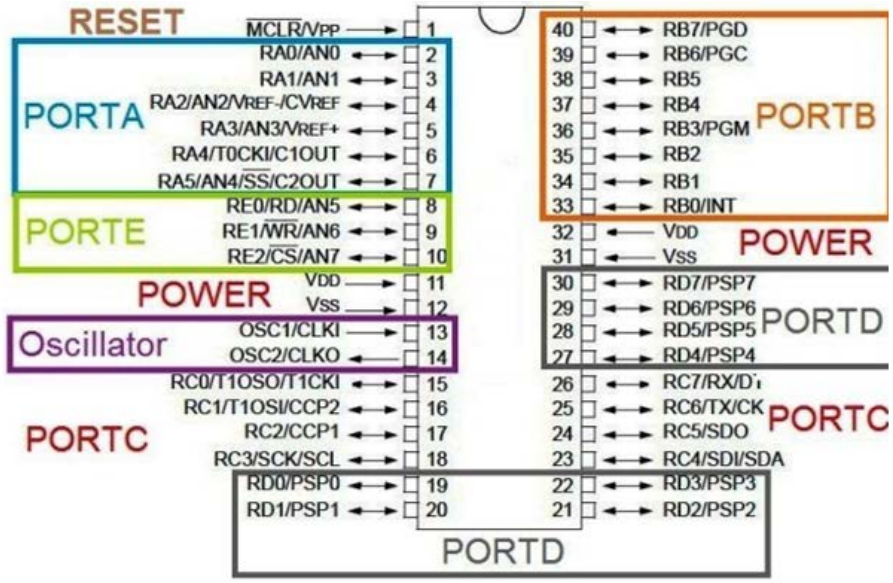
PIC18F45K22 mikrodenetleyicisi Mikrochip firmasının ürettiği [1] ve Aşırı Düşük Güç (XLP) teknolojisini kullanan düşük güç tüketimine sahip bir mikrodenetleyicidir [2]. PIC18F45K22 mikrodenetleyicisi bünyesinde barındırdığı UART ara birimi, SPI ve I^2C haberleşme protokolleri sayesinde çevre birim elemanları ile haberleşmesi ve işlem yapması kolay olan bir mikrodenetleyicidir. 18F45K22 mikrodenetleyicisi 32768 byte flash program belleğine, 256 byte EEPROM belleğine ve 1536 byte SRAM belleğine sahiptir [3]. Ayrıca PIC mikrodenetleyicileri 4 farklı kılıf ve pin yapısına sahiptir. Şekil 3.1'de gösterildiği gibi 18F45K22 mikrodenetleyici için kılıf paketi 40 pin PDIP modeli kullanılmaktadır. Burada pinlerden 2 tanesi VCC, 2 tanesi ise GND pini olup, geri kalan 36 pin giriş çıkış pini olarak adlandırılmaktadır [1]. Mikrodenetleyicinin çalışabilmesi ve içerisindeki programı yürütmesi için saat sinyali üreten devre elemanına ihtiyacı vardır [4]. Bu ihtiyacı giderebilmek için mikrodenetleyicinin 13 ve 14 numaralı bağlantı bacaklarına 4-20 MHz frekansında sinyaller üreten kristal osilatör bağlantısı yapılabilir[5].

PIC18F45K22 mikrodenetleyicimizin port işlevlerine göre 11 ile 32 numaralı pinlere +5V (VDD) uygulanıp 12 ile 31 numaralı pinlere GND bağlantısı yapılarak besleme yapılmaktadır. 1 numaralı bağlantı pinine mikrodenetleyici istendiği zamanda reset yapmak yani programı en başa almak ve silinebilir hafızasında bulunan verileri temizlemek ve silmek için kullanılmaktadır [6,7]. Mikrodenetleyiciyi resetlemek için 1 numaralı bağlantı pinine bağlantısı yapılan 4.7 K **Ω** dirence sahip pull-up devresine paralel GND hattına bağlı buton bağlantısı yapılmaktadır [1]. Reset pini bağlantısı üzerinde pull-up devresinin kullanım amacı ortamdaki ve çevredeki ekipman ve elektronik cihazların etrafa yaydığı yayılımı ve elektriksel gürültünün neden olabileceği kararsızlık durumunun giderilmesi içindir[8].

PIC18F45K22 mikrodenetleyicsi üzerinde mavi renkler ile temsil edilen PORT A bağlantılı pinler 2-7 numaralı pinlerden oluşmaktadır. Her bir bağlantı içinde pinler birbirinden bağımsız olarak giriş ve çıkış pini olarak atanabilmektedir. Mikrodenetleyici bünyesindeki PORTA pinleri genel olarak karşılaştırma, analog giriş, analog dijital dönüştürücü (ADC), sayıcı, genel amaçlı giriş/çıkış pini olarak kullanılmaktadır [1]. Mikrodenetleyici bünyesinde bulunan ve 6 numaralı pinde yer alan RA4 isimli port diğer PORTA pinlerinden farklı olarak **"Schmitt Tetikleyici"** girişi olarak

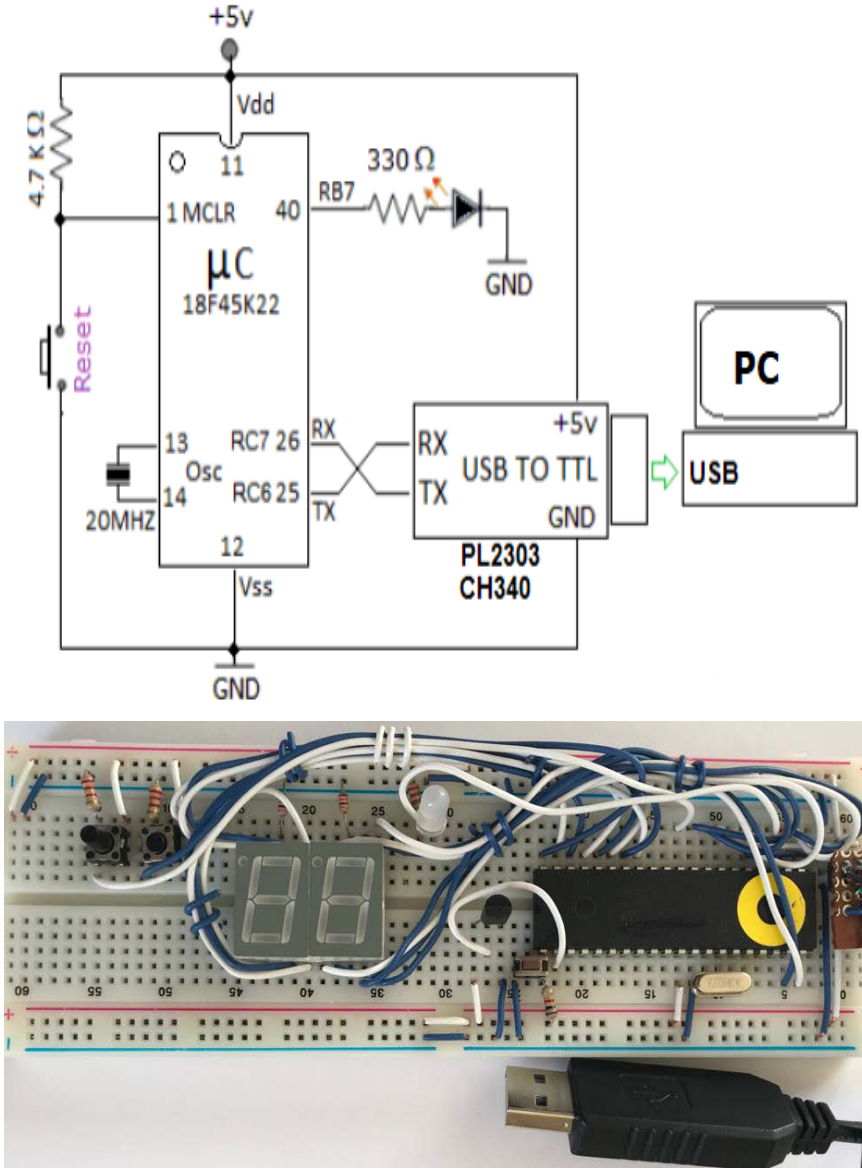
çalışabilmektedir. Schmitt tetikleyici, pozitif geri beslemeli iki konumlu olan kare ve üçgen dalga üretebilen devre elemanıdır [9].

Şekil 3.1'de açık kahve renkte temsil edilen PORT B pinleri birbirinden bağımsız giriş çıkış pini olarak atanabilen, 33-40 numaralı pinlerdir. Şekil 3.1'de renksiz temsil edilen bağlantı pinler PORT C pinleri olup 15-18 ve 23-26 numaralı portlarda yer almaktadır. Aynı PORTA ve PORTB de olduğu gibi PORTC, PORTD ve PORTE pinleri de birbirinden bağımsız şekilde giriş veya çıkış pini olarak atanıp kontrol edebilmektedir. Şekil 3.1'de gri renkte temsil edilen 19-22 ve 27-30 numaralı bağlantı pinleri PORTD pinleri olarak adlandırılır. Son olarak Şekil 3.1'de açık yeşil renk ile temsil edilen ve 8-10 numaralı pinler PORTE pini olarak kullanılmaktadır.



Şekil 3.1. PIC18F45K22 mikrodenetleyicinin pin diyagramı ve portların gösterimi

Şekil 3.2a' gösterilen bootloader (ön yükleyici) devresi ile MikroC de kodlar yazılıp derlendikten sonra. HEX dosyasını mikrodenetleyicinin program hafızasına gönderilir. Şekil 3.2b'de gösterilen board üzerindeki devre ile PIC18F45K22 mikrodenetleyici programlanarak istenilen projeler gerçekleştirilir.



Şekil 3.2. a) PIC18F45K22 mikrodenetleyici bootloader elektronik devre şeması, b) breadbard üzerinde elektronik elemanların gösterimi

Mikrodenetleyici programlanırken bir adet USB to TTL dönüştürücü kullanılır. TTL dönüştürücünün RX bacağı mikrodenetleyici üzerinde 25 numaralı bağlantı pininde bulunan RC6 portuna bağlanmaktadır. Dönüştürücü

üzerindeki TX bağlantı bacağı ise mikrodenetleyici üzerindeki 26 numaralı bağlantı pininde bulunan RC7 portuna bağlanmaktadır. USB to TTL dönüştürücüye ait +5V ve GND güç bağlantı noktalarının ise mikrodenetleyicinin bağlı olduğu güç hattına bağlantısı yapılır. Bağlantıları tamamlanan mikrodenetleyici artık programlanmaya hazırdır. PIC18F45K22 mikrodenetleyicisini programlamak için C, CCS, PIC C, PIC Basic Pro ve MikroC gibi birçok programlama dili kullanılır [12-16]. Kitapda MikroC programlama dili baz alınarak bütün kodlar bu programlama diline göre hazırlanmıştır. Şekil 3.2'de mikrodenetleyi programlamak için hazır hale getirilmiş USB to TTL dönüştürücü kullanılarak programlaması yapılacak olan devrenin diyagramı verilmiştir.

Tablo 3.1: PIC18F45K22 mikrodenetleyici özellikleri

Program hafıza tipi	Flash
Program hafıza boyutu (KB)	32
CPU hızı(mıps/dmıps)	16
Sram (KB)	1536
EEPROM(BYTE)	256
Dijital çevre haberleşme birimleri	2-UART, 2-SPI, 2-I^2C, 2-MSSP(SPI/I^2C)
Yakalama/karşılaştırma/PWM	2 CCP, 3 ECCP
Zamanlayıcılar	3 x 8-bit, 4 x 16- bit
ADC giriş	28 ch, 10-bit
Karşılaştırıcı sayısı	2
Çalışma sıcaklığı(°C)	-40 ile 125
Çalışma voltajı(V)	1.8 ile 5.5
Pin sayısı	40
Düşük güç tüketimi	Evet

PIC18F45K22 mikrodenetleyicisinin pin numaraları, adları, işlevleri ve özellikleri aşağıda verilmiştir.

Pin No	Pin Adı	İşlevi	Açıklama
2	RA0/C12IN0-/AN0	RA0	İki yönlü dijital I/O
		C12IN0-	C1 ve C2 karşılaştırıcıları eviren giriş
		AN0	Analog giriş 0
3	RA1/C12IN1-/AN1	RA1	İki yönlü dijital I/O
		C12IN1-	C1 ve C2 karşılaştırıcıları eviren giriş
		AN1	Analog giriş 1
4	RA2/C2IN+/AN2/ DACOUT/VREF-	RA2	İki yönlü dijital I/O
		C2IN+	C2 karşılaştırıcı evirmeyen giriş
		AN2	Analog giriş 2
		DACOUT	DAC referans çıkışı
		VREF-	A/D referans gerilim (low) giriş
5	RA3/C1IN+/AN3/ VREF+	RA3	İki yönlü dijital I/O
		C1IN+	C1 karşılaştırıcı evirmeyen giriş
		AN3	Analog giriş 3
		VREF+	A/D referans gerilim (high) giriş
6	RA4/C1OUT/ SRQ/T0CKI	RA4	İki yönlü dijital I/O
		C1OUT	C1 karşılaştırıcı çıkışı
		SRQ	SR latch Q çıkışı
		T0CKI	Timer 0 harici saat girişi
7	RA5/C2OUT/SRNQ/ SS1/HLVDIN/AN4	RA5	İki yönlü dijital I/O
		C2OUT	C2 karşılaştırıcı çıkışı
		SRNQ	SR latch Q' çıkışı
		SS1	SPI ikincil seçim girişi (MSSP1)
		HLVDIN	High-Low Voltage Detect (yüksek-düşük gerilim algılayıcı) girişi
		AN4	Analog giriş 4
14	RA6/CLKO/OSC2	RA6	İki yönlü dijital I/O
		CLKO	Saat çıkışı
		OSC2	Osilatör kristal çıkışı
13	RA7/CLKI/OSC1	RA7	İki yönlü dijital I/O
		CLKI	Harici saat kaynağı girişi
		OSC1	Osilatör kristal girişi

36	RB3/CTED2/P2A/ CCP2/C12IN2-/AN9	RB3	İki yönlü dijital I/O
		CTED2	CTMU Edge 2 girişi
		P2A	Geliştirilmiş CCP2 PWM çıkışı
		CCP2	Capture 2 girişi / Compare 2 çıkışı / PWM 2 çıkışı
		C12IN2-	C1 ve C2 karşılaştırıcıları eviren giriş
		AN9	Analog giriş 9
37	RB4/IOC0/ T5G/AN11	RB4	İki yönlü dijital I/O
		IOC0	Değişim kesmesi pini
		T5G	Timer 5 harici saat kapı girişi
		AN11	Analog giriş 11
38	RB5/IOC1/P3A/CCP3 /T3CKI/T1G/AN13	RB5	İki yönlü dijital I/O
		IOC1	Değişim kesmesi pini
		P3A	Geliştirilmiş CCP3 PWM çıkışı
		CCP3	Capture 3 girişi / Compare 3 çıkışı / PWM 3 çıkışı
		T3CKI	Timer 3 saat girişi
		T1G	Timer 1 harici saat kapı girişi
		AN13	Analog giriş 13

17	RC2/CTPLS/P1A/ CCP1/T5CKI/AN14	RC2	İki yönlü dijital I/O
		CTPLS	CTMU darbe üretec çıkışı
		P1A	Geliştirilmiş CCP1 PWM çıkışı
		CCP1	Capture 1 girişi / Compare 1 çıkışı / PWM 1 çıkışı
		T5CKI	Timer5 saat girişi
		AN14	Analog giriş 14
18	RC3/SCK1/ SCL1/AN15	RC3	İki yönlü dijital I/O
		SCK1	SPI mod (MSSP) için senkron seri saat giriş çıkışı
		SCL1	I2C mod (MSSP) için senkron seri saat giriş çıkışı
		AN15	Analog giriş 15
23	RC4/SDI1/ SDA1/AN16	RC4	İki yönlü dijital I/O
		SDI1	SPI data girişi (MSSP)
		SDA1	I2C data giriş çıkışı (MSSP)
		AN16	Analog giriş 16
24	RC5/SDO1/AN17	RC5	İki yönlü dijital I/O
		SDO1	SPI data çıkışı (MSSP)
		AN17	Analog giriş 17

25	RC6/TX1/ CK1/AN18	RC6	İki yönlü dijital I/O
		TX1	EUSART asenkron verici çıkışı
		CK1	EUSART senkron saat pini
		AN18	Analog giriş 18
26	RC7/RX1/ DT1/AN19	RC7	İki yönlü dijital I/O
		RX1	EUSART asenkron alıcı girişi
		DT1	EUSART senkron data pini
		AN19	Analog giriş 19
19	RD0/SCK2/ SCL2/AN20	RD0	İki yönlü dijital I/O
		SCK2	SPI mod (MSSP) için senkron seri saat giriş çıkışı
		SCL2	I2C mod (MSSP) için senkron seri saat giriş çıkışı
		AN20	Analog giriş 20
20	RD1/CCP4/SDI2/ SDA2/AN21	RD1	İki yönlü dijital I/O
		CCP4	Capture 4 girişi / Compare 4 çıkışı / PWM 4 çıkışı
		SDI2	SPI data girişi (MSSP)
		SDA2	I2C data giriş çıkışı (MSSP)
		AN21	Analog giriş 21

Pin No	Pin Adı	İşlevi	Açıklama
21	RD2/P2B/AN22	RD2	İki yönlü dijital I/O
		P2B	Geliştirilmiş CCP2 PWM çıkışı
		AN22	Analog giriş 22
22	RD3/P2C/ SS2/AN23	RD3	İki yönlü dijital I/O
		P2C	Geliştirilmiş CCP2 PWM çıkışı
		SS2	SPI ikincil seçim girişi (MSSP)
		AN23	Analog giriş 23
27	RD4/P2D/ SDO2/AN24	RD4	İki yönlü dijital I/O
		P2D	Geliştirilmiş CCP2 PWM çıkışı
		SDO2	SPI data çıkışı
		AN24	Analog giriş 24
28	RD5/P1B/AN25	RD5	İki yönlü dijital I/O
		P1B	Geliştirilmiş CCP1 PWM çıkışı
		AN25	Analog giriş 25

29	RD6/P1C/TX2/ CK2/AN26	RD6	İki yönlü dijital I/O
		P1C	Geliştirilmiş CCP1 PWM çıkışı
		TX2	EUSART asenkron verici çıkışı
		CK2	EUSART senkron saat pini
		AN26	Analog giriş 26
30	RD7/P1D/RX2/ DT2/AN27	RD67	İki yönlü dijital I/O
		P1D	Geliştirilmiş CCP1 PWM çıkışı
		RX2	EUSART asenkron alıcı girişi
		DT2	EUSART senkron data pini
		AN27	Analog giriş 27
8	RE0/P3A/ CCP3/AN5	RE0	İki yönlü dijital I/O
		P3A	Geliştirilmiş CCP3 PWM çıkışı
		CCP3	Capture 3 girişi / Compare 3 çıkışı / PWM 3 çıkışı
		AN5	Analog giriş 5
9	RE1/P3B/AN6	RE1	İki yönlü dijital I/O
		P3B	Geliştirilmiş CCP3 PWM çıkışı
		AN6	Analog giriş 6
10	RE2/CCP5/AN7	RE2	İki yönlü dijital I/O
		CCP5	Capture 5 girişi / Compare 5 çıkışı / PWM 5 çıkışı
		AN7	Analog giriş 7
1	RE3/VPP/MCLR	RE3	Dijital giriş
		VPP	Programlama gerilim girişi
		MCLR	Aktif-0 Master-Clear (Reset) girişi
11,32	VDD	VDD	Pozitif güç kaynağı
12,31	VSS	VSS	Toprak

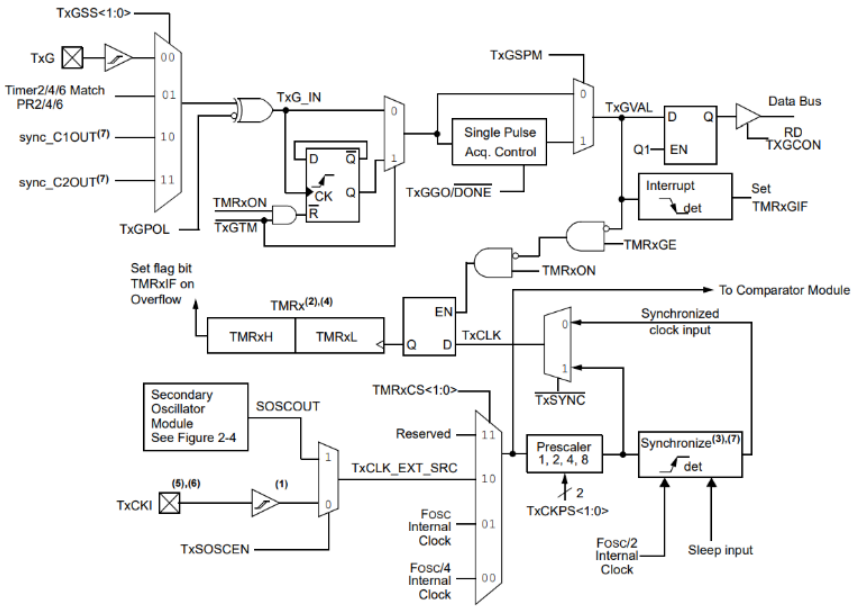
PIC18F45K22 mikrodenetleyicisinin desteklediği CCP modülü, timer modülü, interrupt modülü, ADC modülü, osilatör modülü ve güç yönetim modları kapsayan özellikler aşağıda verilmiştir.

3.1. CCP modülü

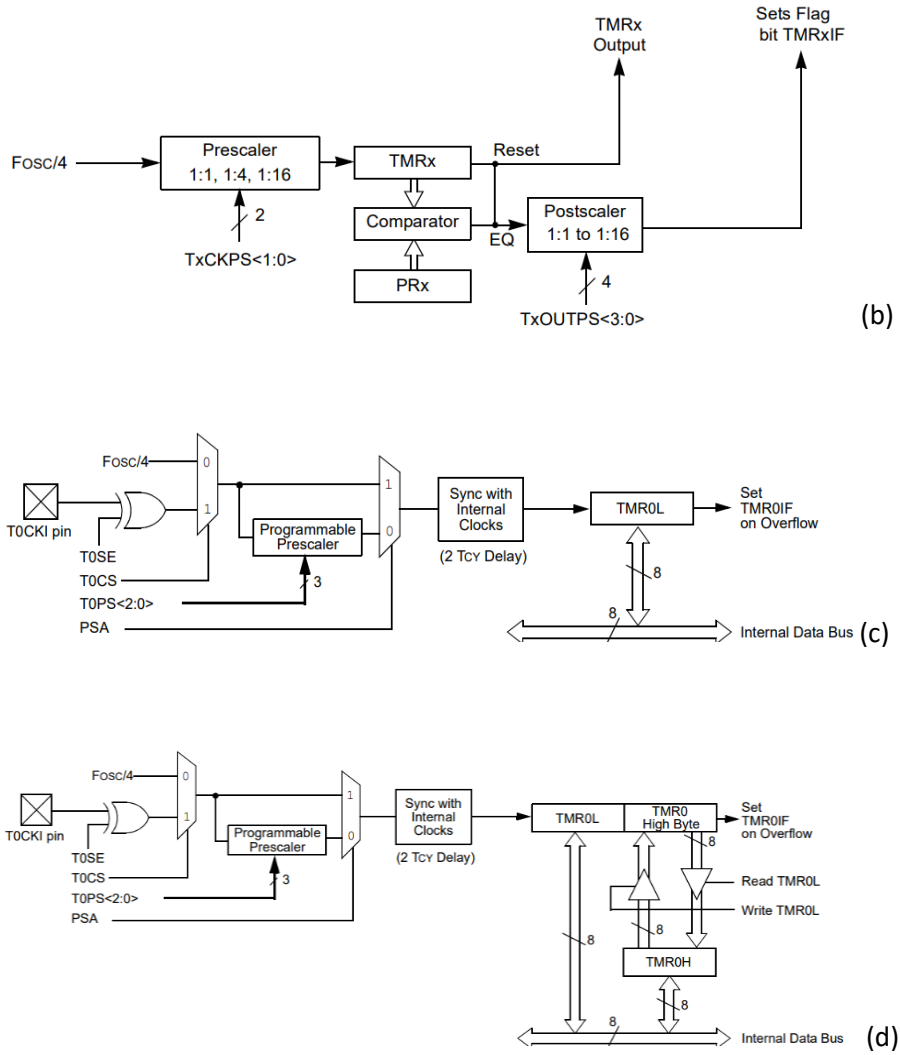
CCP açılımı Capture-Compare-PWM (Yakalama- Karşılaştırma-PWM) ile üç farklı fonksiyonu olan modüldür. CCP modülülün özelliklerinde yer alan "compare" yani karşılaştırma özelliği, sayma işlemini için kullanılmaktadır. PIC18F45K22 mikrodenetleyicisi "compare" işlemini yaparken 16-bit kaynaklı timer1, timer 3 ve timer 5'i kullanır. CCP modulünün bir diğer özelliği ise "capture" yani yakalama özelliğidir. Bu özellik bir sinyalin frekansını veya darbe genişliğini hesaplamak için kullanılmaktadır. CCP son birimi ise çoğu kişinin bildiği ve kullandığı PWM (Darbe Genişlik Modülasyonu) özelliğidir. PWM bir sinyalin ON ve OFF olma sürelerinin ayarlanması sonucu ortaya çıkan bir sinyal türüdür. Genellikle motor hızını kontrol etmek gibi uygulamalarda kullanılır [1,17,18].

3.2. Timer modülü

Mikrodenetleyici içerisindeki zamanlayıcının görevi tetiklendiği zaman kapasitesi kadar saymaktır. Mikrodenetleyici içerisinde 7 adet timer bulunmaktadır. Bunlar genellikle karşılaştırma ve kesme (interrupt) işlemlerinde kullanılır. 3 tanesi 16 bit, 4 tanesi ise 8 bit olan zamanlayıcılar timer0-timer6 arasında adlandırılmaktadır. Bunlar içerisinden timer1, timer3 ve timer5 ise 16 bit zamanlayıcıya ile 65535 sayısına kadar sayma özelliğine sahiptir. Şekil 3.3a'da timer1, timer3 ve timer5 zamanlayıcılarının blok diyagramı verilmiştir. Diğer zamanlayıcılardan timer2, timer4 ve timer6 ise 8 bit zamanlayıcıya sahip olup 255 sayısına kadar sayma işlemi yapabilmektedir. Şekil 3.3 b'de ise 8 bitlik timer2, timer4 ve timer6 zamanlayıcılarının blok diyagramı verilmiştir. Son olarak ele aldığımız timer0 ise yazılımsal olarak hem 8 bit hemde 16 bitlik zamanlayıcıya sahip olup şekil 3.3 c'de 8 bitlik timer0 blok diyagramı şekil 3.3 d'de ise 16 bitlik timer0 blok diyagramı verilmiştir [1,19,20,21]



(a)

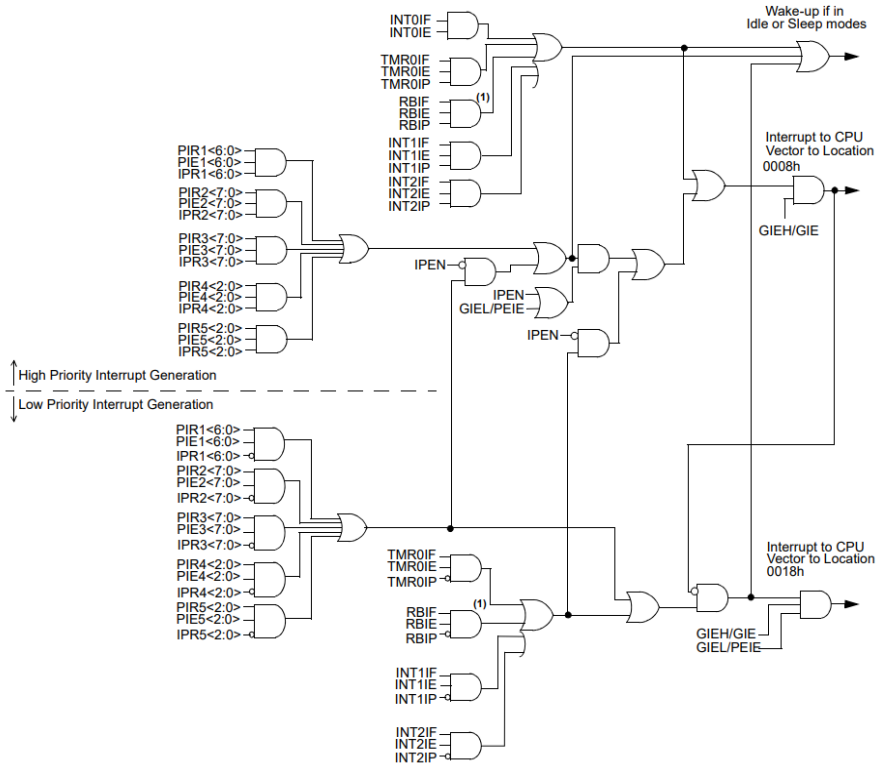


Şekil 3.3. a)16 bitlik timer blok diyagramı b) 8 bitlik timer blok diyagramı c) 8 bitlik timer0 blok diyagramı d) 16 bitlik timer0 blok diyagramı

3.3. Kesme (interrupt) modülü

Kesme (Interrupt) işlemi, bir programın ana akış şemasında alışa gelmiş olan programdan çıkıp anlık olarak başka bir görevi yapmasına denir. PIC18F45K22 mikrodenetleyicisi bünyesinde INTCON, INTCON1, INTCON2, INTCON3, PIR1, PIR2, PIR3, PIR4, PIR5, PIE1, PIE2, PIE3, PIE4, PIE5, IPR1, IPR2, IPR3, IPR4, IPR5 ve RCON olmak üzere 19 adet kesme denetimi yapan komut vardır [1,22-24].

Şekil 3.4'de PIC18F45K22 mikrodenetleyicisinin interrupt modülünün blok diyagramı verilmiştir.

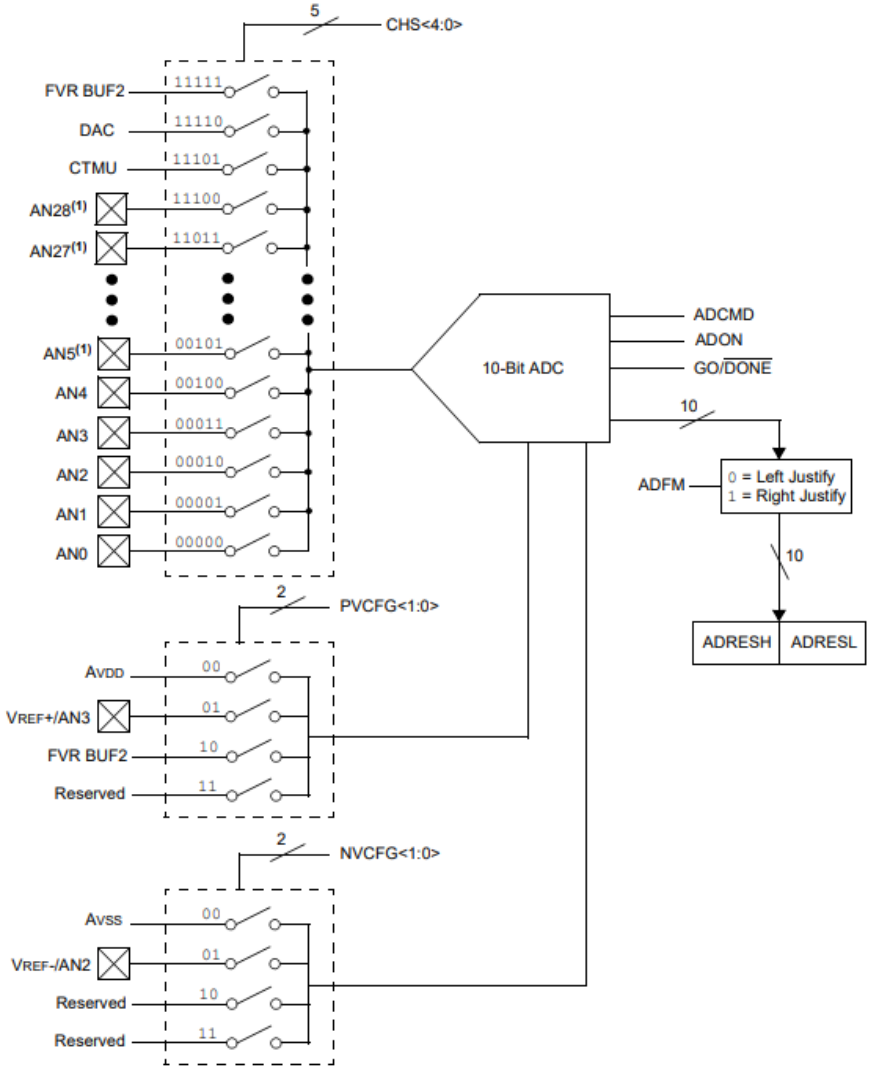


Şekil 3.4. PIC18F45K22 mikrodenetleyicisinin interrupt modülünün blok diyagramı

3.4. Anolog Dijital Dönüştürücü (ADC) modülü

PIC18F45K22 mikrodenetleyicisin bir diğer özelliği ise bünyesinde 10-bit ADC (Anolog-to-digital) bulunur. ADC'ler dış dünyadan sensörler yardımıyla algılanan analog sinyallerin dijital sinyallere dönüştürülmesi işlemidir. 10 bitlik ADC, PIC18F45K22 mikrodenetleyicinin 4 ve 5 numaraları ile temsil edilen RA2 (referans voltajının negatif kutbuna bağlanır) ve RA3 (referans voltajın pozitif kutbuna bağlanır) numaralı pinlere bağlı olan analog referans voltajını referans değer olarak alır. Alınan analog referans değeri 0V=0 ve 5V=1023 olacak şekilde 2^{10}=1024 değere eşit aralıklarla bölünür. Böylece

analog sinyal dijital sinyale dönüştürülür [1,25-28]. Şekil 3.5'de mikrodenetleciye ait ADC blok diyagramı verilmiştir.



Şekil 3.5: PIC18F45K22 mikrodenetleyicinin ADC modül diyagramı

3.5. Osilatör Modülü

PIC18F45K22 mikrodenetleyecisinin içerisindeki programların çalışması için clock (saat) sinyaline ihtiyaç vardır. Mikrodenetleyici için clock sinyalini üreten harici ve dahili osilatör vardır. Mikrodenetleyicide kullanılabilecek 6 tip osilatör vardır. Bunlar;

- RC: Harici direnç-kapasite
- LP: Düşük-güç kristali
- XT: Kristal-rezonatör
- INTOSC: Dahili osilatör
- HS: Yüksek-hız kristal rezonatör
- EC: Harici saat

HS ve EC osilatör tipleri FOSC<3:0> bitini kullanılarak oslatörün hızı kontrol edilir. Birincil saat modları CONFIG1H registerının FOSC<3:0> bitleri kullanılarak seçilebilir. Birincil saat işlemleri için aşağıda verilen bitler kullanılır.

- PRICLKEN (CONFIG1H<5>)
- PRISD (OSCCON2<2>)
- PLLCFG (CONFIG1H<4>)
- PLLEN (OSCTUNE<6>)
- HFOFST (CONFIG3H<3>)
- IRCF<2:0> (OSCCON<6:4>)
- MFIOSEL (OSCCON2<4>)
- INTSRC (OSCTUNE<7>)

OSCCON, OSCCON2 VE OSCTUNE registerları cihazın saat işletiminin kontrolünü yapar. Ana sistem saat seçimi: MSCS bitleri SCS<1:0> ana saat kaynağını seçer. Dahili frekans seçimi ise Internal Oscillator Frequency Select bitleri IRCF<2:0> dahili osilatör bloğunun frekans çıkışını seçer. Birincil saat modları CONFIG1H registerının FOSC<3:0> bitleri kullanılarak seçilebilir.

R/P-0	R/P-0	R/P-1	R/P-0	R/P-0	R/P-1	R/P-0	R/P-1
IESO	FCMEN	PRICLKEN	PLLCFG	FOSC<3:0>			
bit 7							bit 0

3.6. Güç Yönetim Modları

PIC18F45K22 mikrodenetleyicisi yedi (7) farklı güç yönetiminde çalışmaktadır. Güç yönetimleri üç (3) farklı kategoride sınıflanmaktadır. Bunlar;

- Run Modes (Çalışma Modları): Bu çalışma modunda mikrodenetleyici normal çalışmasını yapar. Kullanıcı tarafından yapılandırılan bütün çevresel birimler aktiftir.
- IDLE Modes (Boşta Modları): Bu modda çevresel donanım birimleri çalışmaya devam ederken CPU birimi isteğe bağlı olarak kapatılabilir. Bu çalışma modundan kesme, WDT veya RESET işlemi sonucu çıkılır.
- Sleep Mode (Uyku Modu): IDLEN bitinin sıfırlanması ile bu moda seçilir ve osilatör kapatılır. Kesme, WDT veya RESET işlemi aktif olunca sleep modu devre dışı bıraklır.

Güç yönetim modları arasında geçişler ve mod seçiminde iki (2) farklı etken vardır. Bunlar;

- CPU'ya saat bağlı durumu,
- Saat kaynağının seçimi,

IDLEN biti (OSCCON<7>) CPU saat durumunu kontrol eder ve SCS<1:0> bitleri (OSCCON<1:0>) ise saat kaynağını seçer. Aşağıdaki güç yönetim modlarının nasıl aktif edileceği verilmiştir.

Mode	OSCCON Bits		Module Clocking		Available Clock and Oscillator Source
	IDLEN(1)	SCS<1:0>	CPU	Peripherals	
Sleep	0	N/A	Off	Off	None – All clocks are disabled
PRI_RUN	N/A	00	Clocked	Clocked	Primary – LP, XT, HS, RC, EC and Internal Oscillator Block(2) This is the normal full-power execution mode.
SEC_RUN	N/A	01	Clocked	Clocked	Secondary – SOSC Oscillator
RC_RUN	N/A	1x	Clocked	Clocked	Internal Oscillator Block(2)
PRI_IDLE	1	00	Off	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC
SEC_IDLE	1	01	Off	Clocked	Secondary – SOSC Oscillator
RC_IDLE	1	1x	Off	Clocked	Internal Oscillator Block(2)

Note 1: IDLEN reflects its value when the SLEEP instruction is executed.
2: Includes HFINTOSC and HFINTOSC postscaler, as well as the LFINTOSC source.

Kaynaklar

[1] PIC18F45K22 mikrodenetleyici, Erişim Tarihi: 21/10/2019 https://www.microchip.com/wwwproducts/en/PIC18F45K22#additional-features

[2] Ivey, B. nanoWatt and nanoWatt XLP™ Technologies: An Introduction to Microchip‟ s Low-Power Devices. *AN1267, Microchip*, 2009.

[3] AL-Farsi, H. S. H., & Malathi, B. N. Accident Notification System by using Two Modems GSM and GPS.

[4] Artuğ, N. T. Labvıew ile mikrodenetleyicili bir endüstriyel otomatik sıcaklık kontrol sistemi, 2010.

[5] Vasuja, M., Mishra, A. K., Chauhan, U. S., Chandola, D., & Kapoor, S. (2018, April). Image Transmission Using Li-Fi. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 287-292). IEEE.

[6] Çolak, İ., & Bayındır, R. PIC 16F877 ile DA Motor Hiz Kontrolü. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, *11*(2), 277-285, 2011.

[7] Üstündag, M. PIC Kontrollü Kesintisiz Güç Kaynağı Tasarımı ve Gerçeklemesi. *Technological Applied Sciences*, *3*(1), 151-157. Yüksek lisans Tezi, 2006.

[8] Erem, H., & Altaş, D. İ. H. PIC ile nesne sayım sistemi. Yüksek Lisans Tezi, 2008.

[9] Valdes-Perez, F. E., & Pallas-Areny, R. *Microcontrollers: fundamentals and applications with PIC*. CRC press, 2009.

[10] Beberoglu, E., Tokmakci, M., & Ozdemir, A. T. Is kazalarini onleyebilmek icin bakim/onarim personelinin kullanabilecegi bir giyilebilir emniyet sisteminin tasarlanmasi. *arXiv preprint arXiv:1903.03059*, 2019.

[11] Aziz, A. S. Design of Low-Cost Multi-Waveforms Signal Generator Using Operational Amplifier. *Iraqi Journal of Applied Physics*, *14*(1), 13-18, 2018.

[12] Fadıl, S., Albayrak, S., & Tepe, G. Su Depolama Tanklarında Su Seviye Kontrolünün Kablosuz Olarak Yapılması Wireless Water Level Control Of a Water Storage Tank, 2018.

[13] Koc, C., & Keskin, R. PIC kontrollü aktif bir bum dengeleme sisteminin geliştirilmesi ve modellenmesi. *Journal of Agricultural Sciences*, *17*(1), 2012.

[14] Coşkun, S., Pehlivan, I., Akgül, A., & Gürevin, B. A new computer-controlled platform for ADC-based true random number generator and its applications. *Turkish Journal of Electrical Engineering & Computer Sciences*, *27*(2), 847-860. 2019.

[15] Sarıkaş, A., Ziya, E. K. Ş. İ., Yücelbaş, C., & Buldu, A. A novel SMS application with GSM control on numerator systems. *Turkish Journal of Electrical Engineering and Computer Science*, *22*(1), 97-105, 2014.

[16] Eroğlu, U. B. Ö., & Yüksel, S. Kodlama Eğitiminde Mikroişlemci Uygulamaları. *EJONS*, 2018.

[17] CCP/ECCP Modülleri, Erişim Tarihi: 21/10/2019 http://elektronik hobi.net/ccp-ve-eccp-modulleri/

[18] Barr, M. Pulse width modulation. *Embedded Systems Programming*, *14*(10), 103-104, 2001.

[19] Okur, M., & Şahin, F. Turbo Döngüsel Bir Motorda Döner Valf Ve Elektromanyetik Valf Uygulamalarının Türbin Gücüne Etkisi. *Journal of the Faculty of Engineering & Architecture of Gazi University*, *28*(1), 2013.

[20] Çetin, Ö. 80C51 Mikrodenetleyicilerinde Timer-Counter Yapılarının FPGA Mimarileri Kullanılarak Geliştirilmesi Murat ÇAKIROĞLU1 Ahmet Turan ÖZCERİT1 Halil İbrahim ESKİKURT1.

[21] Timer Kesmesi, Erişim Tarihi: 21/10/2019 https://mikrodunya. wordpress.com/2012/07/11/not-17-timer1-kesmesi/

[22] Çotuk, H. *PIC mikrodenetleyiciler için gerçek zamanlı işletim sistemi* (Master's thesis, TOBB Ekonomi ve Teknoloji Üniversitesi-Fen Bilimleri Enstitüsü-Bilgisayar Mühendisliği Anabilim Dalı), 2008.

[23]http://www.serdaraytekin.com/docs/pl/kesme.html

[24]Handler, I. Explanation of PIC 16F84A processor data sheet--Part 3: Final Overview of PIC.

[25] Walden, R. H. Analog-to-digital converter survey and analysis. *IEEE Journal on selected areas in communications*, *17*(4), 539-550, 1999.

[26] Abualsaud, A. A., Qaisar, S., Ba-Abdullah, S. H., Al-Sheikh, Z. M., & Akbar, M. (2016, December). Design and implementation of a 5-bit flash ADC for education. In *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)* (pp. 1-4). IEEE, 2016.

[27] Tanaka, S., Niitsu, K., & Nakazato, K. A low-power inverter-based CMOS level-crossing analog-to-digital converter for low-frequency biosignal sensing. *Japanese Journal of Applied Physics*, *55*(3S2), 03DF10, 2016.

[28] ADC Kullanımı, Erişim Tarihi: 21/10/2019 https://www.mcu-turkey.com/adc-kullanimi-uzerine/

BÖLÜM 4

4. MİKRO C YAZIM KURALLARI, DEĞİŞKEN TİPLERİ VE UYGULAMALAR

4.1. Noktalama İşaretleri

Mikro C'de kullanılan noktalama işaretleri, (diğer adıyla ayraçlar) köşeli ayraç ("[ ]"), parantez ("( )"), küme gösterimi ayraç ("{ }") nokta, virgül, üst üste iki nokta, eşittir, noktalı virgül ve kesme işareti karakterlerinden oluşmaktadır. Bu karakterlerden çoğu aynı zamanda operatör işlevini de gerçekleştirmektedir [1-3]. Tablo 4.1'de programlama dillerinde kullanılan karakterlerin anlamları ve örnek kullanımları verilmiştir.

Tablo 4.1. Programlamada kullanılan karakterlerin açıklamaları

Karakter	Açıklama	Örnek
[ ]	Dizilerin ifade edilmesinde kullanılır.	short matris[2] [3];
( )	Gruplama, şart ve fonksiyon ifadelerinde kullanılır.	x = a * (b + 2);

{ }	Blok başlangıcı ve bitişinin ifade edilmesinde kullanılır.	if (a==b)
		{
		…
		}

,	Bir grup değişkeni veya parametreyi birbirinden ayırmak amacıyla kullanılır.	short a, b, c;

;	Deyimlerin sonlandırılmasında ve for çevrim kontrol deyimi parametrelerinin ayrılmasında kullanılır.	x = a + b;
		
		for (Başlangıç Değeri ; Şart; Iteratör)
		{ …. }

:	Etiket deyim ifadesinde kullanılır. (Ayrıca ? : operatöründe de kullanılır)	Basla: … Goto Basla;
*	İşaretçi (Pointer) tanımlanmasında kullanılır.	short *x;
=	Atama operatörü olarak kullanılır.	x = a + b;
# ##	Derleyici önişlemci direktifi olarak kullanılır	#define hipotenus(a,b) sqrt(a*a + b*b)

4.1.1. Ayraçlar

Köşeli ayraçlar ("[ ]") tek ve çok boyutlu dizileri ifade etmede kullanılmaktadır. Aşağıdaki örnekte köşeli parantez kullanılarak tek ve çoklu boyutlu diziye ait örnek verilmiştir.

```
char degisken1, str[ ] = "ornek";
int matris[2][3] ;          /* 3 x 4 boyutunda matris*/
ch = str[3] ;               /*  3'üncü eleman */
```

4.1.2. Parantez

Parantezler matematiksel, mantıksal ifadeleri gruplandırmada ve fonksiyon tanımlama da kullanılır. Parantezler ayrıca şartlı durumları belirtme fonksiyon kollarındaki dağılmalarda ve fonksiyon parametre işlemlerinde

kullanılmaktadır. Aşağıdaki örnekte parantez için örnek uygulamalar verilmiştir.

```
a = b * ( c + d ) ;     /* işlem önceliğini belirtir
if (m==n)  ++y ;        /* şartlı durumları belirtir
fonksiyon() ;      /*parametresiz fonksiyonları çağırır
void
fonksiyon2(int a);     /* parametreli fonksiyon
```

4.1.3. Küme Ayraçları

Küme ayraçları { }, bir bloğun başlama ve bitişini belirtmede kullanılmaktadır. Aşağıdaki örnekte koşul ifadelerinde küme ayracının kullanımı ile ilgili örnek verilmiştir.

```
if (t == y)  {
    ++x;
    fonksiyon( );
}

if  (şart)
    { . . . };    /* noktalı virgülün yanlış kullanımı! */
else
    { . . . };
```

4.1.4 Virgül

Virgül (“ , “) satır halinde yazılan değişkenleri birbirinden ayırt etmek veya parametreleri birbirinden ayırmak için kullanılmaktadır. Aşağıdaki örnekte virgül işaretinin kullanımı ilgili örnekler verilmiştir.

```
//değişkenlerin birbirinden ayrılması

void foksiyon(int a, float b, char c);

foksiyon (x , y);    // iki farklı değişken kullanarak fonksiyonu çağırır

// fonkisyonun iki değişken ile çağırılmasına başka bir örnek

fonksiyon((exp1, exp2), (exp3, exp4, exp5));
```

4.1.5 Noktalı Virgül

Noktalı virgül (“ ; “) deyimleri sonlandırma işleminde kullanılır. Eğer bir MikroC ifadesi veya boş bir ifade noktalı virgül ile devam ediyorsa ifade deyimi olarak anlaşılır [5-6]. Noktalı virgül ile devam eden ifade deyimleri hesaplanır ancak sonuç değeri ihmal edilir [7-8]. Aşağıdaki örnekte noktalı virgül işaretinin kullanımı ile ilgili örnekler verilmiştir.

```
k+4;    // k + 4 değeri hesaplanır fakat değer önemsenmez

for (i = 0 ; i < n; i++);
```

4.1.6. İki Nokta Üst-Üste

Etiket olarak kullanılan deyimlerin iki nokta üst-üste (“ : ”) ile bitmesi gerekmektedir. Aşağıdaki örnek bir etiketin iki nokta üst-üste işareti ile tanımlanması verilmiştir.

```
başla:  m = 5;

goto başla;
```

4.1.7. Yıldız İşareti

Yıldız işareti (“ * “) bildirimlerin içinde işaretçi tanımlanmasında kullanılmaktadır. Aşağıdaki örnekte ifadeye işaretçi ataması verilmiştir.

```
char *    char_ptr;    // bir ifadeye işaretçi ataması
```

Yıldız işareti bir diğer görevi ise matematiksel olarak çarpma işlemini gerçekleştirme de kullanılmaktadır. Aşağıdaki örnekte yıldız işaretinin örnek olarak kullanılması verilmiştir.

```
a = b*c;// yıldız işaretinin operatör olarak kullanılması
```

4.1.8. Eşittir İşareti

Eşittir işareti (“=”) ifadelerde atama operatörü olarak kullanılmaktadır. Bunun yanında listelerde değişken tanımlamalarından sonra da kullanılır. Aşağıdaki örnekte hem atama hem de liste oluşturma işlemi için örnek bir kullanım verilmiştir.

```
int test [6] = {1, 2, 3, 4, 5, 6};

int a = 6;

int x, y, z;

x = y + z;
```

4.1.9. Diyez İşareti

Diyez işareti (“#”), ön-işlemci direktifini belirtmek amacıyla kullanılmaktadır. Derleyici anlamına da gelmektedir. Kod oluşumuyla ilişkili olması gerekmektedir. Ön-işlemci tarama esnasında (“#”) ve (“##”) dizgeciklerin başka bir dizgecik ile değiştirilmesinde ve bunların birleşiminde operatör görevi de gerçekleştirmektedir.

4.2. Değişken Tipleri

C programlama dilinde kullanılan değişkenler mutlaka bir değişken tipinde tanımlanması gerekmektedir. Kısaca ifade etmek gerekirse, her bir nesnenin, oluşturulacak fonksiyonunun ve tüm ifadelerinin tipleri derleme sırasında kesin bir şekilde belirlenmiş olması gerekmektedir [9-10]. Değişken tipleri temel görevleri arasında ilk olarak ihtiyaç duyulan belleğin paylaşımında doğru hesaplama yapmak gerekmektedir. Bunun yanı sıra değişkenlere erişim sırasında nesne içerisindeki bit örneklerini doğru analiz etme, tiplerin kontrol edilmesi sırasında geçersiz atama durumlarını tespit etmede kullanılmaktadır.

Tipler nesneler için ayrılacak bellek miktarını belirler. Bunun yanında nesnelerin bellek kısmında var olan bit örneklerinin nasıl yorumlanacağını ve programın bunu nasıl gerçekleştireceğini belirlemede kullanılmaktadır. Kullanılan bir veri tipi, değerler kümesi gibi düşünülebilir. Kullanılan değerler için geçerli işlemler de bu kümeye aittir.

Buna ek olarak derleme-zamanı işlemi için kullanılan sizeof işlemi, standart ya da kullanıcı tarafından belirlenmiş olan tipin boyutunu byte olarak belirlemede kullanılmaktadır. Kullanılacak program ve başlıklar, ne olduğu anlaşılır tam olarak belli tanımlamalar ve tiplerden oluşmalıdır. Bu sayede, mikro C içerisinde oluşturulmuş programda her bir nesneye karşılık bellekte var olan bit örneklerine erişim sağlanabilir ve yorumlanıp istenirse değiştirilebilir [7-8].

4.3.Temel Tipler

Kendinden daha küçük birimlere ayrılamayan tipler temel tipler olarak tanımlanmaktadır. Diğer bir ifadeyle yapılandırılmamış tip olarak ifade edilmektedir. Temel olarak kullanılan tiplerinden bazıları; int, char, void, double ve float’ dır [11].

4.3.1. Aritmetik Tipler

Aritmetik veri tipi belirli anahtar kelimeler kullanılarak oluşturulmaktadır. Int, float, char, void ve double, bu temel tiplere yardımcı olarak önlerine eklenen unsigned, signed, long ve short ile tümleşik tipler ve kayan noktalı tipler oluşturulmaktadır [7].

4.3.2.Tümleşik Tipler

Int ve char tiplerinin farklı türleri ile kullanılması ile tümleşik veri tipini oluşmaktadır. Long, unsigned, signed ve short tipleri int ve char tiplerinin türevleri olarak adlandırılmaktadır. Bu türevler ana tip önüne eklenerek tümleşik veri tiplerini oluşturmaktadır.

Char ve int tiplerine signed ve unsigned türevleri eklenerek tümleşik tipler oluşturulmuştur. Unsigned eklerinin bulunmadığı tümleşik tipler otomatik signed olarak yorumlanmaktadır. İstisnai bir durum olarak char türevsiz kullanımında unsigned olarak yorumlanır. Ayrıca signed ve unsigned türevleri tek başlarına da kullanılması mümkündür. Tek başlarına kullanılmalarında ise varsayılan olarak signed int ve unsigned int kullanılmaktadır. Short ve long türevleri ise yalnızca int tipine örnek olarak kullanılabilmektedir. Short ve long türevleri yalnız olarak kullanıldıklarında int tipine ek olarak kullanılmaktadır. Kısaca, short için short int olarak yorumlanırken, long için long int olarak kullanılmaktadır [11].

4.3.3.Kayan Noktalı Tipler

Kayan noktalı tipler genellikle aritmetik tipler içerisinde kullanılmaktadır. Long ve double ön eklerinin float ve double tipleri ile kullanılması sonucunda kayan-noktalı tipleri oluşmaktadır. Tipler bir nesne için ne kadar bellek ayrılacağını belirlemede kullanılmaktadır. Ayrıca nesnenin depolama bölgesinde bulunan bit örneklerinin program tarafından nasıl yorumlanacağını da belirlemektedir [12]. Tablo 4.2'de aritmetik tipler verilmiştir.

Tablo 4.2 Aritmetik tipler [13]

TİP	*BOYUT*	*ARALIK*
BİT	1-BİT	0/1
SBİT	1-BİT	0/1
(UNSİGNED) CHAR	1 BYTE	0-255
SİGNED CHAR	1 BYTE	-128 +127
(SİGNED) SHORT (İNT)	1 BYTE	-128 +127
UNSİGNED SHORT (İNT)	1 BYTE	0-255
(SİGNED) İNT	2 BYTE	-32768 +32767
UNSİGNED İNT	2 BYTE	0-65535
(SİGNED) LONG (İNT)	4 BYTE	-2147483648 - +2147483647
UNSİGNED LONG (İNT)	4 BYTE	0 - +4294967295
FLOAT	4 BYTE	$-1.5\ 10^{45}$ - $3.4\ 10^{38}$
DOUBLE	4 BYTE	$-1.5\ 10^{45}$ - $3.4\ 10^{38}$
LONG DOUBLE	4 BYTE	$-1.5\ 10^{45}$ - $3.4\ 10^{38}$

4.4. Ek Yazım Kuralları

Değişken isimlerini tanımlamada aşağıdaki kurallara dikkat edilmesi gerekir.

- Değişken atamalarda ya da isimlendirilmelerinde a-z İngiliz alfabesindeki harfler ve 0-9 arasındaki rakamlar kullanılmaktadır. Türkçe karakterler isimlendirme ve değişken atamalarında kullanılamaz [14].
- Değişken atamalarında değişkenin ilk karakteri harf ya da alt çizgi (“_”) olabilir.
- İsimlendirmelerde harflerin küçük yâda büyük olması önemli değildir. Standart C dilinde olduğu gibi küçük-büyük harf duyarlılığı yoktur.
- Mikro C komutlarında olan kelimeler değişken olamazlar [14].

Tablo 4.3'de değişken ismi olarak tanımlanamayacak olan Mikro C komutlarının anahtar kelimeleri verilmiştir [15].

Tablo 4.3 Anahtar kelimeler

· absolute · asm · at · auto · bit · bool · break · case · catch · char · class · code · const · try · typedef · typeid	· continue · data · default · delete · do · double · else · enum · explicit · extern · false · float · for · friend · typename · union	· goto · if · inline · int · io · long · mutable · namespace · operatör · org · pascal · private · protected · public · unsigned · using · while	· register · return · rx · sfr · short · signed · sizeof · static · struct · switch · template · this · throw · true · virtual · void · volatile

4.5. Sabitler

Sabitler, yazılan programın çalışma sırasında belirlenmiş nümerik veya karakter değerini taşıyan ve değişmeyen değerlerdir [14]. Mikro C derleyicisinde aşağıdaki sabit tanımlamaları kullanılmaktadır.

- Tamsayı (integer)
- Kayan nokta (floating-point)
- Karakter (character)
- Karakter dizisi (string)
- Numaralama (enumeration)

4.5.1.Tamsayı Sabitler

Mikro C'de decimal, hexadecimal, binary ve octal tamsayı sabitleri kullanılmaktadır. Tablo 4.4'de tamsayı sabitleri ve örnek kullanımları gösterilmiştir.

Tablo 4.4 Tamsayı sabitleri

Sabit Türü	**Ön Ek**	**Örnek Kullanımlar**	
		Sabit	**Geçerli Veri Tipi**
Binary	0b (0B)	0b01001000	unsigned short
		0b01000001	Short
Octal	0	0777	int
		01001	short
Decimal	…	200	unsigned short
		65	short
Hexadecimal	0x(0X)	0xC367	unsigned int
		0x41	short

4.5.2. Karakter Sabitleri

Mikro C'de karakter sabiti belirlemek için tek tırnak (') işareti kullanılmaktadır. Örneğin 'M','2', '!' ifadeleri ile tek karakterlik sabit tanımlanmaktadır.

4.5.3. String Sabitler

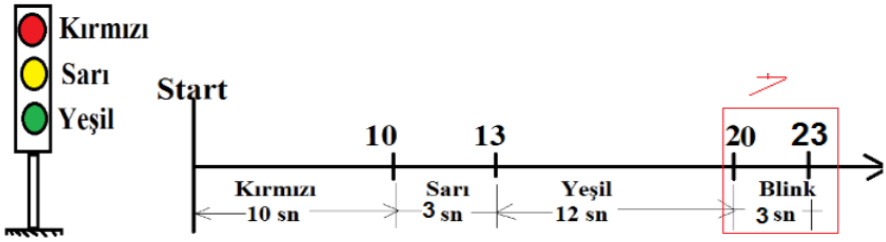
Mikro C'de çift tırnak("") veya karakter sabiti tanımlama da kullanılan tek tırnak (') işareti ile string sabitler tanımlanmaktadır. Tablo 4.5'de metin şeklindeki sabitlerin kullanım şekli gösterilmiştir.

Tablo 4.5. Metin şeklindeki sabitlerin kullanım şekli

lama	Örnek	
	Örnek Kullanım	**Karşılığı**
Çift tırnak (“) ile İfade edilen söz dizimleridir.	“Isparta Üniversitesi”	“Isparta Üniversitesi”
String sabitlerde kaçış karakterlerinden	“Ad \tSoyad”	“Ad Soyad”
Yan yana yazılan stringler birleştirilir.	“Isparta”“Üniversitesi”	“Isparta Üniversitesi”
Ters kesme işareti (\ ) alt satırdan devam edileceği	“Isparta \ Üniversitesi”	“Isparta Üniversitesi”

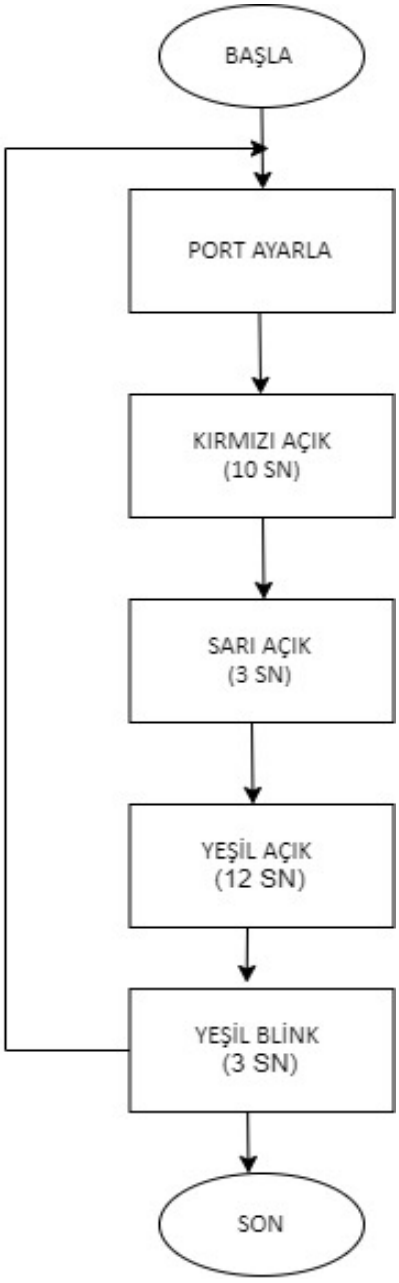
4.6. PIC 18F45K22 Mikrodenetleyici İle Trafik Işığı Uygulaması

Trafik ışıklarının kontrol altına alınması, sürücü ve yayalara gerekli bilgilerin verilmesi, ikazların yapılması için diğer trafik işaretleri ile birlikte kullanılanır. Genel olarak elektrik çeşitli renkli ışıklar ile çalışan trafik işaretlerine ışıklı işaret veya sinyal denilmektedir. Bugün ışıklı işaretlerde kabul edilen renklerden kırmızı durmak gerektiğini, yeşil yolun açık olduğunu, sarı ise işaret sistemine göre durmayı veya harekete hazır olunmasını ifade etmektedir (Şekil 4.1).



Şekil 4.1. Trafik ışığı ledlerin örnek zamana bağlı olarak çalışması

Bir yol kavşağındaki trafik ışığı modeli /akış diyagramı Şekil 4.2’de gösterilmiştir.

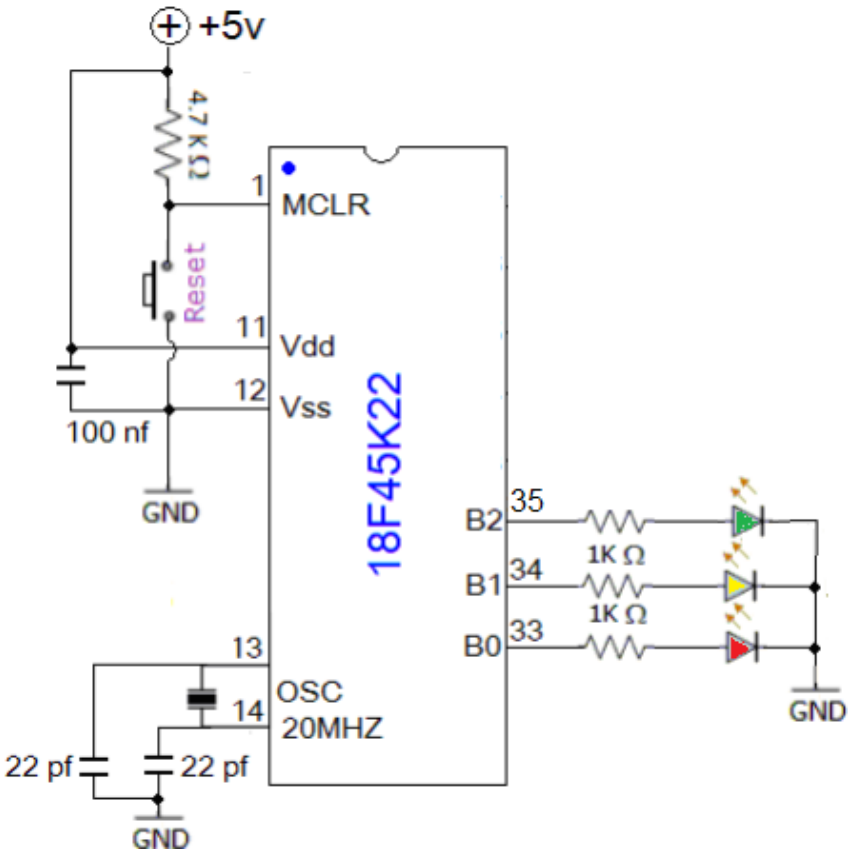


Şekil 4.2. Trafik ışığı uygulamasının akış diyagramı

PIC 18F45K22 mikrodenetleyici ile trafik ışığı uygulaması Mikro C kodları aşağıda verilmiştir.

```
/* TRAFİK IŞIĞI UYGULAMASI*/
void main() {
  TRISB = 0;
    PORTB = 0;
    While(1) {
PORTB = 1; // KIRMIZI LED ON
delay_ms(10000);
PORTB = 2; // SARI LED ON
delay_ms(3000);
PORTB = 4;
delay_ms(12000); // YEŞİL LED ON
PORTB = 0; // YEŞİL LED OFF
delay_ms(500);
PORTB = 4; // YEŞİL LED ON
delay_ms(500);
PORTB = 0; // YEŞİL LED OFF
delay_ms(500);
PORTB = 4; // YEŞİL LED ON
delay_ms(500);
PORTB = 0; // YEŞİL LED OFF
delay_ms(500);
PORTB = 4; // YEŞİL LED ON
delay_ms(500);
   }
}
```

PIC 18F45K22 mikrodenetleyici ile trafik ışığı uygulaması elektrik bağlantı şeması Şekil 4.3'de verilmiştir.



Şekil 4.3. Trafik ışığı uygulamasının mikrodenetleyici bağlantı şeması

Kaynaklar

[1] Simon, J., Szakáll, T., & Čović, Z. Program3ming mobile robots in ANSI C language for PIC MCU's. In Proc. of the 4th Serbian-Hungarian Joint Symposium on Intelligent Systems, SISY,131-137, 2006.

[2] MikroC, Erişim Tarihi 26/10/2019 https://docplayer.biz.tr/55085951-Mikroc-dili-ile-mikrodenetleyici-programlama-ders-notlari.html

[3] Kaçmaz, E. PIC mikrodenetleyici kullanarak ağ bağlantılı gömülü sistem tasarımı iklimlendirme cihaz kontrol ünitesi uygulaması (Master's thesis, Anadolu Üniversitesi). 2007.

[4] Walpa, O. C düzeyindeki mikrodenetleyicileri programlamak için modern mikroC geliştirme ortamı. Modern Elektronik , (6), 64, 2010.

[5] Sungur, C., Terzioğlu, H., & Anadol, M. A. PV Beslemeli PIC Kontrollü Trafik Sinyalizasyonu,

[6] Ahmet, Ö. Z. E. K., & Karal, Ö. PIC Mikrodenetleyici ile Uzak Mesafe Haberleşmesi Kullanılarak Trafik Sinyalizasyon Modellemesi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 10(4), 43-46, 2011.

[7] Arslan, M., Mawlave, W. A., & Altan, Ö. Bilgisayar Kontrollü Kameralı Robot Kolu.

[8] C Dili, Erişim Tarihi: 18/10/2019 https://www.yazilimbilisim.net/c-dili/c-dili-yazim-kurali/

[9] https://stackoverrun.com/tr/q/11653157

[10] https://docs.microsoft.com/tr-tr/cpp/c-language/c-character-constants?view=vs-2019

[11] Özcan, Ö. Ü. C. Hafta 8 C Dilinde Değişken Tipleri ve Temel Giriş/Çıkış İşlemleri

[12] İbrahim, D. PIC Mikrodenetleyici Projeleri C. Temel ve İleri Düzeyde, 2, 2008.

[13]C sharp Erişim Tarihi: 26/10/2019 https://www.wikibooks.org/wiki/C_Sharp_Programlama_Dili/De%C%9Fi%C5%9Fkenler

[14] Programlamaya Giriş, Erişim Tarihi: 26/10/2019 https://www.academia.edu/37713016/Programlamaya_Giri%C5%9F_ve_Algoritmalar_Ders_Notlar%C4%B1

[15] C dili, Erişim Tarihi: 26/10/2019 https://www.yazilimbilisim.net/c-dili/c-dili-yazim-kurali/

BÖLÜM 5

5. MİCRO C DİLİNDE ÖNİŞLEMCİ DİREKTİFLERİ

5.1 Önişlemci (Preprocessor)

Önişlemci, derleme işleminin bir parçasıdır. Derleme işleminin başında önişlemci devreye girer ve program gerçek anlamda derlenmeden hemen önce yapması gereken işlemleri kontrol eder. Önişlemci, program içerisinde "#" karakteri ile başlayan satırlarda verilen işlemleri gerçekleştirir. Diğer bir ifadeyle C programlama dilinde # karakteri ile başlayan satırlar, önişlemci direktifleri olarak kullanılmaktadır. Kısaca, önişlemci, kaynak dosyalarımızı alarak, verilen önişlemci direktiflerine göre yeni dosya oluşturan bir düzenleyici olarak kullanılmaktadır. Ön işlemcinin çalışma şekli;

(stdio.h+benim.h+prog.c) → Önişlemci → C Derleyicisi→ Amaç→Kod şeklindedir.

Ön işlemcinin kütüphaneden almış olduğu dosyaları C derleyecisinde düzenlediği program kodlarına dahil edilerek çalıştırılabilir bir kod haline getirmek için kullanılır [1].

5.1.1 # Operatörü

Mikro C programlama dilinde # karakteri ile başlayan satırlar, önişlemci direktiflerinin çalışması için kullanılır. Önişlemci bir satırda # karakterini gördüğü zaman o satırdaki görevin kendisine ait olduğunu anlar ve o satırda belirtilen işlemleri gerçekleştirir.

#include <stdio.h>

Komut satırı ile # karakteri önişlemcinin devreye girmesini sağlar. Include komutunu gören önişlemci, derleme işlemi başlamadan hemen önce stdio.h (standart giriş/çıkış kütüphanesi) dosyasını bularak kodlarına dahil eder. Sonrasında ise, diğer satırları kontrol ederek işlemleri tamamlayarak derleme işlemine başlar.

5.1.2 ## Operatörü

operatörü sözcüklerin birbirlerine bağlanması için kullanılır. Önişlemci ## komutu ile dizgecikleri birleştirdikten sonra kendi simgesini siler. ## operatörü genellikle tanıtıcıları oluşturmak için kullanılmaktadır. Aşağıdaki iki

örnekte ## operatörünün iki farklı kullanımı ile dizgeciklerin birleştirilmesi gösterilmiştir [1].

#define SPLICE (x,y) çağrısı x ## _ ## y ile , x_y şeklinde,

#define SPLICE (CNT, 2) çağrısı ile cnt_2 şeklinde dizgecikler birleştirilmiştir.

5.2 Önişlemci Direktifleri

Direktif vermek günlük kullanımda talimat vermek, emretmek ve buyurmak anlamına gelmektedir. Micro C programında önişlemci direktifi olarak genellikle #define komutu ve #include komutları kullanılmaktadır. Mikro C programlama dilinde önişlemci direktifi olarak define komutu; sabit ifadeler, metotlar, sorgular ve makro gibi tanımlama işlemlerinde sıkça kullanılmaktadır [2]. Include komutu ise; programa genellikle başlık dosyası eklemek için kullanılmaktadır. Başlık dosyası, standart giriş çıkış işlemlerini içeren bir kütüphane olabileceği gibi, kendimize ait fonksiyonların bulunduğu bir dosya da olabilmektedir [3].

5.2.1 #define komutu

Derleyici, programı derlerken define ifadesi yanında tanımlanmış olan sabiti programda yerleştirir ve o ifade yerine sabiti yerleştirerek programın derlenmesini sağlar [4]. Programda kullanılan sembolik isim atamaları, başta ana program olmak üzere bütün alt programlarda da kullanılmaktadır. Kısaca ifade etmek gerekirse tanımlama bütün fonksiyonlarda kullanılabilir [5]. Define komutu sadece değişken tanımlama işlemlerinde değil aynı zamanda sabit ifadeleri tanımlamada da sıkça kullanılmaktadır. Örneğin;

```
#define PI 3.1415926
```

Yukarıdaki komut satırı ile önişlemci, derleme başlamadan hemen önce tüm program içerisindeki PI ifadelerini ona karşılık gelen 3.1415926 ifadesi ile değiştirmektedir.

Define komutu kullanılırken iki önemli unsura dikkat edilmesi gerekmektedir. İlk olarak tanım ile değer arasında sadece boşluk karakteri kullanılmalıdır. İkincisi ise Mikro C programlama dilinde program kodları yazılırken satır sonuna eklenen “;” işaretinin eklenmemesi gerekmektedir. Tanımlamaların hepsinde büyük harf kullanılması gerekmektedir. Aşağıda verilen iki örnek uygulama ile define komutunun kullanımı ifade edilmiştir.

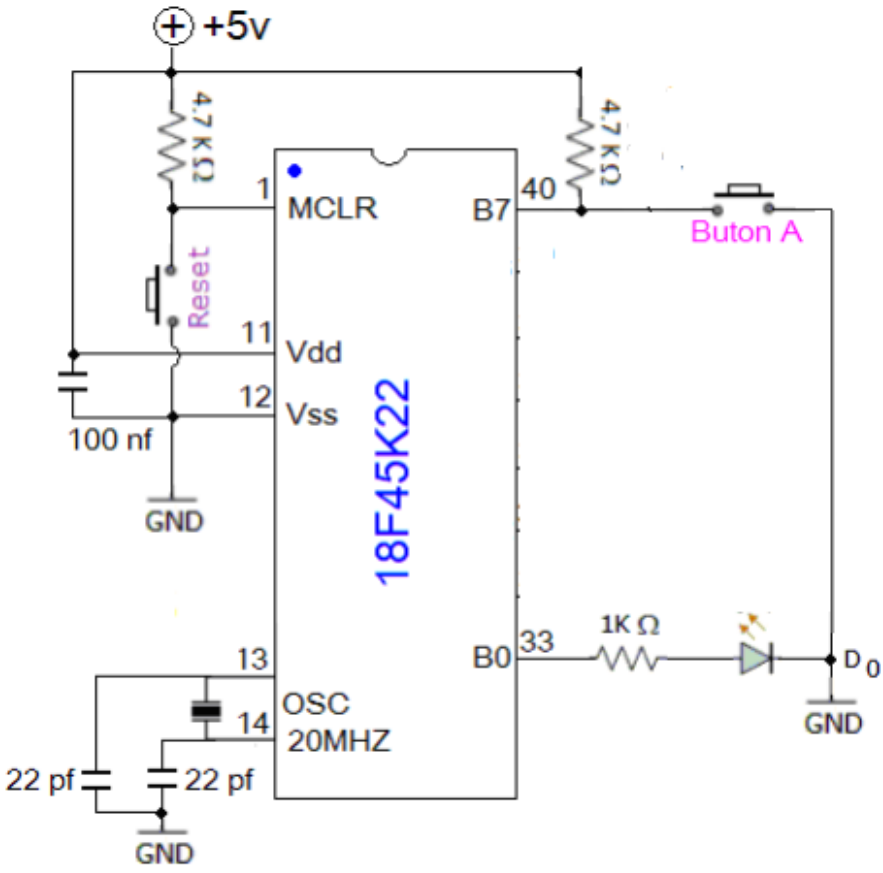
```
#define yak button (&porta,3,100,1)
int sayı=3, a, d=0,
char hiz=234,
void main ()
{
  long deger=123456789;
}
```

#define komutu ile buton elemanın tanımlaması yapılmaktadır. (&, butonun bağlı olduğu port',3 'pin',100 'Butonun basılı tutulması gereken süre','lojik-1 mi?lojik-0 mı? olduğunda aktifleştiği') yazılarak buton tanımlaması yapılır[6].

PIC 18F45K22 mikrodenetleyici ile #define komutu ile ilgili uygulaması Mikro C kodları aşağıda verilmiştir.

```
/* #define komutu uygulaması*/
#define led PORTB.B0      // PORTB.B0 ifadesine led elemanı
atanır.
#define butonA PORTB.B7  // PORTB.B7 ifadesine butonA elemanı
atanır
void main ()
{
     While (1)
     {
          while (butona==1)
               {
                    Led=1;
                    Delay_ms(500);
                    Led=0;
                    Delay_ms(500);
               }
     }
}
```

Uygulamada ilk satırda define komutu kullanılarak PORTB.B7 ifadesi ile led kelimesi eşleştirilmiştir. Böylece tanımlama neticesinde birden fazla pin adlandırılmasına gerek kalmadan led, buton gibi elemanların aktif-pasif durumu program üzerinde kolaylıkla ifade edilmesi sağlanmıştır. Şekil 5.1'de uygulamanın mikrodenetleyici bağlantı şeması gösterilmiştir.



Şekil 5.1. 18F45K22 mikrodenetleyicinin elektrik bağlantı şeması

5.2.2 # include komutu

Önişlemci direktifi olan #include komutu dosyanın içeriğini mevcut kaynak dosyaya eklemektedir. Include komutu tanımlama işlemlerinin yanı sıra, (.h) ile ifade edilen kitaplık dosyalarını tanımlamak içinde kullanılmaktadır. Kütüphane başlık dosyaları açılı ayraçlar (< >) arasına alınarak tanımlanmaktadır. Include komutu önişlemcinin standart başlık dosyalarını sistem dizinleri arasında aramasını sağlamaktadır [7]. Örneğin;

```
#include <stdio.h>
```

komut satırı ile Mikro C kurulum dizini altında stdio.h dosyasını bulunarak Mikro C'deki programa eklenmesi sağlanmaktadır. Mikro C kurulum dizininde yer almayan .h kütüphanelerinin programlama eklenmesi için < > karakterlerinin yerine "" çift tırnak işareti kullanılarak bilgisayar içerisindeki konumu verilmesi gerekmektedir[8].

```
#include "c:\myfile.h\test.h"
#include "header.h" // header dosyası çağırır
Void main ()
{
void Led_yak (unsigned char sure)
{
LED_TRIS = 0; // bu değişken header dosyasında belirtildi.
LED=0; // bu değişken header dosyasında belirtildi.
Vdelay_ms(sure*saniye);  //  bu  değişken  header  dosyasında
belirtildi.
LED = 1;
}
void LED_FLASH (unsigned char sure)
{
int i;
LED_TRIS = 0;
for(i=0;i<yanma_suresi;i++)
{
LED=0;
Vdelay_ms(flash_sure*saniye);
LED = 1;
Vdelay_ms(flash_sure*saniye);
}
}
```

Kaynaklar

[1] Taşbaşı, G. M. (2010). İleri C programlama. Altaş.

[2] Sozluk, Erişim Tarihi: 26/10/2019 https://www.sozluk.gov.tr/

[3]Define Erişim Tarihi: 26/10/2019 https://steemit.com/tr/@etasarim /nilemcibildirimleridefineincluecprogramlamadili-19vcjbrith

[4] C programlama, Erişim Tarihi: 26/10/2019 https://web.itu.edu.tr/ hulyayalcin/MAK104E_Programlama/c_programlama_2.pdf

[5] Taşbaşı, G. M. İleri C programlama. Altaş, 2010.

[6] Sharp define, Erişim Tarihi: 26/10/2019 http://www.sektorharita. com/sharpdefine-onislemci-komutunedir.html

[7] Mikro C, Erişim Tarihi: 26/10/2019 https://www.elektrikport.com/teknik-kutuphane/mikroc-ile-c-programlama-dersleri-1-elektrikport-akademi/8673#ad-image-0

[8] Olsson, M. Variables. In Modern C Quick Syntax Reference (pp. 9-19). Apress, Berkeley, CA, 2019.

BÖLÜM 6

6. DİZİLER

6.1. Giriş

C programlarında kullanılan tanımlı değişkenler belirli sayıdan fazla oldukları zaman, program kodları yazılırken problemler ortaya çıkmaktadır. Bu sorunları azaltmak için program yazılırken dizi kavramı kullanılmaktadır [1]. Dizi, aynı tip verilerin birbiri ardına tanımlanmasıdır. Diziler aynı tip verilere tek bir isimle erişmek için kullanılmaktadır. Bir dizinin bütün elemanları bellekte arka arkaya saklanır. Diziler kullanılarak, aynı isimle birden fazla değişkene erişilerek işlemler gerçekleştirilir [2].

Dizi tanımlarken öncelikle short, char, int, float, double… gibi verinin türü belirtilmesi gerekmektedir. Akabinde dizinin adı ve en son olarak köşeli parantez ([]) ile dizinin eleman sayısı belirtilmelidir. Dizi içerisinde tanımlanan değişkenlerin veri türleri ile aynı olmalıdır. Örneğin; değişkenleri tanımlarken a1, a2, a3 … a300 gibi tek tek tanımlamak yerine int a[300]; olarak tanımlanarak 300 elemanlı bir dizi oluşturulmaktadır.

Dizi elemanlarının tümü int veri türünde olmalıdır. Oluşturulan dizi 300 elemandan oluştuğu için sadece dizi adını programda yazarak istenilen elemana ulaşılması mümkün değildir. Bu nedenle herhangi bir elemana ulaşmak için o elemanın dizideki sayısını da belirtmek gerekir. Örneğin dizinin birinci elemanı daima [0] indeksi ile çağrılmaktadır. Dizinin son elemanı ise (n-1) olarak elde edilmektedir. İndeks dizilerde değişken isminin (dizinin adı) atanmasının ardından köşeli parantez içerisinde belirtilen numara olarak ifade edilir.

Diziler Mikro C programında tablo, matris ve veri depoları oluşturmada oldukça sık kullanılmaktadır. Mikro C programında diziler tek boyutlu ve çok boyutlu olarak tanımlanmaktadır.

6.1. Tek Boyutlu Diziler

Mikro C programlama dilinde aynı veri tipinden olan değişkenler tek bir isim altında toplanarak tek boyutlu diziler oluşturulmaktadır. Aynı veri türünden ve farklı isimlere sahip çok fazla sayıda değişken tanımlamak yerine, dizi bildirimi yapıp tek isim kullanarak aynı sayıda değişken tanımlamalıdır [1].

Tek boyutlu dizi tanımlarken;

```
<Veri türü> <isim> [Eleman Sayısı]
```

Komut yapısı kullanılmaktadır. Bu gösterimde ilk olarak dizi elemanlarının bellekte depolanacağı verilerin tipi, bu elemanlara erişim yapılırken kullanılacak dizi adı ve son olarak köşeli parantezler içerisinde dizinin eleman sayısı belirtilmektedir [3].

Örneğin;

```
short rakam[10]={63,6,91,79,102,109,125,7,127,111};
  0   1   2    3    4    5    6   7   8    9
```

Örnek kod satırındaki dizinin ilk indeksi 0'dır. İndeks sayısı 10 elemanlı bir dizi için 0'dan başlayıp indeks 9 ile sonlanmaktadır. Bununla birlikte dizinin son elemanının indisi dizinin eleman sayısının 1 eksi ile ifade edilmektedir.

```
int tekboyut[5]={1,2,3,4,5} ;

//tekboyut[0]=1
//tekboyut[1]=2
//tekboyut[2]=3
//tekboyut[3]=4
//tekboyut[4]=5
```

6.1.1. Tek Boyutlu Dizilerde Atama

Bir dizinin tanımlanması yapıldıktan sonra, atama işlemcisini kullanarak diziye değer ataması gerçekleştirilebilmektedir [1].

```
int a[50]; // int 50 elemanlı bir dizi oluşturulur.

a[0] = 16; // Dizinin ilk elemanına 16 değeri atanır.

a[17] = 52; // Dizinin 18. Elemanına 52 değeridir.

int sayi[8]={0}; //Sayı dizisinin elemanlarını 0 yapar.
```

Eğer bildirim esnasında ilk değer olarak atanan değerler dizinin eleman sayısından az ise dizinin diğer elemanları varsayılan olarak 0 değerini atanmaktadır.

```
int sayi[8]={1,2,3,4};
```

komut satırı ile bir tanımlama işlem gerçekleştirildiğinde dizi elemanlarının ilk değerleri;

```
int sayi[8]={1,2,3,4,0,0,0,0};
```

şeklinde gerçekleşmektedir.

İlk değer ataması yapılan dizilerin bildirimleri esnasında dizi boyutunun yazılmasına gerek yoktur. Bu durumda derleyici program, dizinin boyutunu

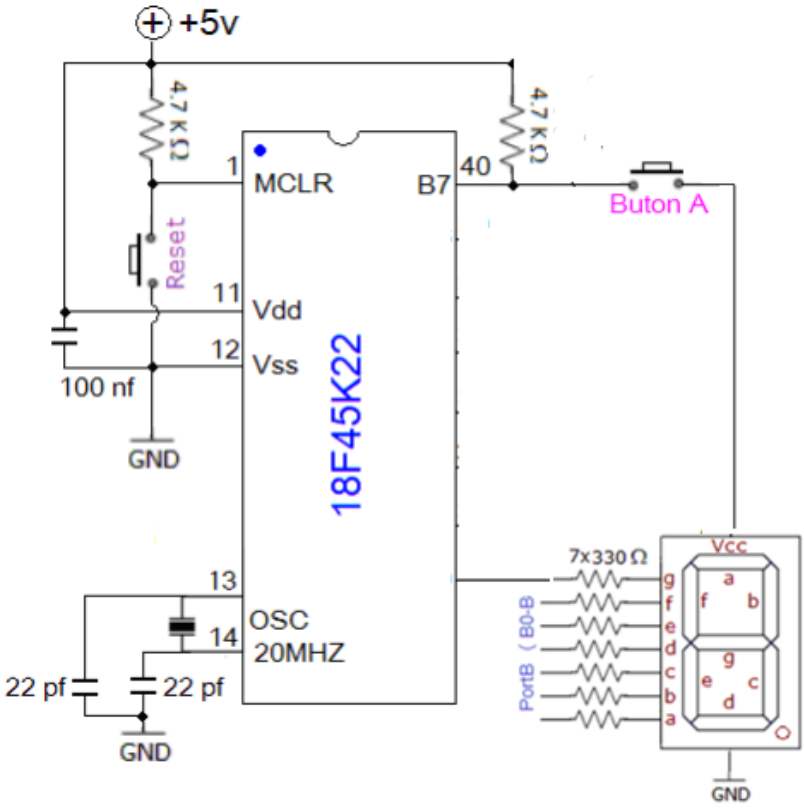
diziye atanan değerlerin sayısından belirlenecektir. Böyle bir durumda dizinin bütün değerlerine atama yapılması gerekmektedir [4].

```
int sayi[]={4,5,6,7,8};
```

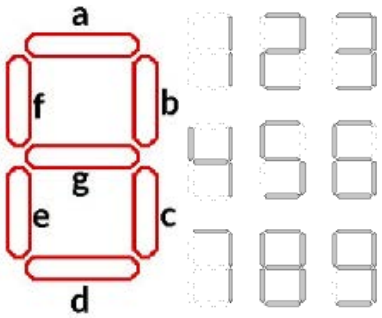
Dizinin tam sayı tipindeki bir değeri, bellekte 4 byte yer kaplarken 50 elemanlı bir dizi bellekte 200 byte yer kaplamaktadır. Aşağıdaki yer alan sıramatik program kodunda tek boyutlu bir dizi yapısı kullanılarak seven-segment bir displayin gösterimi hazırlanmıştır.

```
#define bekle delay_ms(2000)
#define ButonA portB.B7
short rakam[10]={63,6,91,79,102,109,125,7,127,103};
short say=0;
void main()
{    TRISB=128;
     PORTB=63;
     Bekle;
     say=0;
     while(1)
     {
     if (ButonA==1) say=say+1;
     while(ButonA==1) {delay_ms(10);}
     PORTB=rakam[say];
     if (say>9) say=0;
     }
}
```

PIC 18F45K22 mikrodenetleyici ile sıramatik uygulamasının elektrik bağlantı şeması Şekil 6.1'de gösterilmiştir. Ayrıca Şekil 6.2'de 7Segment display gösterimi verilmiştir.



Şekil 6.1. PIC18F45K22 mikrodenetleyici sıramatik devre şeması



Digit Shown	Illuminated Segment (1 = illumination)						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	0	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Şekil 6.2. Seven-Segment Display gösterimi

6.2. Çok boyutlu diziler

Çok boyutlu diziler genellikle Mikro C programında matris tanımlamasında kullanılmaktadır. Çok boyutlu diziler daha çok satır ve sütunlar şeklinde oluşturulmuş veri kümelerindeki çok boyutlu bilgileri veya veri tablolarını saklamak için kullanılmaktadır. Matrisler iki boyutlu dizilerden oluşmaktadır. Dizi tanımlama esnasında ilk verilen indeks satırı ifade ederken, ikinci verilen indeks ise sütunu temsil etmektedir [5].

Çok boyutlu dizinin tanımlanması;

```
<veri türü><isim>[satır][sütun][derinlik]…
```

Şeklinde tanımlanmaktadır. Çok boyutlu bir diziyi tanımlarken, eleman değerini atamak mümkündür. Şekil 6.3'de int A[6][7] dizisinin gösterimi verilmiştir.

A	0	1	2	3	4	5	6
0	$A[0,0]$	$A[0,1]$	$A[0,2]$	$A[0,3]$	$A[0,4]$	$A[0,5]$	$A[0,6]$
1	$A[1,0]$	$A[1,1]$	$A[1,2]$	$A[1,3]$	$A[1,4]$	$A[1,5]$	$A[1,6]$
2	$A[2,0]$	$A[2,1]$	$A[2,2]$	$A[2,3]$	$A[2,4]$	$A[2,5]$	$A[2,6]$
3	$A[3,0]$	$A[3,1]$	$A[3,2]$	$A[3,3]$	$A[3,4]$	$A[3,5]$	$A[3,6]$
4	$A[4,0]$	$A[4,1]$	$A[4,2]$	$A[4,3]$	$A[4,4]$	$A[4,5]$	$A[4,6]$
5	$A[5,0]$	$A[5,1]$	$A[5,2]$	$A[5,3]$	$A[5,4]$	$A[5,5]$	$A[5,6]$

Şekil 6.3. 6x7 boyutlu matrisin tanımlanması.

Çok boyutlu dizilerde dizi elemanlarına ulaşmak için indeks numaraları kullanılmaktadır. Dizide dizinin boyutu kadar indeks numarası bulunmaktadır. Örneğin A çok boyutlu dizisinin 2. satır 3. sütununda ki elemanına ulaşmak için a[2][3] yazılması gerekmektedir [4].

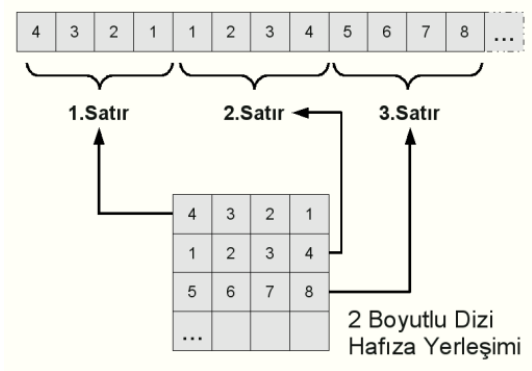
```
cokboyutlu[2][1]=4
//2 x 1 boyutundaki dizinin 3.satır 2.sütun elemanın //değeri 4 olarak atanır.
a[0][1][0]=12;        //Üç boyutlu dizinin satır, sütun ve //derinlik gösterimi.
int[][] dizi = new int[4][6];
// 4 satırı, 6 sütunu olan ve en fazla 24 eleman
// içeren 2 boyutlu dizi adlı int dizisi tanımlandı.
Float Boyut[3][3]={1,2,3,4,5,6,7,8,9};
//3x3 boyutundaki dizinin elemanları
```

Dizi tanımlarken yukarıdaki gibi bir değer atama yapılırsa dizi elemanları Tablo 6.1'de görülen değerleri alacaktır.

Tablo 6.1. 3x3 boyutlu bir dizinin satır ve sütun yerleşimi

Boyu Dizi Elemanları	Sütun 0	Sütun 1	Sütun 2
Satır 0	1	2	3
Satır 1	4	5	6
Satır 2	7	8	9

Girilen değerler sırasıyla belirtilen hücrelere atanır. Şekil 6.4'de görüldüğü gibi bilgisayar dizi elemanlarını hafızada arka arkaya gelen bellek hücrelerine saklanmaktadır [5,6].



Şekil 6.4. 2 Boyutlu Dizinin Hafıza Yerleşimi

Dizi tanımlama yöntemlerinden bir diğeri ise kümele yöntemidir. Aşağıda verilen örnekte 3x3 boyutlu bir dizinin kümeleme yöntemi ile tanımlama işlemi gerçekleştirilmiştir.

```
float boyut[3][3]={{1,2,3},{4,5,6},{7,8,9}};
```

3 satır ve 3 sütundan oluşan float tipinde çok boyutlu bir dizi oluşturulmuştur.

6.3. Karakter dizileri

Karakter dizileri Mikro C programında sözel kelime ve cümle gibi verileri saklamak için kullanılmaktadır. Dizi tanımlaması yapılırken veri türü olarak char [] tipi kullanılmaktadır [6]. Karakter dizileri tanımlanırken;

```
char dizgi_değişkenin_ismi[boyut];
```

```
char a[7] = {'I','S','P','A','R','T','A'};
```

```
char a[8] = "ISPARTA"; //son eleman olarak ‘\n’
```

```
// elemanı eklendiği için eleman sayısı bir fazla belirtilmektedir.
```

Yukarıdaki örnekte ‘/0’ ifadesi null (boş karakter) adıyla adlandırılan değerdir. String ifadenin bittiği anlamına gelmektedir. Bir string genellikle harflerden oluşan karakter dizisini ifade etmektedir [7].

Girilen verileri bellekte saklayabilmek için yeterli büyüklükte bir dizi oluşturulması oldukça önemlidir. Örneğin;

```
char cumle[20];
```

şeklinde yapılan bir tanımlamada kod satırı ile 20 elemanlı bir dizi oluşturularak her bir elemanın bir harfi temsil etmesi sağlanmaktadır. Bu değişkene değer atarken;

```
cumle [] = “merhaba”;
```

şeklinde yazılabileceği gibi,

```
cumle[0] = 'm' ;
cumle[1] = 'e' ;
cumle[2] = 'r' ;
cumle[3] = 'h' ;
cumle[4] = 'a' ;
cumle[5] = 'b' ;
cumle[6] = 'a' ;
cumle[10] = '\n';
```

şeklinde değer ataması yapılabilmektedir.

Değer atama işlemi;

```
char cumle[]={'M','e','r','h','a','b','a','\n'};
```

olarak da yapılabilmektedir.

6.3.1. Karakterlerin hafızada saklanması

Karakter dizilerinin hafızada saklanması ile tek bir karakterin hafızada saklanması arasında önemli bir fark vardır. Karakterler, hafızada ilgili karakterin ASCII kod karşılığı olarak yani bir tamsayı olarak saklanmaktadır ve 1 byte yer kaplamaktadır. Dizilerin en son karakteri daima dizgi sonunu ifade eden NULL ya da ‘ \n ’ karakteridir. Bu karakterin kullanılmasıyla derleyici karakter dizisinin sona erdiğini göstermektedir. Böylece tek bir A harfi hafıza da karakter olarak saklandığında ASCII kod karşılığı olan 65 tamsayısı olarak saklanırken, A dizgisi iki karakter olarak; Bir A harfi bir de NULL karakteri olarak saklanmaktadır. Bu nedenle toplamda 1+1 2 byte yer kaplamaktadır [8].

Kaynaklar

[1] C Programlama, Erişim Tarihi: 26/10/2019 http://www.ibrahimbayraktar.net/2013/12/c-programlama-diziler-arrays.html

[2] Diziler, Erişim Tarihi: 26/10/2019 https://tr.wikibooks.org/wiki/C_Sharp_Programlama_Dili/Diziler

[3]C programlama Erişim Tarihi: 26/10/2019 https://web.itu.edu.tr/hulyayalcin/MAK104E_Programlama/c_programlama_1.pdf

[4] Mikrodenetleyiciler, Erişim Tarihi: 26/10/2019 http://tec.ege.edu.tr/dersler/ileri%20mikrodenetleyiciler%20dersnotu_2014.pdf

[5] Cok boyutlu diziler, Erişim Tarihi: 26/10/2019 http://www.ismailgursoy.com.tr/cok-boyutlu-diziler/

[6] Cprogramlama, Erişim Tarihi: 26/10/2019 http://www.cagataycebi.com/programming/c_programming/c_programming_12.html

[7]http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/c/string.htm

[8] https://www.yazilimgelistiricileri.com/c-karakterler-ve-karakter-dizileri/

BÖLÜM 7

7. YEDİ PARÇALI GÖSTERGE (SEVEN SEGMENT DİSPLAY) İLE 0-99 SAYICI UYGULAMASI

7.1. GİRİŞ

Sayıcılar; zamanlayıcı sistemler, endüstriyel üretimde, depolamada ve daha birçok alanda kontrol ve süreç analizi için kullanılan cihazlardır. Sayıcı sistemler, sistem girişine gelen sinyalleri sayan sistemlerdir. Bu sistemler, ileri-geri sayabilen sistemler olarak çeşitlilik gösterirler. Sayıcı sistemler ile pekçok uygulama gerçekleştirilmektedir. Motor devir hız ölçme, peryot ve frekans ölçme v.b örnekler verilebilir [1]. Sayıcı sistemlerinde gösterge olarak 7 segment led (7-parçalı sayısal Led) kullanılmaktadır.

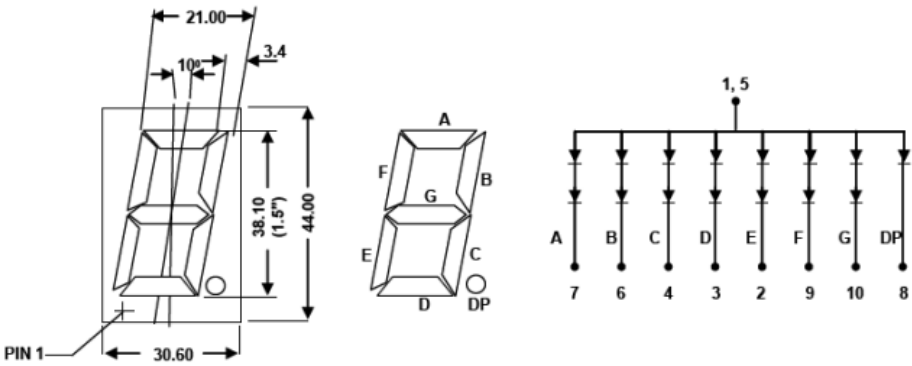
7 Segment Led diye adlandırılan displayler günlük hayatta birçok alanda kullanılmaktadır. Teknolojinin gelişmesine paralel olarak gelişimleri hızla artan seven segment displayler popüler olarak kullanılan elektronik devre elemanlarıdır. Seven segment displayler 1908 yılında icat edilmiştir. Seven segment yapısı üç adet yatayda ve dört adet dikeyde çubuk ledler ile oluşturulmuş 8 rakamını gösteren şekildedir[2,3]. İç yapıda bulunan çubuk LED lerin her birine segment adı verilmektedir. Seven segment göstergeler, sensör ve benzeri elemanlar tarafından toplanan verilerin mikrodenetleyicilerde işlenmesinden sonra somut olarak gösterilmesinde kullanılmaktadır. Seven segment displayler genelde zamanlama amacıyla kullanılan cihazlar, dijital saatler, kol saatleri, radyolar ve hesap makinelerinde sıkça kullanılmaktadır [4].

7.2. Yedi Parçalı Gösterge (Seven Segment Display)

Seven segment displayler, ledlerin parçalar halinde ışık vererek çalışması mantığına dayanan led tabanlı göstergelerdir. Seven segment display'ler 7 tane led den oluşan ve elektronik devrelerde rakamlar ("0,1,2,3,4,5,6,7,8,9") ve bazı karakterleri gösterebilmek için kullanılmaktadır. Şekil 7.1' de seven segment gösterge elemanı verilmiştir.

Şekil 7.1. Seven segment gösterge elemanı

Seven segment displayler, birbirinden bağımsız olarak çalışan yedi bölmeden oluşan elemanlardır. Seven segment displaylerde LED'lerin çok küçük hacimli olması, seven segment displaylerin boyutlarının da küçük olmasının en büyük nedenidir. Küçük hacimli LED'ler belirli ölçülerde bir eleman içerisinde toplanarak seven segment displayleri oluştururlar. Seven segment gösterge elemanının ölçü ve gösterge içerisindeki led bağlantıları Şekil 7.2'de verilmiştir.



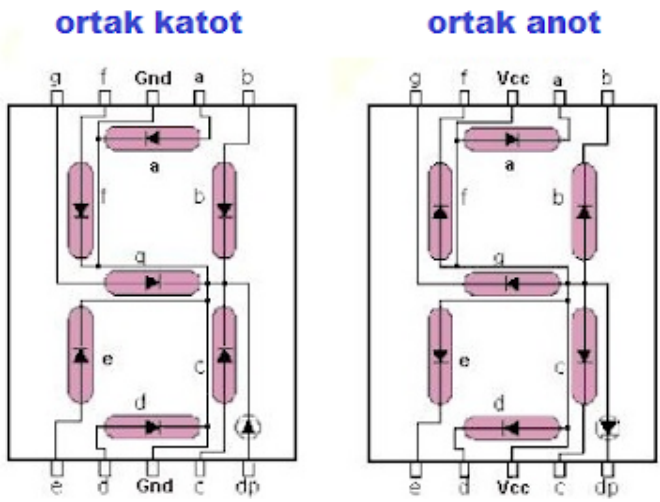
Şekil 7.2 Seven segment gösterge ölçü ve iç bağlantıları

Seven segment displayler içerisinde bulunan ledlerin katotları tek bir bağlantı noktasında toplanmış ise "ortak katot display" olarak adlandırılmaktadır. Eğer, ledlerin anot bağlantıları tek bir noktada toplanmış ise bu displaylere "ortak anot display denilmektedir. Seven segment displaylerin her bir ledine harfler ile isimlendirme yapılmıştır. Ledler sırası ile [a, b, c, d, e, f, g ve dp] isimlerini almaktadır. Seven segment displaylerde bulunan yedi adet ledler

dışında birde sağ alt köşede nokta olarak kullanılan bir led daha bulunmaktadır. Bu led sayısal gösterimlerde ondalıklı sayıların gösterilebilmesi için kullanılmaktadır. Seven segment göstergelerde ortak olan ucun tek olması sorun oluşturabileceği için ve devre elemanı olarak kullanılırken kolaylık sağlaması amacıyla iki adet alt ve üstte olmak üzere pin bulunmaktadır.

Ortak anot seven segment displaylerde ledlerin anot pinleri yani (+) uçları birbiriyle bağlantılıdır. Bu durumda gösterge üzerinde yanmasını istediğimiz led için gerekli uca GND sinyali verilmelidir. Ortak katot displayler de sekiz (8) katot ucu ve iki (2) adet anot ucu olmak üzere 10 pin bulunmaktadır.

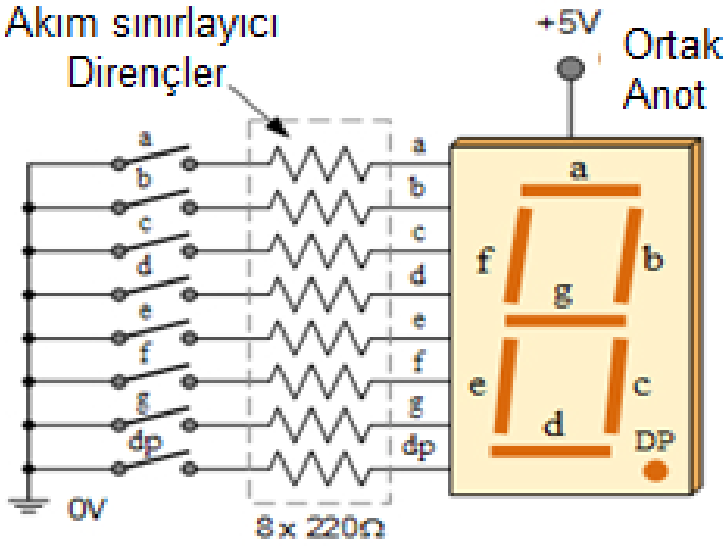
Ortak katot seven segment displaylerde ledlerin anot pinleri yani (-) uçları birbiriyle bağlıdır. Bu durumda gösterge üzerinde yanmasını istediğimiz led için gerekli uca Vcc sinyali verilmelidir. Ortak katot displayler de sekiz (8) tane anot ucu ve iki (2) adet katot ucu olmak üzere 10 pin bulunmaktadır. Ortak anot ve katot seven segment iç yapısı şeması Şekil 7.3'de gösterilmiştir.



Şekil 7.3. Ortak anot ve katot seven segment yapısı [5]

Ledler belirli akımlarda ışık yayan elemanlardır. Seven segment göstergelerde led parlaklıkları uygulanan akım ile doğru orantılı bir şekilde artmaktadır. Ancak ledler belirli bir akım değerinden sonra sınır akım noktasına ulaşarak uygulanan yükü kaldıramaz ve yanarlar. Seven segment displaylerin akım korumalarının olmaması nedeniyle led üzerinden geçen akımın sınırda tutulması gerekir. Seven segment displaylerde akım koruması için pinler

üzerinde harici olarak dirençler bağlanmaktadır. Şekil 7.4'de örnek olarak ortak anot gösterge ile direnç ve anahtar bağlantıları verilmiştir.



Şekil 7.4. Ortak anot seven segment ve anahtar bağlantısı [4]

Yukarıdaki şekilde verilen seven segment devresinde her bir pine anahtar ile direnç bağlanmıştır. Her bir anahtar bağlı olduğu ledi çalıştırmaktadır. Örneğin; seven segment göstergede 5 rakamını görüntüleyebilmek için [a, f, g, c ve d] ledlerinin yanması gerekmektedir. Ortak anot seven segment displayde GND sinyali giriş olarak kullanılmıştır. Ledlerin yanması için gerekli anahtarların bireysel olarak kapatılması gerekmektedir. Dolayısıyla display üzerinde istenilen ledlerin yanması için giriş pinlerine lojik 1 ya da lojik 0 verilmelidir. Bu devrenin elektronik devre ile gerçekleştirilmesi pratik olmayan bir durumdur. Bu durumda seven segment displayleri çalıştırmak için kod çözücüsü ya da sürücüsü olarak bilinen belirli entegreler kullanılmaktadır. Bu entegreler ile seven segment üzerindeki istenilen led yakılabilir. Ayrıca daha fazla sayıda display aynı anda kullanılmak istenirse de 7447 gibi entegreler kullanılmaktadır. Seven segmentlerde istenilen rakamları oluşturacak lojik bilgiler Şekil 7.5'de verilmiştir.

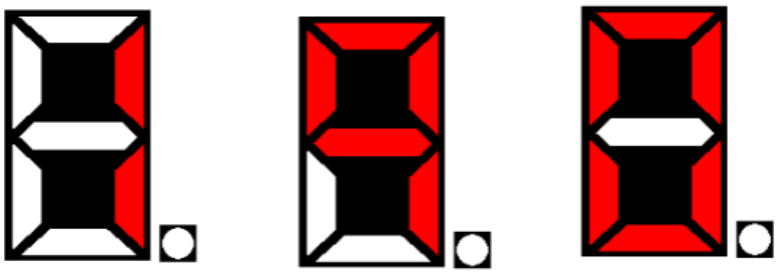
DESİMAL KARŞILIĞI	GİRİŞ				ÇIKIŞ						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	1	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

Şekil 7.5. Ortak katot display lojik tablosu

Şekil 7.5'deki durum incelendiği zaman:

- Display üzerinde 3 rakamını gösterecek ledlerin yanması için için seven segment displaye binary olarak "1111011" bilgisinin gönderilmesi gerekmektedir.
- Display üzerinde 8 rakamını gösterecek ledlerin yanması için için seven segment display'e binary olarak "1111111" bilgisinin gönderilmesi gerekmektedir.
- Display üzerinde 5 rakamını gösterecek ledlerin yanması için için seven segment display'e binary olarak "1011011" bilgisinin gönderilmesi gerekmektedir.

Seven segment göstergelerde oluşan rakamların örnek görüntüsü şekil 7.6'da verilmektedir.



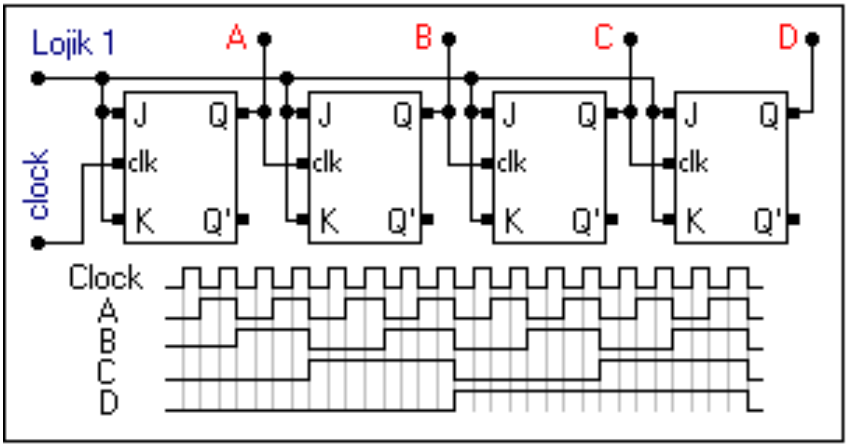
Şekil 7.6. Seven segment örnek çalışma şekli

7.3. Sayıcı Devreleri

Sayıcı devreler ardışık olarak belirlenen bir durumu giriş darbelerine bağlı olarak tekrarlayan devrelerdir. Sayıcı devreleri kontrol sistemlerinde, dijital ölçüm cihazlarında, zamanlama devrelerinde sıklıkla tercih edilmektedir. Sayıcılar asenkron ve senkron sayıcılar olmak üzere ikiye ayrılırlar.

7.3.1.Asenkron Sayıcılar

Asenkron sayıcılar dalgacık veya seri sayıcı olarak isimlendirilmektedir. Asenkron sayıcılarda bulunan flip-flopların tetiklemesi kendinden önceki flip-flop çıkışına bağlıdır. Asenkron sayıcılarda bulunan flip-floplar Şekil 7.7'de gösterildiği gibi seri bağlanmaktadır.

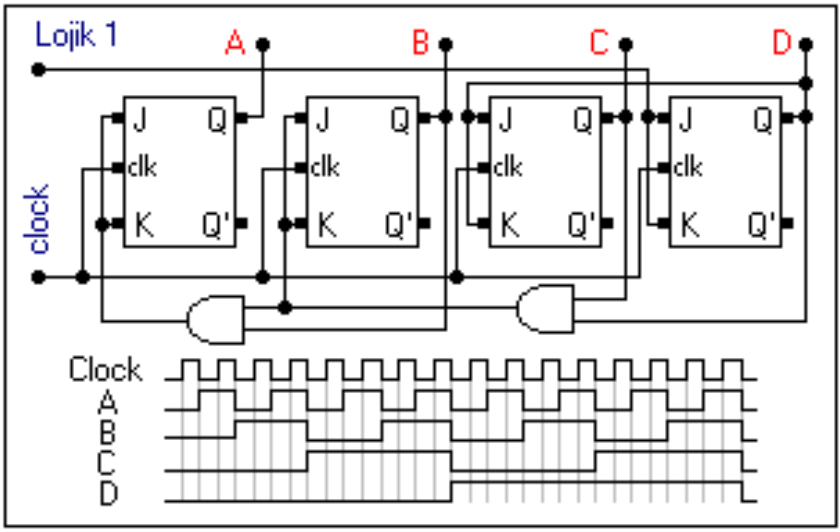


Şekil 7.7. bit (4 çıkışlı) asenkron sayıcı [6]

Flip-floplarda bulunan Q çıkışlar sonraki flip-flop'un "Clock" ucuna bağlanmaktadır. Ancak bu durumun bir dezavantajı vardır, sayıcılarda yavaşlamaya sebep olmaktadır. Şekil 7.7'de devre altında bulunan grafiklerde flip-flopların pals sinyalleri verilmiştir.

7.3.2. Senkron Sayıcılar

Senkron sayıcılarda bulunan flip-flopların clock uçları birbirlerine bağlanmış şekildedir. Flip-flopların hepsi aynı anda clock palsi almaktadırlar. Şekil 7.8'de senkron sayıcı flip-flop devresi verilmiştir.



Şekil 7.8. 4 bit (4 çıkışlı) senkron sayıcı [6]

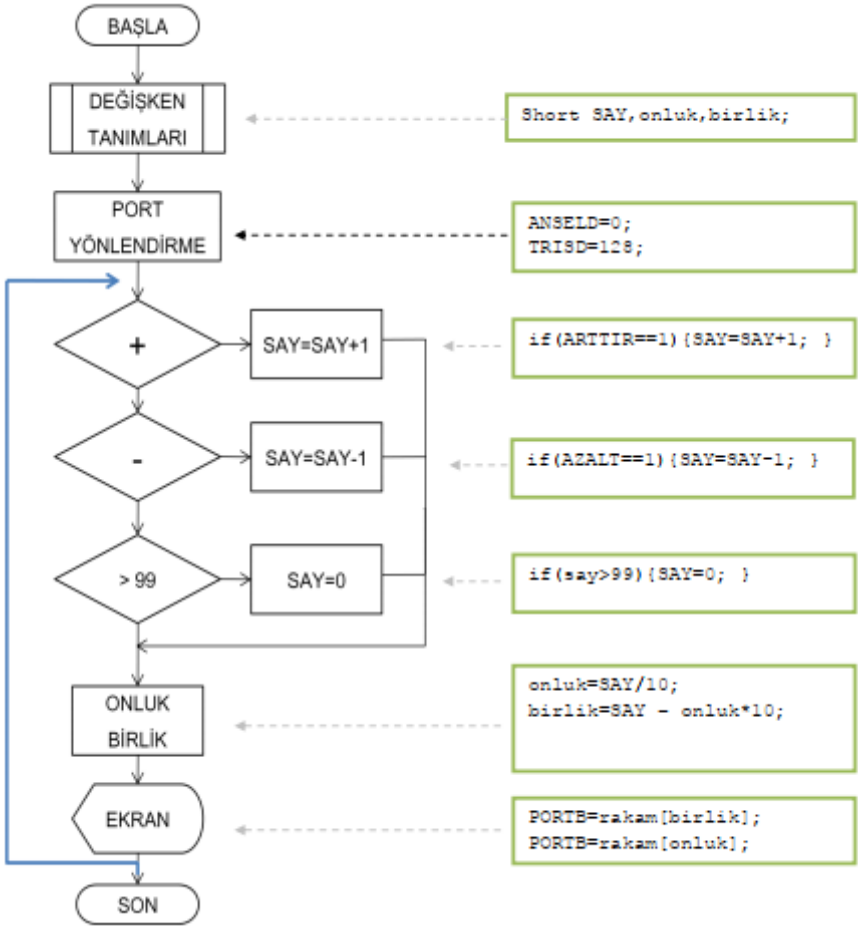
Senkron sayıcıların asnkron sayıcılara göre daha karmaşık bir yapısı bulunmaktadır. Senkron sayıcılarda flip-flop'ların clock uçları birbirlerine bağlanmaktadır. Tüm clock uçlar aynı anda pals sinyali alabilmektedir. Senkron sayıcı devrelerinde pals sinyallerinin aynı anda alınması devreye hız avantajı sağlamaktadır. Şekil 7.8' de devre altında bulunan grafikte flip-flop' ların pals sinyalleri görülmektedir.

7.4. MikroC ile Sayıcı Devresi Uygulaması

PIC18F45K22 mikrodenetleyici ile 0-99 sayıcı devresi gerçekleştirilecektir. Burada 0'dan başlayarak 99'a kadar sayma işlemi yapılacaktır. BUTON A ve BUTON B'yi B portunun, B7 (40) ve B5 (38) pinlerine 4.7 K veya 10K dirençler yardımıyla +5 volta çekerek pin girişlerini PULLUP yapılmıştır. Direnç kullanılmasının iki sebebi bulunmaktadır.

- Butona basıldığında +5V ve GND nin kısadevre olmasını önleme,
- Pinlerin rastgele giriş sinyalleri (gürültü) almasını önlemektir.

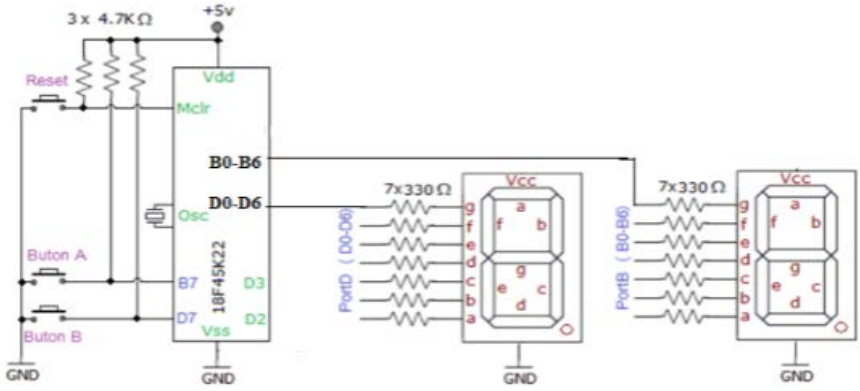
PIC 18F45K22 mikrodenetleyici ile 0-99 arası uygulaması Micro C kodları aşağıda verilmiştir. Ayrıca Şekil 7.9'da akış diyagramı verilmiştir.



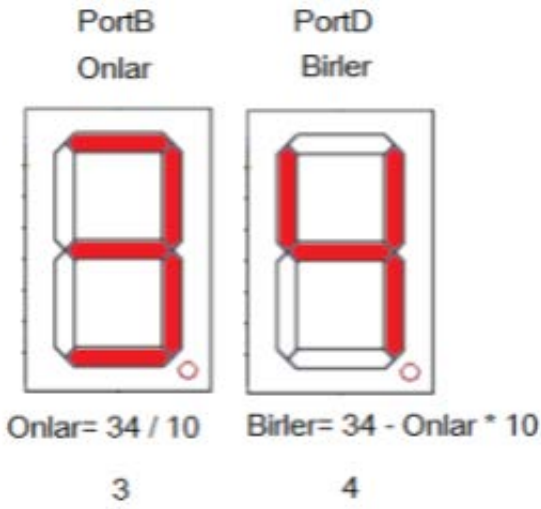
Şekil 7.9. 0-99 sayıcı akış diyagramı

```
/* 0-99 Sayıcı Uygulaması */
#define  ButonA portD.B7
#define  ButonB portB.B7
void main () {
TRISB=128;
TRISD=128;
PORTB=rakam[0];
PORTD=rakam[0];
if (ButonA==1)
say=say+1;
if (ButonB==1)
delay_ms(500);
if (ButonB==1)
say=say-1;
if (ButonA==1)
delay_ms(500);
say=53;
onluk=say/10;//53/10=5.3=5 tam sayı olarak alınır.
birlik=say-onluk*10;   // 53 - 5*10 = 3
PORTD=rakam(onluk);
PORTB=rakam(birlik);
onluk=say/10;
birlik=say-onluk*5;
PORTD=rakam[onluk];
PORTB=rakam[birlik];
if (say > 99) say=0;
if (ButonA==1)
{
     if (ButonB==1)
     say=0;
}
ANSELA=0; // PortA digital ayarlandı.
ANSELD=0; // PortD dijital ayarlandı.
ANSELB=0; // PortB dijital ayarlandı.
}
```

PIC 18F45K22 mikrodenetleyici ile 0-99 sayıcı uygulamasının elektrik bağlantı şeması Şekil 7.10'da gösterilmiştir. Ayrıca Şekil 7.11'de 7Segment display gösterimi verilmiştir.



Şekil 7.10. 0-99 sayıcı devresi



Şekil 7.11. İki haneli sayıların gösterimi

Kaynaklar

[1] Güven, B. Mikrodenetleyicili endüstriyel sayıcı/zamanlayıcı sistemi. 2005.

[2] Borchert, B. Segment-based visual cryptography, 2007.

[3] 7 Segment Display, Erişim Tarihi: 26/10/2019 https://320volt.com/7-segment-led-display-nedir-nasil-kullanilir/

[4] 7 Segment Tutorial, Erişim Tarihi: 26/10/2019 https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html

[5] Sayıcı, Erişim Tarihi: 26/10/2019 https://elektrokod.wordpress.com/2013/12/09/7-segment-display-sayici-uygulamasi/

[6] Sayıcılar, Erişim Tarihi: 26/10/2019 http://sayicilar.8m.net/Tekno.htm

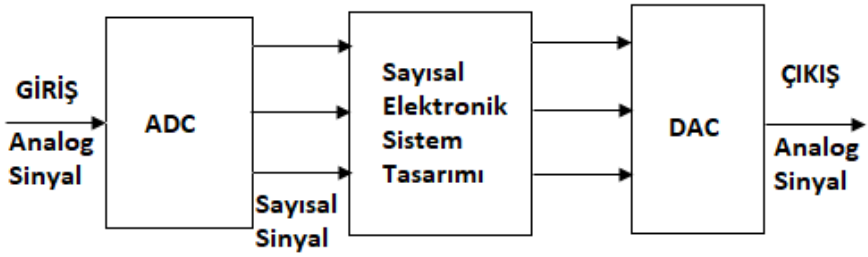
BÖLÜM 8

8. MİKRODENETLEYİ TABANLI ANALOG SAYISAL DÖNÜŞTÜRME (ADC) VE SICAKLIK ÖLÇME VE KONTROL UYGULAMASI

8.1. Giriş

Sıcaklık, günlük hayatta karşımıza her alanında çıkan bir niceliktir. Hastalanınca ateş ölçemede, elektrik su ısıtıcısında sıcaklık ölçmede, ütünün ısı ayarlanmasında, fırınlarda yemeklerin pişirilmesinde, endüstride arabaların boyamasında, demirin ısıl işleminde vb. uygulamalarda sıcaklık ölçümü önemlidir [1-3]. Sıcaklık, ısı, basınç, ağırlık, nem oranı, ışık şiddeti, ses şiddeti gibi büyüklükler analog olarak değişirler [4].

Fiziksel bir büyüklük veri şekline dönüştürülürken analog işaret olarak adlandırılır. Bu analog işaretlerin algılanması ve değerlendirilmesi, ancak insanoğlu tarafından mümkünken bilgisayarlar ve mikroişlemciler tarafından mümkün değildir. Dijital sistemlerin dış dünya ile bağlantılarını sağlamak için ölçülen fiziksel büyüklüklerin dijital sistemin anlayabileceği sayısal değerlere dönüştürülmeleri gerekir. Analog bilgiyi sayısal değerlere dönüştüren elemanlara analog dijital çevirici (ADC Analog Dijital Converter) adı verilir [5] (Şekil 8.1).



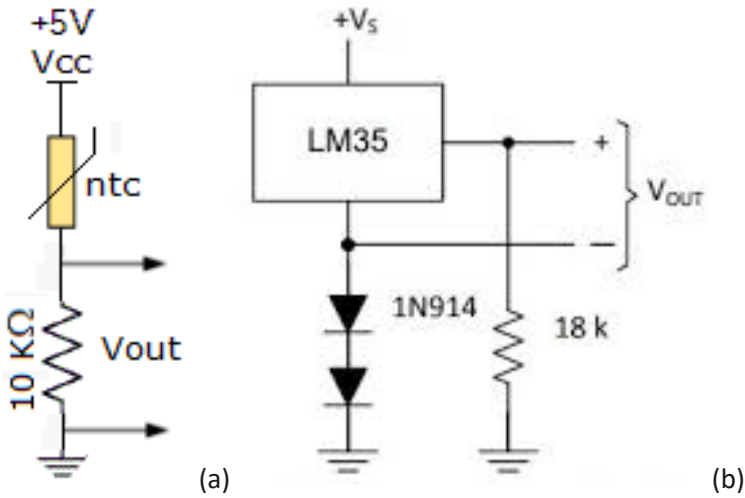
Şekil 8.1. Analog Sayısal Dönüşüm

8.2. Analog Sayısal Dönüştürme (ADC)

Elektronik analog elektronik ve sayısal elektronik olmak üzere ikiye ayrılır. Analog elektronik genellikle fiziksel ortamlardan veri toplamak amacıyla kullanılan sensörlerde sinyal dönüşümleri ve algılamada kullanılır. Sıcaklık, basınç, ışık şiddeti, nem gibi fiziksel değerlerin ölçülmesi, transdüserler yardımıyla elektrik gerilimlerine dönüştürülmesidir.

Sıcaklık ölçme işleminde PTC, NTC gibi direnç değişimine dayalı sıcaklık sensörleri kullanılır. Otomobil soğutma suyu sıcaklığı ölçümü (Hararet Sensörü), buzdolabı, klima gibi cihazlarda ntc kullanılır. NTC'de sıcaklık değişiminde direnç değişimi olur. Elektronik cihazlarda kullanabilmek amacıyla direnç değişiminin gerilime dönüştürülmesi gerekir. Şekil 8.2a'da seri bağlı direnç ile sıcaklık değerinin Vout olarak gerilime dönüştürülmesi gösterilmiştir.

Şekil 8.2b'de LM35 sıcaklık entegresi kullanılarak sıcaklık değeri Vout olarak gerilime dönüştürülmüştür. Vout analog sinyal olup 0V (GND) Vs (+5v) arasında sıcaklığı bağlı olarak Vout gerilim değeri alır.



Şekil 8.2. a) NTC sıcaklık sensörü b) LM35 sıcaklık sensörü

Analog sinyalleri mikrodenetleyiciler yardımıyla okumak ve ALU (Aritmetik mantıksal) işlemler yapabilmek için dijital sinyallere dönüştürülmesi gereklidir. Sayısal sinyaller bit olarak tanımladığımız 1 ve 0 lardan oluşur. Bir bit yalnız başına elektrik devresinden akımın geçmesi (1) veya geçmemesi (0) olarak tanımlanabilir. Diğer bir ifadeyle devredeki led lambanın ışık vermesi (1), vermemesi (0) olarak tanımlanabilir.

8 bit'in yan yana gelmesi ile oluşan ifadeye byte denir. Bu bölümde ADC dönüşümünü açıklamak için; LM35 sıcaklık sensörü ele alınmıştır. LM 35 sıcaklık sensörü, ölçmüş olduğu sıcaklık değerini gerilime dönüştürmektedir. Dönüştürme işleminde her 1 ºC de 10mv luk bir gerilim artışı oluştur. Analog

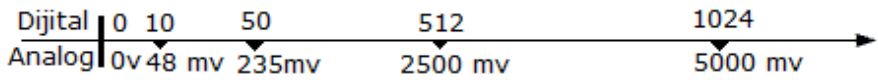
sinyalin dijital sinyale dönüştürülmesi için, ADC devreleri (ADC0804 vb) ile gerçekleştirilmektedir. Ancak bazı mikrodenetleyiciler de ADC dönüşüm işleminini gerçekleştiren dahili birimleri bulunmaktadır.

Örneğin LM 35 Vout çıkışında ölçülen 235 mv, 235mv / 10mv = 23.5 ºC olarak sıcaklık değeri bulunur. Vout çıkışı 10'a bölünerek ortam sıcaklığı bulunur. Vout analog bir değerdir.

Başka bir uygulamada, 235mv analog sinyalini, 10 bit ADC girişi bulunan mikrodenetleyici kullanarak dijital sinyale çevirme işleminde mikrodenetleyicinin A portunun A0 pini kullanılmıştır. Mikrodenetleyiciye maximum ölçüm değeri için Vref sinyali vermektedir. Varsayılan olarak 5V = 5000mV olarak kabul edilmektedir.

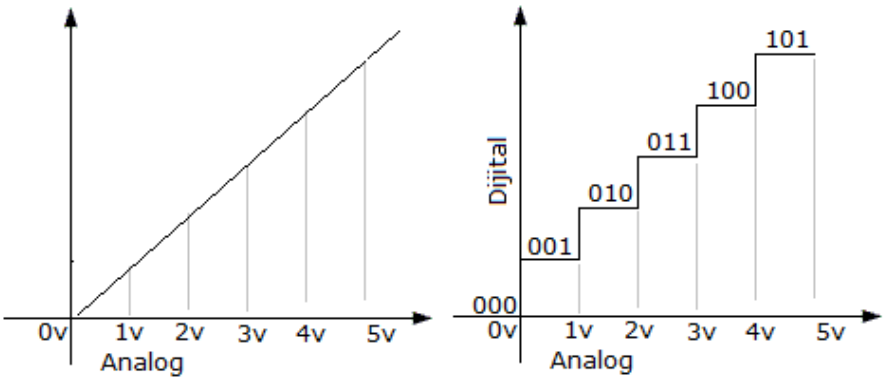
- Ölçüm aralığımız 0V- 5000mV
- Çözünürlük = 5000 / 1024 = 4.8 mV ölçebileceğimiz en küçük değerdir. Bu değer kayan noktalı işlemler yapmamak için 5mv olarak almak işlemde kolaylık sağlamaktadır.
- Dijital Sinyal= Analog Sinyal / Çözünürlük = 235mv / 5mv =48.9=50 olarak bulunur.

Şekil 8.3'de bazı değerin analog ve dijital değerleri verilmiştir. Ölçülebilen en büyük değer +Vref (5000mv) 2^10=1024'e karşılık gelmektedir.



Şekil 8.3. LM35 sıcaklık sensörün analog ve dijital değerleri

Şekil 8.4'de 0-5 Volt arası analog sinyalin sayısal sinyale dönüştürülmesi ve binary kodlanması gösterilmiştir. Örneğin, 3V analog sinyalin karşılığı 011 dijital sinyal olarak kodlanmaktadır. Aradaki değerler çevirme işlemi sırasında kaybolur. Örneğin (3.5V)



Şekil 8.4. Analog ve sayısal sinyal ve kodlama

8.3. Sıcaklık Ölçme Ve Kontrolü

Sıcklık ve kontrol işlemlerine, sera sıcaklık ölçüm ve kontrolü, kalorifer kazanı sıcaklık ölçüm ve kontrolü, otomobil motor suyu sıcaklığı ölçüm ve kontrolleri gibi örnekler verilebilir. Aşağıdaki uygulamada LM35 sıcaklık ölçüm sensörü kullanılmıştır. Bu sensörün temel özelliği sıcaklık başına 10mv gerilim üretmesidir. Şekil 8.5'de LM23 pin bilgileri verilmiştir. Vout ölçülen sıcaklığın gerilim olarak çevirilir.

Şekil 8.5. LM35 Sıcaklık Sensörü bağlantısı

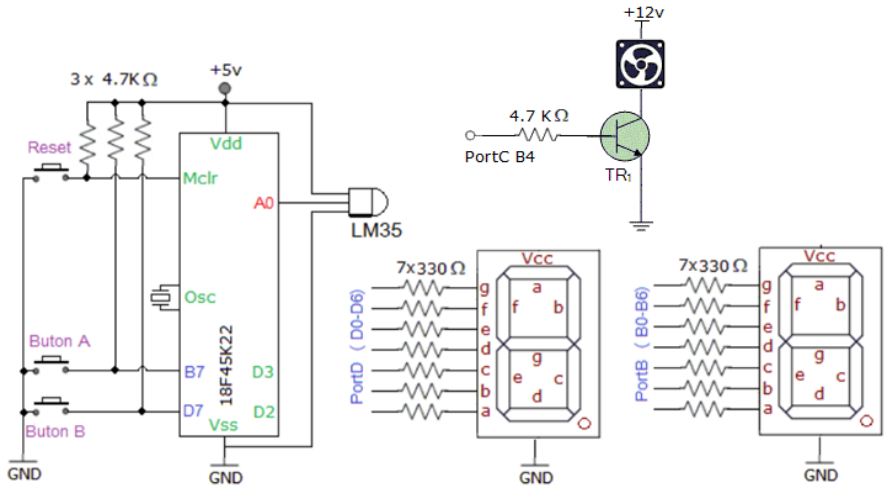
8.4. PIC 18F45K22 Mikrodenetleyi İle Sıcaklık Uygulaması

18F45K22 PIC Mikrodenetleyicide 1 adet 10 bit ADC(analog dijital dönüştürücü) bulunmaktadır. A portu A0, A1, A2, A3, A4, A5 pinleri kullanılır. B, C, D, E portları kullanılarak 30 ADC girişi kullanılabilir. PIC 18F45K22 mikrodenetleyici ile LM35 sıcaklık sensörü ile termometre MikroC uygulama kodları aşağıda verilmiştir.

```
/* 0-99 DERECE TERMOMETRE */
// ONLAR HANESİ PORTD abcdefi-->d0,d1,d2,d3,d4,d5,d6
// BİRLER HANESİ PORTB abcdefi-->b0,b1,b2,b3,b4,b5,b6
#define ARTTIR PORTB.B7
#define AZALT  PORTD.B7
#define FAN    PORTC.B4
// Rakam tanımları hexadecimal olarak verilmiştir.
Short rakam[10] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7C,0x07,0x67};
  short SET=20; // SET DEĞERİ
  short onluk;
  short birlik;
  int oku;
  int sicaklik;
  short ayarla=0;
void main() {
  ANSELA=255; // A PORTU TÜMÜ ANALOG YAPILDI
  ANSELB=0; // B PORTU TÜMÜ DİJİTAL YAPILDI
  ANSELC=0; // C PORTU TÜMÜ DİJİTAL YAPILDI
  ANSELD=0; // D PORTU TÜMÜ DİJİTAL YAPILDI
  TRISB=128; // PORTB.B7 GİRİŞ DİĞER PİNLER ÇIKIŞ
  TRISC=0; // PORTC ÇIKIŞ
  PORTB=0;
  PORTC=0;
  TRISD=128; // PORTD.B7 GİRİŞ DİĞER PİNLER ÇIKIŞ
  PORTD=0;
  ADC_Init();
      while(1)
   { //ayar bölümüne git
          if((ARTTIR==1) && (AZALT==1))
            {
              ayarla=1;
              delay_ms(1000); //ayar bölümünden çık
            }
           while(ayarla)
            {
              if((ARTTIR==1) && (AZALT==1))
            {

       ayarla=0;delay_ms(1000); //ayar bölümünden çık
```

```
        }
   if(ARTTIR==1)
     {
      SET=SET+1;
      delay_ms(300);
     }
   if(AZALT==1)
     {
     SET=SET-1;
     delay_ms(300);
     }
   if(SET<0)
     {
     SET=0;
     }
     if (SET>99)
       {
       SET=0;
       }
      onluk=SET/10; //ONLUK HANESİNİ HESAPLA
      birlik=SET-onluk*10; // BİRLİK HANESİNİ HESAPLA
      PORTB=rakam[birlik]; // SEGMENTE YAZDIRILMASI
      PORTD=rakam[onluk];
 }
  delay_ms(1000);
  oku=ADC_Read(0); //A0 CH0 KANALINDAN ANALOG SINYAL OKU
  oku=oku*5; // Giriş gerilimini hesapla
  sicaklik=oku/10; //LM35 gerilim sicaklık dönüşümü
  onluk=sicaklik/10; //ONLUK HANESİ RAKAMINI HESAPLA
  birlik=sicaklik-onluk*10; //BİRLİK HANESİ HESAPLA
  PORTB=rakam[birlik]; //RAKAMLARIN SEGMENTE YAZDIRILMASI
  PORTD=rakam[onluk];  //KONTROL BÖLÜMÜ
  if (sicaklik>=SET)
    {
    FAN=1;
    }
    else
    {
    FAN=0;
    }
}
}
```

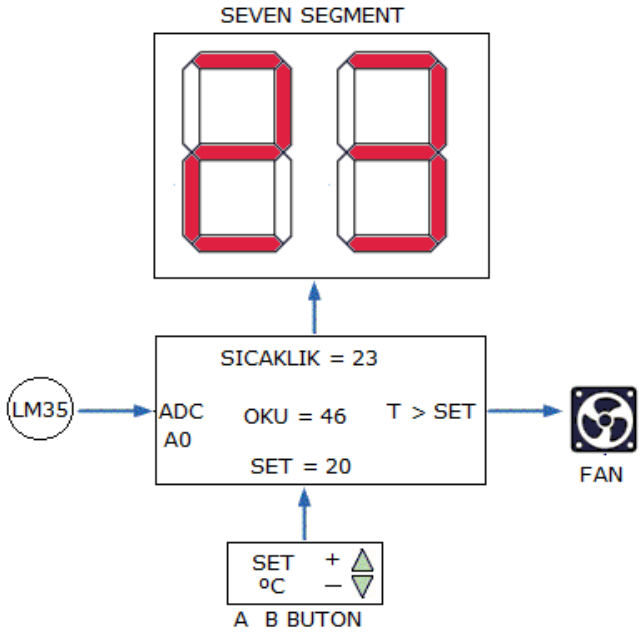
18F45K22 mikrodenetleyici kullanılarak ortam sıcaklığının LM35 sıcaklık sensörü ile okunması ve kontrolüne ait devre şeması Şekil 8.6'de verilmiştir.



Şekil 8.6 LM35 sıcaklık ölçümü devre şeması.

Burada, devre çalışması adım aşamaları aşağıda verilmiştir.

- LM35 ortam sıcaklığını gerilim olarak Vout olarak çıkışa verir.
- 18F45K22 nin A0 analog (ADC) girişinden Vout gerilimi okunarak sayısal (dijital) değere dönüştürülür.
- Ölçüm sonucu led seven segmentlere 0-99 arasında yazdırılır.
- Set (istenen ortam sıcaklığı) değeri A ve B butonları yardımıyla belirlenir.
- Set ve ölçüm sonucu sürekli karşılaştırılarak, set değerini aşması durumunda soğutma fanı çalışır (Şekil 8.7).



Şekil 8.7. LM35 sıcaklık ölçüm ve kontrolü blok şeması

Kaynaklar

[1] Megep Ders notu, Sıcaklık Ölçümü, Erişim Tarihi: 26/10/2019 http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/S%C4%B1cakl%C4%B1k%20%C3%96l%C3%A7%C3%BCm%C3%BC.pdf

[2] Küçükkömürler, A, Taşdelen, K. Mikrodenetleyici Kontrollü, Termoelektrik Kaynaklı Kablosuz ve Sensörsüz Yüksek Sıcaklık Duyum Devresi. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 16.3: 255-259, 2014.

[3] Çayırlıoğlu, İ., Görgünoğlu, S. Mobil telefon ve PIC mikrodenetleyici kullanarak uzaktan esnek kontrol sağlanması. Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, 2.1: 23-27. 2010.

[4] ADC, Erişim Tarihi: 26/10/2019 http://ee.tek.firat.edu.tr/sites/ee.tek.firat.edu.tr/files/LJ1B5_%20ADC_DAC_2.pdf

[5] Mikrodenetleyici Analog Sistemler, Erişim Tarihi:26/10/2019 http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Mikrodenetleyici%20Ile%20Analog%20%C4%B0%C5%9Flemler.pdf

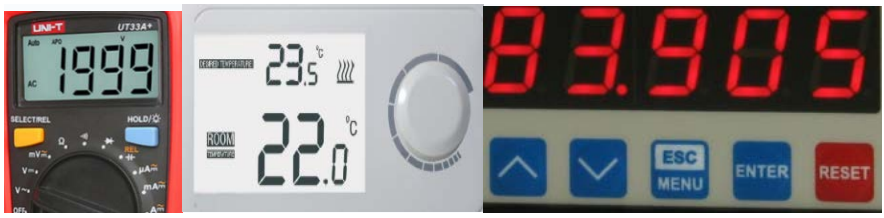
BÖLÜM 9

9. MİKRODENETLEYİCİ İLE LCD TEXT 2x16 YAZI YAZDIRMA VE DİJİTAL SAAT

9.1. Sıvı Kristal Ekran (Liquid Crystal Display/LCD)

LCD; Liquid Crystal Display kelimelerinin kısaltması "sıvı kristal ekran" anlamındadır. Bu kristal görüntülerin her birine piksel adı verilmektedir. Bu piksellerin birleştirilmesi ile de gerçek bir görüntü elde edilmektedir [1]. LCD, elektrikle kutuplanan sıvının, ışığı tek fazlı geçirmesi ve önüne eklenen bir kutuplanma filtresi ile de gözle görülebilmesi ilkesine dayanan bir görüntü teknolojisidir [2].

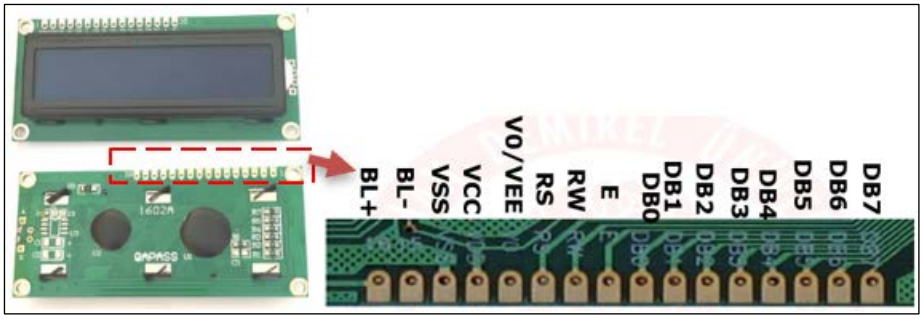
LCD paneller ilk olarak 1960'lı yıllarda kullanılmaya başlanmıştır. Günümüzde ise, Şekil 9.1'de görüldüğü gibi ekran üzerinde sadece harf ve rakamlar yazdırılabilir olduğu için genellikle basit elektronik devrelerde ve enerji kaynaklarının kısıtlı olduğu uygulamalarda (ölçme aletleri, saatler, sayaçlar, termostatlar vbg.) kullanılmaktadır [3].



Şekil 9.1. LCD Örnek Kullanım Alanları

9.2. PIC Mikrodenetleyici ile LCD Kullanımı

LCD panelleri, dijital gösterge (saat, tartı, kronometre, skorboard v.b) panellerinde ya da otomasyon sistemlerinde sensör verilerini (sıcaklık, basınç ve nem vb.) gözlemlemek için kullanılır [4]. Kullanılan LCD modelleri karakter ve satır sayısına göre ifade edilmektedir. Bunlar 16×1, 16×2, 32×2, 20×2, 40×2, 128×64, 240×128 boyutlarında olmaktadır. Bunlar arasında robot projelerinde yaygınlıkla 2x16 boyutlarındaki LCD paneller kullanılmaktadır. Şekil 9.2'de16x2'lik bir LCD modülün şekli verilmiştir.



Şekil 9.2. 16x2 LCD Modül

9.2.1. LCD Panel Bacakları ve Bağlantı Şekilleri

LCD panellerin çoğunda tek sıra halinde 16 pin bulunur. Bu pinlerden ilk 14 tanesi kontrol için son iki tanesi ise eğer varsa arka ışık için kullanılmaktadır [5].

VSS: LCD panelin şase ucudur. Şase bağlantısı Vss ucu ile gerçekleştirilmektedir.

VDD: LCD panelin besleme ucudur. Besleme gerilimi (+5V) VDD bacağından verilmektedir.

VEE veya **V0**: LCD panelin kontrast ayarının yapıldığı ucudur. Bu kısma bir potansiyometre bağlanarak ekranın ışık ayarı yapılmaktadır.

RS: Regester Select ifadesinin kısaltılmış halidir. Bu data ucu ile denetleyiciden gönderilen bilgilerin hangi port ile göndereceği belirlemede kullanılmaktadır.

RW: Read Write ifadesinin kısaltılmış halidir Gönderilen bilgilerin yazıldığı data ucudur. Bu bacak bağlantısı yardımı ile LCD panele yazılacak karakterler gönderilmektedir.

E: Denetleyici ile LCD panel arasında bir veri alış verişi gerçekleştirileceği zamanda denetleyici Enable bacağının bağlı olduğu portu aktif konuma getirerek veri transferini sağlamaktadır.

D0/D7: Bu uçlar veri uçlarıdır.

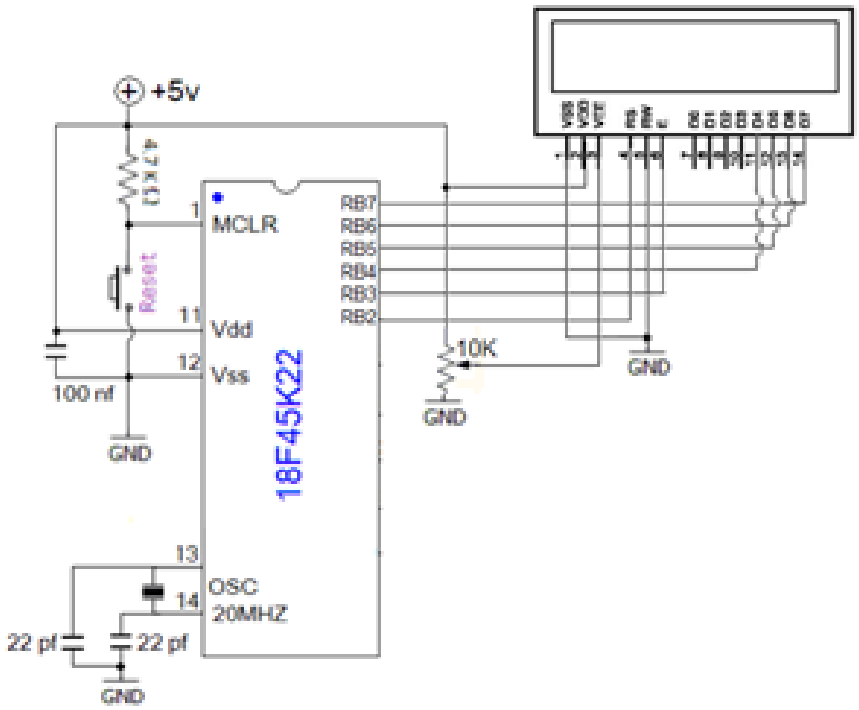
BLA: Bu arka ışıklandırmanın +5 volt için anot ucudur.

BLK: Bu arka ışıklandırmanın katot ucudur. Toprağa bağlanmalıdır.

9.3. Mikrodenetleyici ve LCD Bağlantısı

Mikro C programında LCD display kullanırken, LCD display pinlerinin mikrodenetleyicinin hangi pinlerine bağlanacağı doğru bir şekilde tanımlanması oldukça önemlidir. Bu tanımlamalar yapıldıktan sonra Mikro C Kütüphanesi kullanılarak gerekli LCD komutları işleve alınmaktadır [6].

İlk olarak LCD'nin D0 ve D7 arasındaki veri uçları PIC18F45K22 mikrodenetleyicisinin D portuna bağlanır ve bit olarak tanımlanır. Bu bağlantı yapısı istenildiği gibi değiştirilmesi mümkündür. Şekil 9.3'de PIC18F45K22 mikrodenetleyici ile 2x16'lık LCD ekranının bağlantısı şeması verilmiştir.



Şekil 9.3. PIC 18F45K22 mikrodenetleyici ile LCD bağlantı şeması

Pinlerin tanımlanmasının ardından aynı işlem tris(yönlendirme) kaydedicileri için de tanımlama bit olarak tekrarlanması gerekmektedir.

```
//LCD Ekran Port Bağlantı Numaraları
```

```
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;
// LCD Ekran Pinlerinin Port Yönlendirilmesi
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;
```

9.4. Mikro C LCD Kütüphanesi

Micro C programı genel kullanımlı LCD paneller (HD44780 uyumlu denetleyicilerle) ile haberleşmek için (4-bit ara birimli) bir kütüphane oluşturmuştur [7]. Bu kütüphane kullanımı esnasında dikkat edilmesi gereken iki nokta vardır; ilk olarak kütüphane işlevlerinden biri kullanılmadan önce LCD'nin bağlandığı portun çıkış olarak ayarlanması gerekmektedir. Bir diğeri ise LCD portu başlangıç konumuna getirilmelidir [8].

9.4.1. Kütüphane İşlevleri

i. Lcd_Init

```
void Lcd_Init (char *port);
```

LCD displayi kullanıma hazır hale getirir ve PIC mikrodenetleyicisinin tanımlı pinlerine bağlı LCD modülünü başlatmak için kullanılır.

Kullanımı şekli ise `Lcd_Init();` şeklindedir.

ii. Lcd_Out

void Lcd_Out (char row, char col, char *text);

LCD'nin belirtilen satır ve sütununa girilen metni yazdırmak için kullanılan komuttur. Hem karakter dizileri hem de karakter dizi değişmezleri metin olarak aktarabilmektedir.

Kullanımı şekli: Lcd_Out (1, 3, "merhaba"); "merhaba " yazısını LCD ekranın 1. satır ve 3. karakterden itibaren yazdırmamızı sağlar.

iii. Lcd_Out_Cp

```
void Lcd_Out_Cp ( char * text);
```

Geçerli imleç konumunda LCD'ye metin yazdırır. Hem string değişkenleri hem de değişmezler metin olarak aktarabilmektedir.

Kullanımı şekli: `Lcd_Out_Cp("buradayım!");` şu anki imleç pozisyonuna "buradayım" yazdırmamızı sağlar.

iv. Lcd_Chr

```
void Lcd_Chr ( char row, char col, char character);
```

Belirtilen konumda LCD'ye karakter yazdırır. Hem değişkenler hem de değişmezler bir karakter olarak aktarabilmektedir.

Kullanımı şekli: `Lcd_Chr (2, 3, 'i');` "i" karakterini 2. satır, 3. karakterden başlayarak yazdırmamızı sağlar.

v. Lcd_Char_Cp

```
void Lcd_Char_Cp (char character);
```

Karakteri LCD'ye imleç pozisyonundan başlayarak yazar. Hem değişkenler hem de değişmezler character olarak aktarabilmektedir.

Kullanımı şekli: `Lcd_Char_Cp('e');` "e" karakterini imlecin bulunduğu yere yazdırmamızı sağlar.

vi. Lcd_Cmp

```
void Lcd_Cmp(char command);
```

Komutu LCD'ye gönderir. Önceden tanımlanmış sabitler işleve aktarılabilmemizi sağlamaktadır.

Kullanımı şekli: `Lcd_Cmp(LCD_CLEAR);` LCD'ye ekranını temizle komutu göndermek için kullanılmaktadır.

9.4.2. LCD Sabit Fonksiyonları

LCD sabit fonksiyonları ve bu fonksiyonların görevleri şu şekildedir.

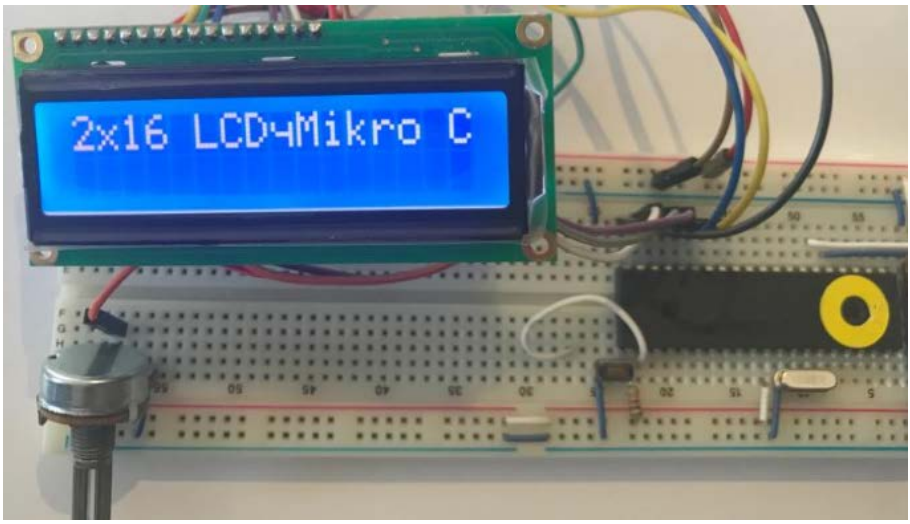
```
_LCD_FIRST_ROW  //İmleci 1. satıra taşımaktadır.
_LCD_SECOND_ROW //İmleci 2. satıra taşımaktadır.
_LCD_THIRD_ROW  //İmleci 3. satıra taşımaktadır.
_LCD_FOURTH_ROW //İmleci 4. satıra taşımaktadır.
_LCD_CLEAR    // Ekranı temizle komutu verir.
_LCD_RETURN_HOME // İmleci başlangıç konumuna getirir.
_LCD_CURSOR_OFF   //İmleci kapatmaktadır.
_LCD_UNDERLINE_ON //İmlecin altını çizkomutu vermektedir.
_LCD_BLINK_CURSOR_ON //İmlecin yanıp-sönme işlemi
_LCD_MOVE_CURSOR_LEFT //Display data RAM değişmeden imleç sola taşınır.
_LCD_MOVE_CURSOR_RIGHT //Display data RAM //değişmeden imleç sağa taşınır.
_LCD_TURN_ON //LCD ekranı aç komutudur.
_LCD_TURN_OFF //LCD ekranı kapat komutudur. _LCD_SHIFT_LEFT //Display data RAM değişmeden ekran sola kayar.
_LCD_SHIFT_RIGHT//Display data RAM değişmeden ekran sağa kayar.
```

Aşağıdaki uygulamada, LCD'nin birinci satır ve birinci sütununa "2x16 LCD" ifadesi ikinci satır ve birinci sütununa ise "Mikro C" ifadesini yazdırmak için örnek hazırlanmıştır.

```
/* LCD Uygulaması */
sbit LCD_RS at RB0_bit;
sbit LCD_EN at RB1_bit;
sbit LCD_D4 at RB2_bit;
sbit LCD_D5 at RB3_bit;
sbit LCD_D6 at RB4_bit;
sbit LCD_D7 at RB5_bit;
sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB4_bit;
sbit LCD_D7_Direction at TRISB5_bit;
void main()
{
```

```
Lcd_Init();          // LCD Display hazırlandı
Lcd_Cmd(_LCD_CURSOR_OFF); // ekrandaki imleci kaldırır.
Lcd_Cmd(_LCD_CLEAR);     //LCD ekranını temizler.
Lcd_Out(1,1,"2x16 LCD"); //1 satır ve 1.Sütundan yazdırır.
Lcd_Out(2,1,"Mikro C");  //2.satır ve 1.sütundan yazdırır.
}
```

MikroC uygulama kodu sonrasında 2x16 LCD üzerinde Şekil 9.4'de gösterilmiştir.



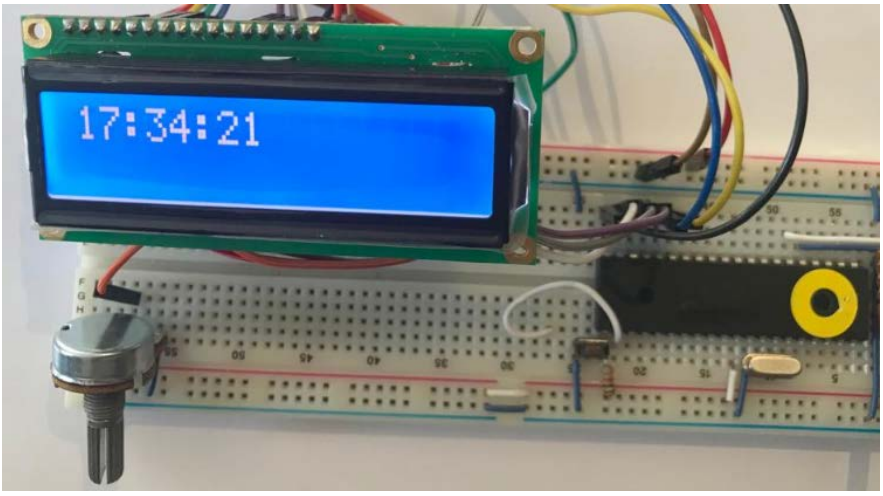
Şekil 9.4 2x16 LCD MikroC yazımın gösterimi

Aşağıdaki örnek uygulamada LCD dijital saat uygulaması MikroC kodları verilmiştir.

```
/* LCD ile dijital saat uygulaması */
char txt1[]="MERHABA DUNYA";
char txt2[]="HELLO WORD";
char txt[5];
// LCD Tanımları

Sbit LCD_RS at RD2_bit;
Sbit LCD_E at RD3_bit;
Sbit LCD_D4 at RD4_bit;
Sbit LCD_D5 at RD5_bit;
Sbit LCD_D6 at RD6_bit;
Sbit LCD_D7 at RD7_bit;


Sbit LCD_RS direction at TRISD2_bit;
```

```
Sbit LCD_E  direction at TRISD3_bit;
Sbit LCD_D4 direction at TRISD4_bit;
Sbit LCD_D5 direction at TRISD5_bit;
Sbit LCD_D6 direction at TRISD6_bit;
Sbit LCD_D7 direction at TRISD7_bit;

// SAAT DEĞİŞKENLERİ
short san=0;
short dak=0;
short sat=0;
void main() {

ANSELD=0;

LCD_Inıt();
LCD_Cmd(_LCD_CLEAR); // LCD EKRANINI SİL
LCD_CMD(_LCD_CURSOR_OFF); // KURSÖR İPTAL

LCD_Out(1,1,txt1);
LCD_Out(2,1,txt2);
Delay_ms(2000);
LCD_Cmd(_LCD_CLEAR);

while(1)
{
ByteToStr(san,txt); // Saniye değiş. karaktere dönüştürür.
LCD_Chr(1,7,txt[1]); // 1. satır 7. sütuna yazar.
LCD_Chr(1,8,txt[2]); // 1. satır 8. Sütuna yazar.
if(san>59){dak=dak+1; san=0;}//san>59 büyükse saniye=0

LCD_Chr(1,6,';');
ByteToStr(dak,txt);
LCD_Chr(1,4,txt[1]);
LCD_Chr(1,5,txt[2]);
if(dak>59){sat=sat+1;
dak=0; if(sat>23){sat=0; dak=0; san=0;}}
LCD_Chr(1,3,':');
ByteToStr(sat,txt);
LCD_Char(1,1,txt[1]);
LCD_Char(1,2,txt[2]);
}
delay_ms(1000); san=san+1;
```

MikroC uygulama kodu sonrasında 2x16 LCD üzerinde Şekil 9.5'de gösterilmiştir.



Şekil 9.5. LCD üzerinde dijital saat gösterimi

Kaynaklar

[1] LCD ekran çeşitleri, Erişim Tarihi: 07/10/2019 https://teknokoliker.com/2013/10/lcd-ekran-cesitleri-ve-lcd uygulamalari.html

[2] LCD ekran, Erişim Tarihi: 26/10/2019 https://diyot.net/lcd-ekran/

[3] PIC ile LCD, Erişim Tarihi: 26/10/2019 https://komhedos.com/PIC ile-lcd-display-uygulamasi/

[4] LCD PIC arayüzü, Erişim Tarihi: 26/10/2019 https://electrosome.com/lcd-PIC-interfacing/

[5] PIC ile LCD Display, Erişim Tarihi: 26/10/2019 https://komhedos.com/PIC-ile-lcd-display-uygulamasi

[6] Şahin, H., and Dedeoğlu, K. S., MikroC ve PIC18F4550. Altaş Yayıncılık, 2012.

[7] 16x2 LCD, Erişim Tarihi: 14/10/2019 https://components101.com/16x2-lcd-pinout-datasheet

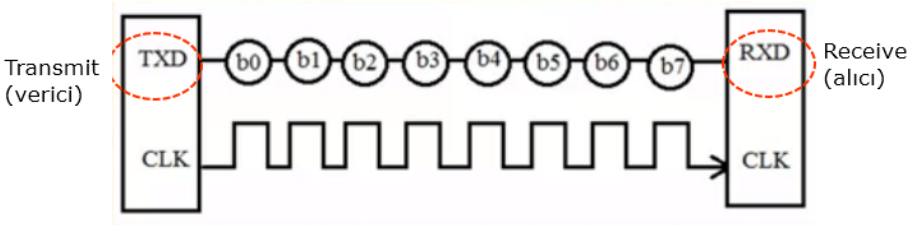
[8] LCD library, Erişim Tarihi: 26/10/2019 https://download.mikroe.com/documents/compilers/mikroc/PIC/help/lcd_library.htm#lcd_init

BÖLÜM 10

10. MİKRODENETLEYİCİ İLE SERİ PORT İLE HABERLEŞME

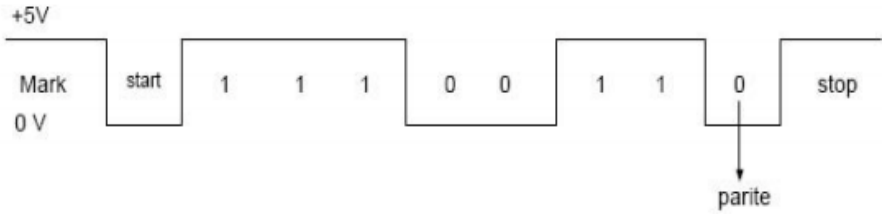
10.1. Seri Haberleşme

Haberleşme sistemlerin birbirleriyle eş zamanlı(senkron) veya eş zamansız(asenkron) olarak bilgi alışverişinde bulunmasıdır. Veri iletimi seri veya paralel olarak gerçekleşebilir. Seri iletişim paralel iletişime göre daha basit ve az donanım, veri hattı gerektirmektedir. Bu yüzden seri haberleşme daha fazla kullanılan bir iletişim yöntemidir. Analog olarak sayısal haberleşme yapmak, bilginin güvenli ve hızlı şekilde iletilmesi bazı sorunlara yol açtığı için seri veri haberleşme standartları RS-232 ve RS-485 geliştirilmiştir. Şekil 10.1.'de gösterildiği gibi seri haberleşmede bilginin iletimi tek hat üzerinden bitlerin sırasıyla bir zaman aralığında alıcı ve verici arasında gönderilmesiyle gerçekleşmektedir [1-3].



Şekil 10.1. Seri haberleşme bitlerin gösterimi

Seri haberleşmede iki yöntemle veri iletimi gerçekleşebilir. Bunlar eş zamanlı(senkron) veri iletimi ve eş zamansız(asenkron) veri iletimidir. Senkron veri iletiminde alıcı ve verici arasındaki iletişimde zamanlama ve veri kanalı olmak üzere iki kanal vardır. Gönderici alıcıya iletişimin başlaması için bir karakter gönderir. Eğer alıcı gönderilen karakteri okursa iletişim başlar ve veri gönderilir. İletişim veri gönderilene kadar devam eder. Asenkron veri iletiminde ise iletişim herhangi bir anda olabilir ayrı bir zamanlama kanalı yoktur. Veri gönderilmediği zaman hat boşta kalır ve buda senkron veri iletimine göre asenkron iletişimin daha yavaş olduğunu gösterir. Asenkron iletişimde Şekil 10.2.'de gösterildiği gibi veri iletiminin başladığını alıcıya bildiren başta başlangıç (start) biti (0 biti) ve sonda iletişimin bittiğini bildiren dur (stop) biti (1 biti) iletilen veriyle birlikte gönderilir [4-5].



Şekil 10.2. Asenkron veri iletimi

Seri haberleşmede kullanılan terimler aşağıdaki gibidir:

i. İletişim Hızı (Baud Rate): Seri iletişimde alıcı ve verici arasındaki verinin iletim hızını belirler. Saniyede gönderilen bit sayıdır. Birimi bit/saniyedir.

ii. Başlangıç Biti (Start Byte): Seri iletişimde gönderilen veriden önce gönderilen iletişimin başlangıcını belirleyen bittir.

iii. Veri Uzunluğu: Gönderilen verinin uzunluğunu belirler. Veri miktarı 5-9 bit arası olabilir fakat genellikle 8 bit uzunluğundadır.

iv. Eşlik Biti (Parity Byte): İletişim sırasında hatayı bulmak için kontrol olarak kullanılan bittir.

v. Dur Biti (Stop Byte): Veri iletiminin bittiğini alıcıya bildiren bittir [6].

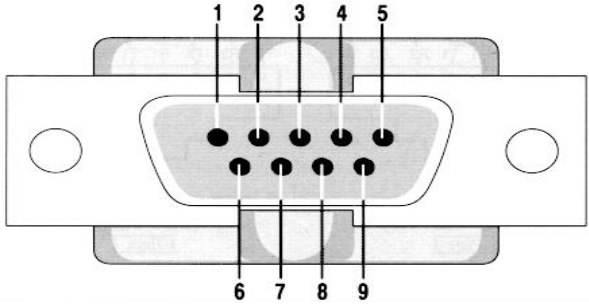
10.2. RS-232 Seri İletişim Protokolü

RS-232 cihazlarla bilgisayarın haberleşmesinde, test cihazları ve kontrol devrelerinde yaygın olarak kullanılan seri haberleşme protokolüdür. Seri haberleşme protokolü olan RS-232 ile veri bilgisi iletimi gönderilen bitlerin aynı hat ile sırasıyla gönderilmesiyle gerçekleşir. Gönderilen her bir bitin belli bir zaman aralığında gönderilmesi gerektiğinden her bir bit 1 ms sürede gönderilir buda bize her bitin voltaj aralığını göstermiş olur. RS-232 konnektöründe seri olarak gönderilen veri ASCII kodlaması olarak 8 bitlik karakterler halinde iletilir. Böylelikle 8 bitlik bir ASCII kodu 8 ms süre ile gönderimi sağlanmış olmaktadır [7-9].

İki cihaz arasında bilgi alışverişinde iletişimin sağlanması üç hat kullanılarak gerçekleşmektedir. Bunlar bilgiyi göndermek, bilgiyi almak ve toprak hattı olarak adlandırılmaktadır. RS-232 ile bilgisayar haberleşmesinde TxD (Transmit Data), RxD (Receive Data) pinleri kullanılarak veri gönderme ve alma işlemi gerçekleştirilir. Mikrodenetleyicilerde hat sinyalleri 0V ve 5V olmaktadır. Fakat RS-232 konnektörü haberleşmesi için -12 ve +12 V değerlerinde gerilim sağlanması gerekmektedir. Bundan dolayı

mikrodenetleyici ile bilgisayar arasında iletişim sağlanabilmesi için gerilim dönüştürücüler kullanılır. Bunlardan en yaygın olarak kullanılan MAX232 entegresidir [2,10].

RS-232 konnektörü bilgisayar ile haberleşmesi esnasında maksimum 20 kbps veri iletim hızında, 15 metre mesafeye kadar haberleşme imkânı sunmaktadır. İletişim alıcı ve verici arasında iki hat (RX/TX) üzerinden gerçekleşmektedir. Genel olarak veri iletimi için D tipi konnektörler kullanılmaktadır. Bunlar 9 ve 25 pin bağlantılı konnektörlerdir. D tipi 9 pinli dişi konnektör (DB-9) yaygın olarak kullanılmaktadır. Birbirleri ile haberleşme sağlayan cihazların RS-232 konnektörlerde bulunan 2 nolu (RxD) ve 3 nolu (TxD) pinleri çapraz olarak bağlantıları yapılmaktadır. 5 nolu (GND) pinleri ise düz olarak bağlanmaktadır. Şekil 10.2.'de konnektör uçları gösterilmektedir [2,11-14].



PİN	YÖN	SİNYAL
1	Giriş (IN)	Veri Taşıyıcı (DCD)
2	Giriş (IN)	Veri Alma (RxD)
3	Çıkış (OUT)	Veri Gönderme (TxD)
4	Çıkış (OUT)	Veri Terminali Hazır (DTR)
5	Toprak (GND)	Toprak Hattı (GND)
6	Giriş (IN)	Veri Hazır (DSR)
7	Çıkış (OUT)	Veri Alımına Hazır (RTS)
8	Giriş (IN)	Veri Alımı Kabul (CTS)
9	Giriş (IN)	Sinyal Algılanması (RI)

Şekil 10.3. Seri port konnektör pinleri [15]

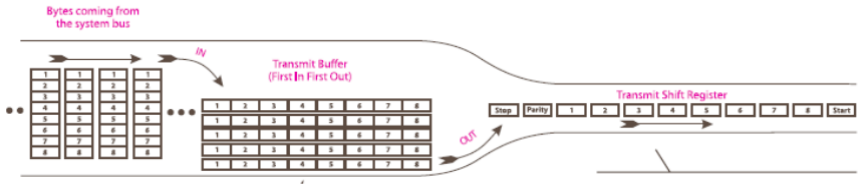
10.3. Seri Haberleşmede UART Kullanımı

UART (Universal Asynchronous Receiver/ Transmitter) seri haberleşme protokolü asenkron paralel ve seri veri dönüşümlerinde kullanılmaktadır. UART RS-232, RS-422 ve RS-485 haberleşme protokollerini desteklemekte ve MCU'dan PC'ye veya PC'den PC'ye veri haberleşmesi sağlamaktadır. UART haberleşmesi TX, RX, VCC ve GND sinyalleriyle kontrol edilmektedir [16,17]. UART haberleşmesi sırasında iletim hızı gönderilen verinin alıcı tarafından doğru alınabilmesi için belli olması gerekmektedir. Yaygın iletim hızları 4800, 9600, 19200, 57600, 115200 bit/saniyedir. Örneğin 57600 bps, saniyede 57600 bit veri gönderildiğini göstermektedir [18].

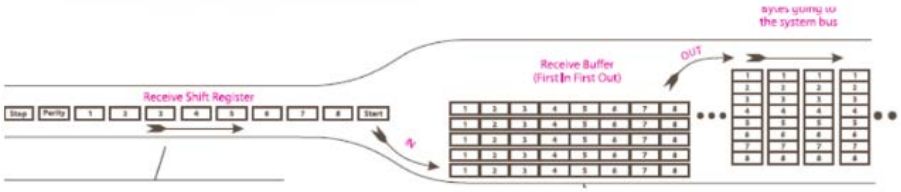
UART ile asenkron seri haberleşme 3 farklı iletişim yöntemi ile gerçekleşmektedir. Bunlar tam-çift yönlü, yarı-çift yönlü ve tek yönlü iletişimlerdir. İletişimde aynı anda iki cihaz arasında bilginin alınıp ve gönderilmesiyle tam-çift yönlü, cihazlardan sadece birinin veri alması veya gönderilmesiyle yarı-çift yönlü ve tek taraflı olarak verinin iletimi ile tek yönlü iletişim gerçekleşmektedir [19]. Alıcı ve gönderici arasında veri iletimi başlangıç biti ile başlamakta, gönderilecek olan 5-9 bit ve iletişim sırasındaki hata kontrolü için eşlik bitiyle devam etmektedir. Eşlik biti her zaman kullanılmayabilmektedir. Veri paketi başlangıcını ve sonunu işaretlemek için ise bir veya iki durdurma biti ile gönderilmektedir [20]. Başlangıç biti 0(low), durdurma biti 1(high) dir. Verinin gönderilmediği yani TxD boş durumda 1 olduğundan verinin geldiği 0 biti ile anlaşılmaktadır. Başlangıç bitinden sonra gönderilmek istenen veriler iletişim hızı bit/saniye olarak gönderilmektedir. Alıcı, durdurma biti yani 1 aldığında ise gönderilen çerçevenin anlamlı olduğunu anlayıp veriyi işlemektedir [21].

UART veri gönderme ve alma işlemlerinde mikroişlemci ile haberleşme sırasında gönderici paralel verileri alıcıya gönderme yaparken seri olarak dönüşüm gerçekleştirmekte ve bu veriyi art arda göndermektedir. Alıcı veriyi RS-232 kanalından alırken ise paralel veriye dönüştürür ve mikroişlemciye ileterek iletim sağlanmaktadır [22].

UART modülü kullanılarak yapılan veri gönderme işleminde, gönderici tarafından alıcıya paralel olarak iletilecek veriler Şekil 10.4'de gösterildiği gibi seri veriye dönüştürülüp iletimi art arda gerçekleştirilmektedir. Alıcı, veriyi RS-232 kanalından alırken ise Şekil 10.5'de gösterildiği gibi veriyi paralel veriye dönüştürerek mikroişlemciye iletmektedir [19-24].

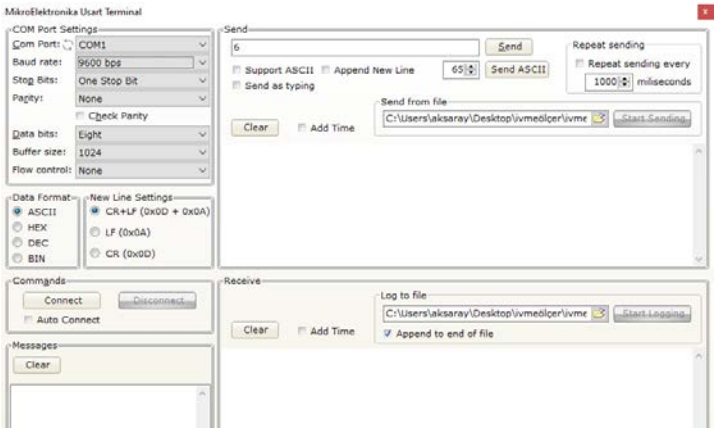


Şekil 10.4 UART Veri Gönderme [24]



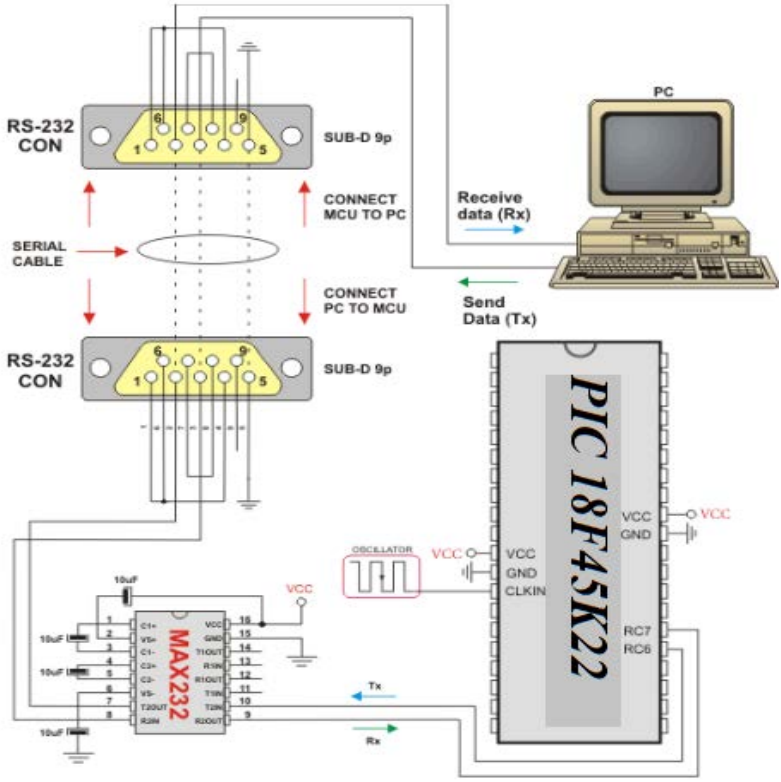
Şekil 10.5 UART Veri Alma [24]

Terminal programından ilk olarak port seçimi yapılır. Port seçimi yapıldıktan sonra Baud Rate (Saniyede Veri Gönderim Oranı) değeri ayarlanmalıdır. Baud Rate oranı MikroC derleyicisinde kullanılan değerle aynı seçim yapılarak verilerin aktarması doğru orantılı olarak gerçekleşmesi sağlanmalıdır. Şekil 10.6'da görüleceği üzere ASCII kodlar tanımlanmış ve MikroC kodu yüklenmiştir. Daha sonra yüklenen kodlar üzerinden Send (Gönder) kısmında kutucuğa istediğimiz harfin ASCII kodunu girerek karşı tarafa(alıcı) gönderme yaparak karşılık gelen harf gösterilmektedir.



Şekil 10.6 MikroC Usart Terminal

Şekil 10.7'de PIC18F45K22 mikrodenetleyici ile RS-232 devre şeması verilmiştir. Devrede PIC18F45K22 mikrodenetleyicinin, Max232, RS-232 ve PC ile bağlantısı verilmiş olup bağlantı pinleri gösterilmektedir. PC ile PIC RS-232 konnektörü ile bağlı olduğunda UART üzerinden veri değişimi mikroC derleyicisi ile yapılabilmektedir [23].



Şekil 10.7. PIC18F45K22 ile RS-232 Bağlantı Devresi [23]

Uygulamada RS-232 haberleşme protokolü ile PC ve mikrodenetleyici arasındaki veri alışverişi seri terminal (UART) aracılığı ile gözlemlenmesi gösterilmiştir [24].

```
char drx;  //RS-232 üzerinden okunan datayı saklar
void main() {
ANSELB = 0;      //Port B dijital olarak ayarlandı
ANSELC = 0;      //Port C dijital olarak ayarlandı
UART1_Init(9600);  //İletim hızı(baud rate) 9600 ayarla
Delay_ms(100);
UART1_Write(10);  //Alt satıra geçer
UART1_Write(13);  //(line feed)
UART1_Write_Text("*****Start*****");
UART1_Write(10);  //Alt satıra geçer
UART1_Write(13);  //(line feed)
while (1) {
if (UART1_Data_Ready()) {   //Eğer veriler alınırsa
drx=UART1_Read();           //Alınan verileri oku
UART1_Write(drx); //UART ile drx veri gönder
}
}
```

Kaynaklar

[1] Çölkesen, R., & Örencik, B. Bilgisayar haberleşmesi ve ağ teknolojileri. Papatya Yayıncılık, 2003.

[2] Erdem, O. A., & Orman, A. Mikrodenetleyicili Asansör Denetiminde Seri Haberleşme Kullanan Bir Model in Gerçekleştirilmesi. International Journal of InformaticsTechnologies, 1(1), 2008.

[3] Güven, B., & Uzun, T. Kontrol sistemlerinde kullanılan veri haberleşmesi teknolojileri. 1. Haberleşme Teknolojileri Ve Uygulamaları Sempozyumu, 2007.

[4] Curry, E. Message-oriented middleware. Middleware for communica tions, 1-28, 2004.

[5]MEGEP, Erişim Tarihi:05/10/2019 http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Analog%20Ve%20Say%C4%B1sal%20Haberle%C5%9Fme.pdf

[6] Durmuş, F. Radyoterapi dozimetrisinde anlık verilerin sayısal ve kablosuz yolla taşınması (Doctoral dissertation, DEÜ Sağlık Bilimleri Enstitüsü), 2009.

[7] Gülercan, M., "Asansör Panel Verilerinin Radyo Dalgalarıyla iletimi" Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü,2002.

[8] Acar, O., "Akıllı Bina Otomasyon Sistemlerinde Seri _letisim– Ethernet Entegrasyonu", Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, 2004.

[9] Aybey, Z., "Asansörde Seri Haberlesme (2), Asansör Dünyası Dergisi, Sayı 60, 2002.

[10] Öner, D., _letisim Kuramı Ders Notları, _stanbul Üniversitesi, 2001.

[11] Atçı, S., Alkan, A., Eskikurt, H. İ., & Kolip, A. Performans veri analizlerine bağlı olarak deneysel bir otomobil klima sisteminin kontrolü. Sakarya University Journal of Science, 19(2), 135-140, 2015.

[12] Taşkaya, H. O., Özgür, D., & Özyılmaz, L. Bilgisayar Kontrollü Kameralı Robot Kolu Tasarımı,

[13] Seyfettin, V. A. D. I., Güler, N., & Bayındır, R. Endüstriyel Alanlarda Kullanılan Veri İletim Tekniklerinin Karşılaştırılması. Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, 2(1), 181-188, 2014.

[14] Mohamed, I. Real Time Informatıon Updatıng System At Hospıtal Patıents Waiting Lounge (Doctoral dissertation), 2016.

[15]Aggsoftware, Erişim Tarihi:05/10/2019 https://www.aggsoft.com/rs232-pinout-cable/serial-cable-connections.htm

[16] Koçlar E. Erişim Tarihi:05/10/2019 https://www.ercankoclar.com /2018/04/uart-iletisim-protokolu-ve-mikroc-kutuphanesi/

[17]İbrahimcayiroglu, Erişim Tarihi:05/10/2019 http://www.ibrahim cayiroglu.com/Dokumanlar/MekatronikProjeUygulamasi/10-Gomulu_Sis temlerin_Seri_Haberlesmesi-Kaan_KARAPINAR-Eyup_Furkan_KAYA.pdf

[18]Herenkeskin, Erişim Tarihi:05/10/2019 http://herenkeskin.com/uart-nedir-ve-nasil-calisir/

[19] Devrim Çamoğlu, Mikrodenetleyiciler ve Elektronik

[20] MIKROE, Erişim Tarihi: 09/10/2019 https://www.mikroe.com/blog/uart-serial-communication

[21] RS232, Erişim Tarihi: 09/10/2019 http://www.fpganedir. com/ornek/rs232/index.php

[22] Aydemir, E. Lattepanda ile Arduino ve PC Kodlama. Eğitim Yayınevi, 2019.

[23] Uart Library, Erişim Tarihi: 09/10/2019 https://download.mikroe.com/ documents/compilers/mikroc/PIC/help/uart_library.htm

[24] BETİ Mikrodenetleyici Eğitim Seti Kullanım ve Deney Kitabı

BÖLÜM 11

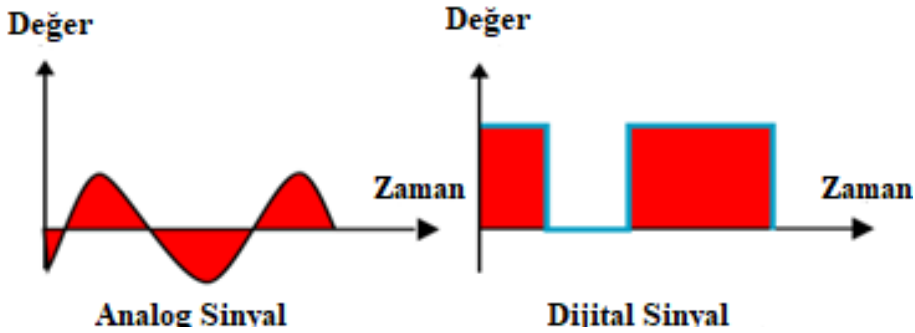
11. MİKRODENETLEYİCİ İLE PWM KONTROLÜ

11.1. GİRİŞ

Mikrodenetleyici, giriş ve çıkış birimlerini içerisinde bulunduran ve tek bir çipte barındıran bütünleşik elektronik elemandır [1]. Mikrodenetleyiciler mevcut girişleri ile dışarıdan gelen bilgiyi analog veya sayısal olarak hafızasına alan, bu bilgiyi gömülü yazılım ile derleyen ve sonucunda anlamlı çıkış elde eden bilgisayarlardır. Sinüs dalgası olarak iletilecek olan analog sinyalin sayısal sinyale dönüştürülmesi için farklı modülasyon yöntemleri vardır. Mikrodenetleyici ile darbe genlik modülasyonu (Pulse-Width-Modulation/PWM) yöntemi, analog devrelerin dijital çıkışlarla kontrolünde kullanılan önemli bir tekniktir [2].

11.2. Analog ve Sayısal Sinyaller

Analog (örneksel) sinyaller; sinüs eğrisi gibi dalgalı ve sürekli olarak değişen sinyallerdir. Analog sistemlerde sinyaller belirli değerler arasında genlik, frekans ve faz parametreleri değiştirilerek sonsuz değer alabilirler. Analog sinyaller bu özelliği ile sayısal sinyallerden ayrılır. Sayısal(dijital) sinyaller açık ve kapalı değerler arasında belirlenmiş sonlu değerleri alabilirler. Şekil 11.1'de analog ve sayısal sinyaller gösterilmiştir [3-4].

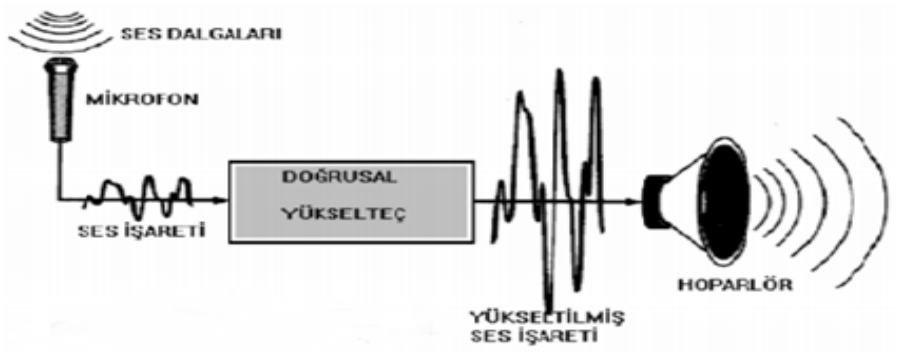


a) b)

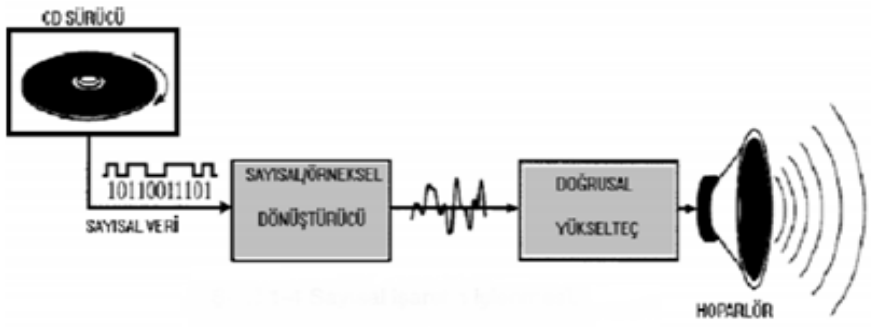
Şekil 11.1 Sinyal Çeşitleri a) analog sinyal b) dijital sinyal

Şekil 11.2'de gösterildiği gibi mikrofon sesinin analog sinyale dönüştürürken sinyal zayıf olacağından yükselteç kullanılarak yükseltilmiş ses işaretine dönüştürülmesi bu şekilde kuvvetlendirilmesi gerekmektedir. Ayrıca şekil

11.3'de ise sayısal bir verinin işlenmesi gösterilmiştir. Sayısal veri ilk önce dijital-analog dönüştürücü ile analog sinyale çevrilerek yükselteçle sinyal güçlendirilmektedir [5-6].



Şekil 11.2 Analog İşaretin İşlenmesi

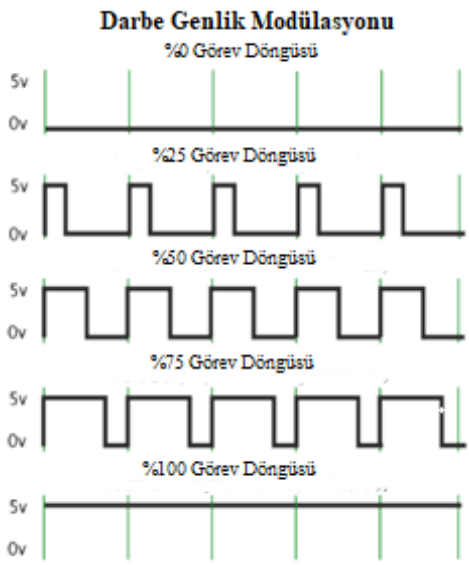


Şekil 11.3 Sayısal İşaretin İşlenmesi

Mikrodenetleyiciler kendi bünyesinde analog bir değerin işlenebilmesi için Analog Dijital Dönüştücü (ADC) ve analog sinyale dönüşütürlmesi için Dijital Analog Dönüştürücüye (DAC) ihtiyacı vardır. Sinüs dalgası olarak iletilecek olan bilginin sayısal iletim ortamına adapte edebilmek için farklı modülasyon yöntemleri vardır. Bu yöntemlerden biriside PWM'dir [7].

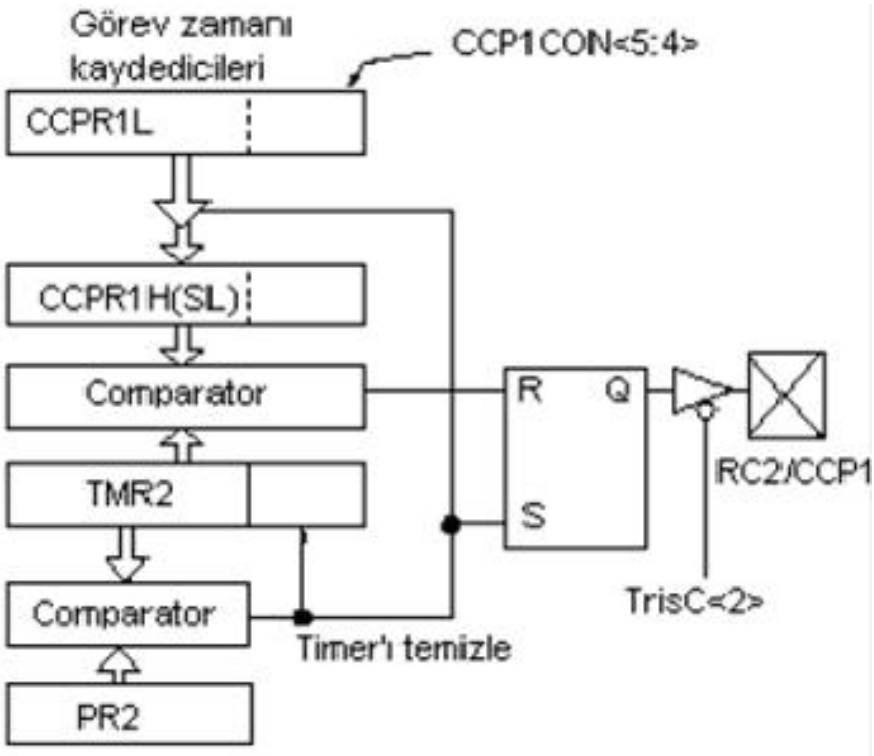
11.3. Darbe Genişliği Modülasyonu (PWM)

PWM açılım olarak Darbe Genişliği Modülasyonu anlamına gelmektedir. PWM analog devrelerin dijital çıkışlarla kontrolünde kullanılan önemli bir tekniktir. PWM yöntemi, güç kontrolü ve dönüşümü olmak üzere birçok farklı uygulamada kullanılmaktadır [8]. PWM sinyal işleme ve iletimi gibi daha fazla elektronik alanlarda uygulanmaktadır [9]. PWM tekniğinde çıkış akımına dayanarak güç kontrolü anahtarlama ile sağlanmaktadır [10]. PWM tekniğinin asıl gayesi kare dalga üretmek ve çıkışta istenilen dalganın yapısını kanal üzerinde çeşitli ayarlamalarla belirlemektir [11]. PWM tekniğinde karşılaşılan görev döngüsü "Duty Cycle" kavramı yapılan işlemin periyodunu gösterir [12]. Devredeki dalganın genliği görev döngüsü boyutuyla orantılıdır [9]. PWM sinyali ile belirli aralıklarda sinyalin açık durumunda +5V ve kapalı durumda ise 0V gönderilerek ortalama gerilim değiştirilir [12]. Şekil 11.4' de gösterilen Darbe Genişlik Modülasyonunda 0-5V değerinde gerilime sahip sinyalin ortalama gerilimini değiştirerek çıkışta bu aralıktaki gerilimler elde edilmektedir. PWM darbe genliklerini değiştirerek çıkışta istenilen sinyali kontrol eder [13]. Dalganın genliği görev döngüsü DC-AC dönüştürücüler PWM metodu ile kontrol edilebilir [14].



Şekil 11.4 PWM (Darbe Genlik Modülasyonu) dalga şekilleri

PIC içerisinde olan CCP modülü compare (karşılaştırma), capture (yakalama) ve PWM bölümlerinden oluşan bir modüldür. CCPx pininden 10 bit çözünürlükte PWM çıkış sağlanır. CCPx biti 1 olduğunda PWM modu etkin olurken bit 0 olduğunda CCPx kullanım dışı olmaktadır. PWM modundayken işlem yapılabilmesi için TRISC<2> biti çıkış olarak konumlandırılmalıdır. Şekil 11.5'de PWM modunun blok diyagramı verilmiştir. PIC ile PWM sinyali üretilirken PWM periyodu PR2 değerine yazılır. PWM periyodu TMR2 ve PR2 değerlerinin eşit oluncaya kadar geçen süredir. Periyot PR2 kaydedicisine yazıldıktan sonra TMR2 temizlenir ve CCPx pini set edilir. Akabinde PWM sinyalinin görev döngüsü değeri CCPRxL'den CCPRxH'a yüklenir [5,28].



Şekil 11.5 PWM modu blok diyagramı

11.4. MikroC Programlama PWM Kütüphanesi

MikroC derleyicisinin PWM kütüphanesi PWM metodunun uygulamasını kolaylaştırmaktadır. PWM kütühanesi bize 4 adet fonksiyon sunar. Bunlar PWM1_Init(), PWM1_Start(), PWM1_Set_Duty() ve PWM1_Stop() dur.

i. PWM1_Init()
PWM1_Init(frekans) şeklinde girilen frekans değeri PWM dalgasının çalışma frekansını belirler. Frekans parametresi Hz cinsinden istenen frekanstır. Sayısal bir sabit olmalı, değişken olmamalıdır. Örneğin;
PWM1_Init(5000); 5 kHz,
PWM1_Init(10000); 10 kHz PWM oluşturur.

ii.PWM1_Start()
Bu fonksiyon PWM işlemlerinin başlaması için yazılır. Bu komut çağrılmadan önce PWM1_Init() çağrılmalıdır.

iii.PWM1_Set_Duty()
Bu işlev PWM'nin görev döngüsünü ayarlamak için kullanılır. Parametre 0 ila 255 arasında değerler alır. Yani 0, % 0, 127, % 50 ve 255, % 100 görev döngüsü anlamına gelir. Bu değerler (Yüzde*255) /100 formulü ile hesaplanır. Bunu kullanmadan önce PWM1_Init() komutu çağırılmalıdır. PWM geçerli değerinin yeni değerlerle değiştirilmesi için kullanılır.

iv.PWM1_Stop()
Bu fonksiyon PWM dalgasını durdurur. Bu komut çağrılmadan önce PWM1_Init() çağrılmalıdır [15].

11.5. PWM İLE LED KONTROLÜ UYGULAMASI

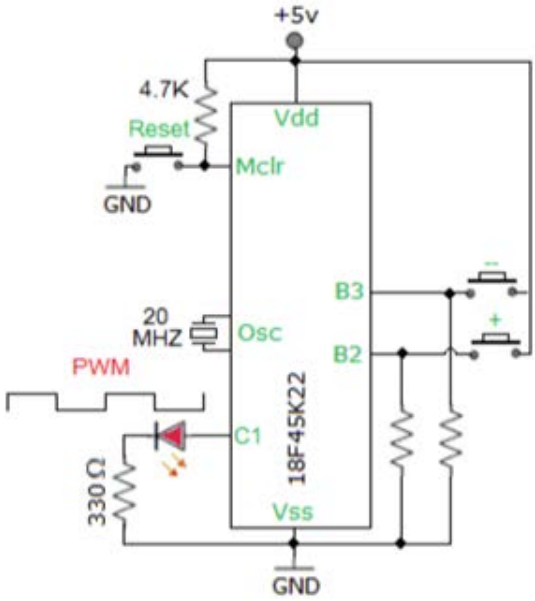
Dijital çıkış kullanarak ledin yanıp sönmesini sağlayabiliriz. Dijital çıkış aradaki değerler yok sayılarak 0 ve 1 lerden oluşur. Bu da ledin ya açık ya da kapalı olması durumudur. Ledin yavaş yavaş sönmesi veya yavaş yavaş kapanması için analog çıkış üretilmesi gereklidir. Bunun için ise PWM tekniği kullanılarak ışığın şiddetini ayarlayabiliriz [14].

PIC18F45K22 mikrodenetleyicisi flash program belleği 32 Kbytes, RAM veri belleği 1536 bytes, 256 bytes EEPROM veri belleğine sahiptir. PIC18F45K22 entegresi düşük güçlü, yüksek performanslı ve hızlıdır. Ayrıca PIC18F45K22 entegresi 2 PWM, 3 adet de güçlendirilmiş PWM çıkış sağlamaktadır. Her

çıkışın periyodu programlanabilmektedir [17-18]. PIC 18F45K22 mikrodenetleyici ile PWM kontrol MikroC uygulama kodları aşağıda verilmiştir.

```
/*PWM Kontrol */
unsigned short wis; // % OLARAK İŞ ZAMANI TANIMLAMA
void main() {
ANSELB =0;           // PORT B DİJİTAL GİRİŞ
ANSELC =0;           // PORT C DİJİTAL GİRİŞ
C1ON_bit = 0;       // COMPARATOR  KAPAT
C2ON_bit = 0;
TRISC = 0;           // PORTC PWM ÇIKIŞI
TRISB=12;            // b2 b3 GİRİŞ OLARAK AYARLANDI
PORTC = 0;          // PORTLARI SIFIRLA
PWM1_Init(1500);     // PWM ÇALIŞMA FREKANSI 1.5 KHZ
wis  = 10;           // BAŞLANGIÇ DUTY AYARI %10
wis= (wis*255)/100;  //YÜZDE HESAPLAMA
PWM1_Start();        // PWM1 BAŞLAT
PWM1_Set_Duty((wis*255)/100);  //PWM AYARLA ÇIKIŞ GÖNDER
                     //(%10 DEĞERİ 25'E KARŞILIK GELİR)
while (1){
  if (PORTB.B2==1)
    {
      Delay_ms(40);
     wis=wis+1;         //DUTY ARTTIR
     if(wis>100) wis=100;   // %100 DEN BÜYÜKSE 100 YAP
     PWM1_Set_Duty((wis*255)/100);//PWM ÇIKIŞA GÖNDER
     }
  if (PORTB.B3==1)
{                     // B3 BUTONUNA BASILIRSA
      Delay_ms(40);
     wis=wis-1;       //DUTY EKSİLT
     if(wis<1) wis=1;
     PWM1_Set_Duty((wis*255)/100);   // PWM  ÇIKIŞA GÖNDER   }
 Delay_ms(5);
  }
}
```

18F45K22 Mikrodenetleyici kullanılarak PWM kontrolüne ait devre şeması Şekil 11.6'de verilmiştir.



Şekil 11.6 PWM Devresi

11.6. PWM DC Motor (Fan) Kontrolü Uygulaması

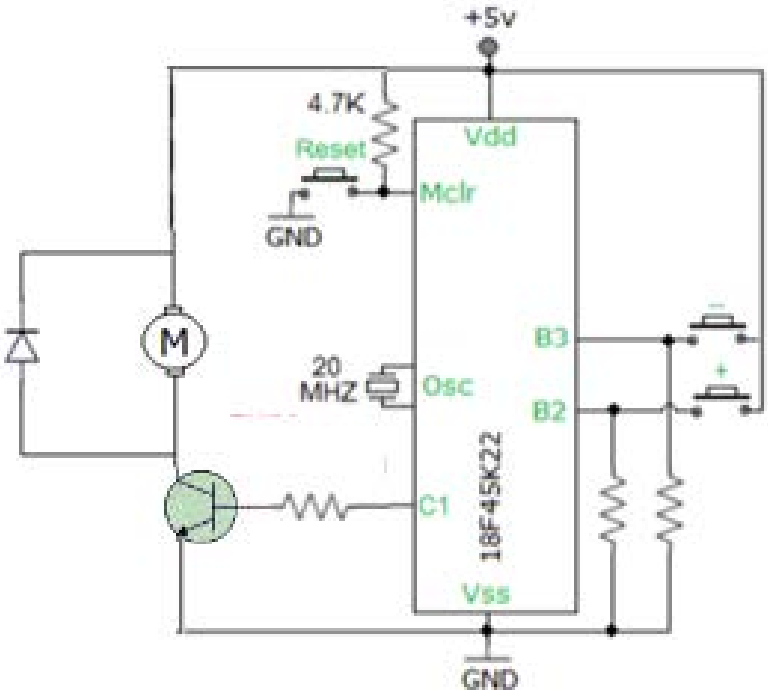
Analog olarak yapılan DC motor ve fan kontrolünde ise motoru yavaşlatmak için akımı veya voltajı azaltan reosta ve potansiyometre kullanılmaktadır. Motor veya fanın güçlenmesiyle kontrol edebilmesi için aynı şekilde reosta veya potansiyometrenin güçlü olması gerekmektedir. Buda reostanın daha büyük ve pahalı olması demektir. PWM metodu bu yüzden analog kontrole göre daha avantajlıdır [29].

PWM kontrol devresi ile DC motorun hızı ve fanın dönme hızı kontrol edilebilir. Motorun hızını yani devir sayısını kontrol etmek için sinyalin darbe genişliğini arttırıp veya azaltırız. Yüksek çözünürlüklü sayıcılar kullanarak kare dalga sinyalin Sinyal-Boşluk Oranı (SBO) (Duty Cycle) bir analog sinyalin spesifik değeri için modüle edilir. Duty Cycle arttığında yani daha uzun sinyal verildiğinde motorun hızını arttırmış oluruz, azaltırsak eğer gerilim azalır ve hız doğru orantılı olarak düşer. Görev döngüsünü ayarlayarak bu şekilde motoru kontrol etmiş oluruz [17].

Genellikle kullanılan 3 farklı elektronik değişken hız sürücüleri vardır. Bunlar voltaj dönüştürücü (VSI), akım dönüştürücü (CSI) ve sinyal-genişlik modülasyon kaynaklı dönüştürücü (PWM) 'dir. Darbe genişlik modülasyonu (PWM) diğer yöntemlere nazaran en verimli değişken hız sürücü inverterleridir. Şebekeden Motora gelen gerilim ve frekans değerini istediğimiz şekilde ayarlayabilmek için Darbe Genişlik Modülasyonu (PWM) tekniği kullanılmaktadır. Motor hızının değiştirilmesi sonucunda motora bağlı olan fan hızının da kontrolü sağlanmış olacaktır. DC fırçasız bir fan için PWM kontrol devresinde fanın sinyal terminalini tespitini sağlayan gerilim sinyali alan doğrultucu bulunmaktadır. Gerilim sinyali kare dalga sinyali şeklinde olmaktadır. Doğrultulmuş gerilim sinyali ile anahtarlama yapan cihaz karşılaştırma yapmaktadır. Bu karşılaştırma sonucunda doğrultulmuş DC gerilim sinyali, referans gerilim sinyalinden düşük veya yüksek ise fan açılmış demektir. Bu nedenle fanı aralıklı olarak çalıştırma işlemi yapılmakta ve dönme hızı kontrolü sağlanmaktadır. Sonuç olarak fanın dönme hızı kontrol edilebilir veya sabit tutulabilir [19-27].

```
unsigned short sduty;    //% OLARAK İŞ ZAMANI TANIMLAMA
void InitMain() {
  ANSELA  = 0;           // PORT A DIJITAL GIRIS
  ANSELC = 0;            // PORT C DIJITAL GIRIS
  C1ON_bit = 0;          // COMPARATOR  KAPAT
  C2ON_bit = 0;
  TRISA = 255;           // PORTA BUTON GİRİŞİ
  TRISC = 0;             // PORTC PWM ÇIKIŞI
  PORTA = 0;
  PORTC = 0;            // PORTLARI SIFIRLA
  PWM1_Init(2000);      // PWM 2KHZ AYARLA
}

void main() {
  InitMain();
    sduty  = 10;           // BAŞLANGIÇ DUTY AYARI  %10 sduty=
(sduty*255)/100;  //YÜZDE OLARAK HESAPLAMA
  PWM1_Start();           // PWM1 BAŞLAT
  PWM1_Set_Duty((sduty*255)/100); // PWM ÇIKIŞA GÖNDER

   while (1) {
     if (RA0_bit) {           // RA0 BUTONUNA BASILIRSA
      Delay_ms(40);
      sduty=sduty+1;
     if(sduty>100) sduty=100;  // %100 DEN BÜYÜKSE 100 YAP
      PWM1_Set_Duty((sduty*255)/100);// PWM ÇIKIŞA GÖNDER
     }
```

```
      if (RA1_bit) {    // RA1 BUTONUNA BASILIRSA
        Delay_ms(40);
        sduty=sduty-1;
        PWM1_Set_Duty((sduty*255)/100);// PWM ÇIKIŞA GÖNDER
      }

      if (RA2_bit) {     // RA2 BUTONUNA BASILIRSA
        PWM1_Stop();      // PWM ÇIKIŞI İPTAL
      }
      Delay_ms(5);
  }
}
```



Şekil 11.78. 18F45K22 mikrodenetleyici ile PWM DC motor kontrolü

Kaynaklar

[1] Bakan, N. D., Özkan, G., Çamsari, G., Gür, A., Bayram, M., Açikmeşe, B., & Çetinkaya, E. Silicosis in denim sandblasters. Chest, 140(5), 1300-1304, 2011.

[2] Valentine, D. Using PIC processors in computer organization. Journal of Computing Sciences in Colleges, 24(1), 116-122, 2008.

[3] Biçer, Z. Sayısal modülasyonlarda dalgacık dönüşüm temelli bir akıllı sınıflandırma sistemi/An intelligent classification system based on wavelet transform for digital modulations, 2007.

[4] Öner, İ. V., Yeşilyurt, M. K., & Yılmaz, E. Ç. Wavelet Analiz Tekniği ve Uygulama Alanları. Ordu Üniversitesi Bilim ve Teknoloji Dergisi, 7(1), 42-56, 2017.

[5] Şahin, H., Dayanık, A., & Altınbaşak, C. PIC programlama teknikleri ve PIC16F877A. Altaş Yayıncılık, 2013.

[6] Zeki, A. Elektronikte Analog Dijital Yanılsamalar.

[7] Ronayne, J. Sayısal Haberleşmeye Giriş, Ceng. FIEE Communications Consultant, Millî Eğitim Basımevi, İstanbul, 1997.

[8] Barr, M. Pulse width modulation. Embedded Systems Programming, 14(10), 103-104, 2001.

[9] George, L. Generating PWM with PIC Microcontroller- MikroC Pro. Erişim Tarihi: 28/10/2019 https://electrosome.com/pwm-PIC-microcontroller/

[10] Stanley, G. R. U.S. Patent No. 6,683,494. Washington, DC: U.S. Patent and Trademark Office, 2004.

[11] Deniz, E., & Altun, H. Beş seviyeli izo. Le DC kaynaklı kaskat inverterin spwm teknigi ile kontrolü. Sakarya University Journal of Science, 11(1), 1-9, 2007.

[12] Karaca, A., & Conker, Ç. Bulanık Kontrolör Esaslı Haptik Robotik El Fuzzy Controller Based Haptic Robotic Hand.

[13] Mazumder, M. I. K., Mahbub, M. M., Rahman, M., & Amin, G. Design and analysis of DC-DC PWM converter and DC-AC converter (Doctoral dissertation, BARC University), 2017.

[14] Korcharz, D., & Ferentz, A. U.S. Patent No. 6,049,471. Washington, DC: U.S. Patent and Trademark Office, 2000.

[15] Kalpak, D. E. DC Voltage Measurement Using the PIC Microcontroller and PWM (doctoral dissertation, european unıversıty of lefke), 2015.

[16] Çil, C. Z. Işık Yayan Diyotlar (LED'LER) ve Aydınlatmada Kullanımı.

[17] AL-Farsi, H. S. H., & Malathi, B. N. Accident Notification System by using Two Modems GSM and GPS.

[18] Ibrahim, D. Advanced PIC microcontroller projects in C: from USB to RTOS with the PIC 18F Series. Newnes, 2011.

[19] Hsieh, H. M. U.S. Patent No. 5,942,866. Washington, DC: U.S. Patent and Trademark Office, 1999.

[20] Tassou, S. A., & Qureshi, T. Q. Comparative performance evaluation of positive displacement compressors in variable-speed refrigeration applications. International Journal of Refrigeration, 21(1), 29-41, 1998.

[21] Qureshi, T. Q., & Tassou, S. A. Variable-speed capacity control in refrigeration systems. Applied thermal engineering, 16(2), 103-113, 1996.

[22] Shuangquan, S., Wenxing, S., Xianting L. Ve Huajun, C., "Performance Representation of Variable-Speed Compressor for Inverter Air Conditioners Based on Experimental Data", International Journal of Refrigeration, 27,1, 805–815, 2006.

[23] Kim, M., & Kim, M. S. Performance investigation of a variable speed vapor compression system for fault detection and diagnosis. International Journal of Refrigeration, 28(4), 481-488, 2005.

[24] Aprea, C., Mastrullo, R., & Renno, C. Experimental analysis of the scroll compressor performances varying its speed. Applied thermal engineering, 26(10), 983-992, 2006.

[25] Jiangjiang W., Dawei A. ve Chengzhi, L., "Application of Fuzzy-PID Controller in Heating Ventilating and Air-Conditioning System", Mechatronics and Automation, Proceedings of the IEEE International Conference, syf. 2217 – 2222, 25-28 June 2006.

[26] Trzynadlowski, A.M. Control of Induction Motors. Acedemic Press: London, 2001.

[27] Prof. Krishna Vasudevan, Prof. G. Sridhara Rao, Prof. P. Sasidhara Rao. Electrical Machines II. National Programme on Technology Enhanced Learning: http://nptel.ac.in/courses/IIT-MADRAS/Electrical_Machines_II/pdf/1_8.pdf

[28] PICProgramlama, Erişim Tarihi:05/10/2019 http://firatdeveci.com/wp-content/uploads/Electronics/Hi-Tech%20ile%20PIC%20Programlama.pdf

[29] BETİ Mikrodenetleyici Eğitim Seti Kullanım ve Deney Kitabı,

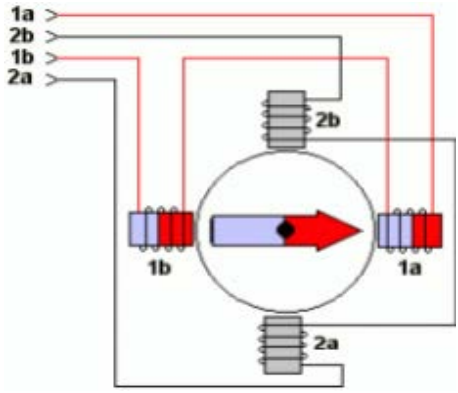
BÖLÜM 12

12. MİKRODENETLEYİCİ İLE ADIM (STEP) MOTOR KONTROLÜ

12.1. Adım (Step) Motorlar

Step motorlar dönme hareketi ile elektrik enerjisini fiziksel enerjiye çeviren elektromekanik cihazlardır [1]. Step motorlar adımlar halinde açısal konumunu değiştiren, çok hassas sinyallerle sürülebilen motorlardır [2]. Motorun yapısına göre adım açısı 180°, 90°, 45°, 18°, 7.5°, 1.8° veya daha farklı açılarda olabilir. Step motorun hızı, motorun bobinlerine uygulanan frekans değiştirilerek kontrol edilebilir [3]. Step motorun dönüş yönü stator sargılarına uygulanan sinyallerin sırasının değiştirilmesi ile sağlanır. Motorun hangi yöne gideceği ve hızı gibi değerler bir mikroişlemci ve sürücü yardımıyla kontrol edilebilir [5].

Step motorlar rotor, stator ve rulmanlardan oluşmaktadır. Rotora bağlı olan şaftın rahat hareket etmesini sağlamak için rulmanlar kullanılır. Statorun birden fazla kutbu bulunur. Kutupların polaritesi elektronik anahtar vasıtasıyla sürekli değiştirilir. Biraz daha detaylı açıklayacak olursak, çalışma prensibi şu şekildedir. Bobinlere elektronik anahtarlar ile enerji verilir ve rotor enerji olan bobinin karşısında durur (Şekil 12.1). Motor dönmek istediği sürece bobinlere sırasıyla pals sinyali verilir. Motorun dönme açısı değişkendir ve motor tercihi yapılırken dönme açısı çok kritik bir parametredir [6,7].



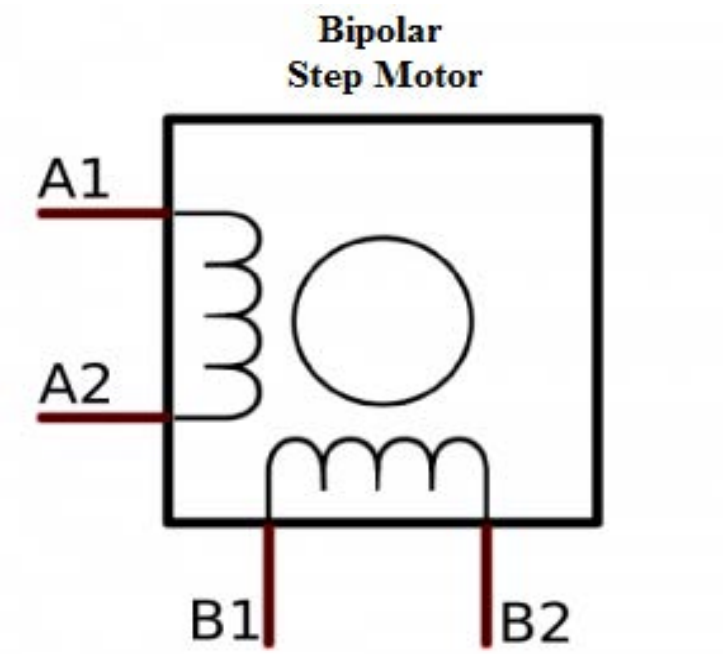
Şekil 12.1. Step Motor Adımı [8]

12.2. Adım (Step) Motor Çeşitleri

Step motorlar uygulama alanlarında genellikle bipolar (tek kutuplu) ve unipolar (çok kutuplu) olmak üzere iki türde kullanılır.

12.2.1 Bipolar Step Motorlar

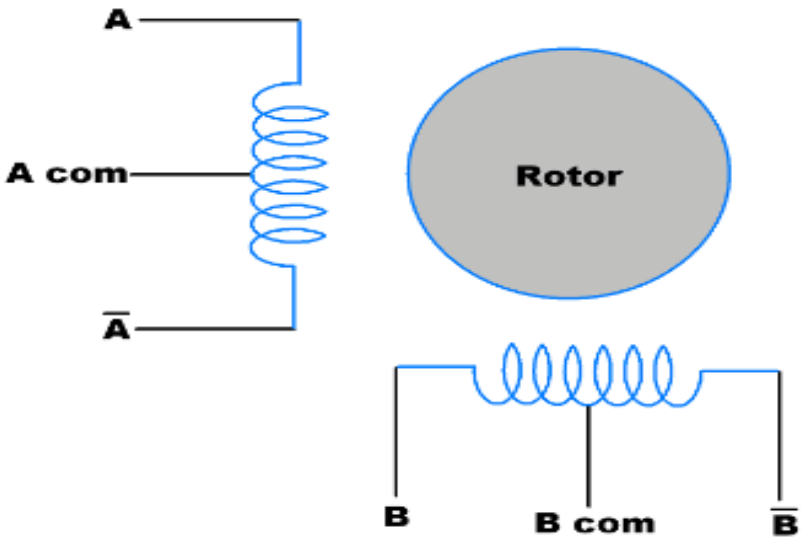
Bipolar step motorlar faz başına tek sargı içeren step motorlardır. Bobinlerine uygulanan sayısal darbelere bağlı olarak belirli bir açıda dönme hareketi yaparlar. Bipolar step motorlarda Şekil 10.2'de görüldüğü gibi 4 uç bulunmaktadır. Bobin uçlarına uygun sırada lojik "0", lojik "1" uygulanarak step motorun dönüşü sağlanır. Bobin uçlarına uygulanan sinyallerin sırası değiştirilerek dönüş yönü değiştirilmektedir [10].



Şekil 12.2. Bipolar Step Motor [9]

12.2.2 Unipolar Step Motorlar

Unipolar step motorlarda 5,6 veya daha fazla uç bulunabilir. Şekil 3' de 6 uçlu unipolar step motorun iç yapısı gösterilmektedir. Genellikle her sarım için ortak bir uç vardır. Ortak bobin ucuna göre diğer bobin uçlarına sırayla lojik sinyaller uygulanır. İki fazlık bir unipolar step motorda 6 uç bulunur. Çoğunlukla bu iki ortak faz uçları motor içinde birleştirilir ve sonuç olarak 5 kablo ucu bulunur [11].



Şekil 12.3. Altı (6) Uçlu Unipolar Step Motor [12]

12.3 Step Motorlarda Uç Tespitinin Yapılması

Step motorlarda genel olarak 4, 5 veya 6 uç bulunmaktadır. 5 uçlu step motorlarda bir orta uç bulunurken 6 uçlu motorlarda iki orta uç bulunmaktadır. Motorun ortak uçlarına besleme gerilimindeki + uç bağlanır. Step motorlarda uç tespiti avometre ohm (veya buzzer) kademesine getirilerek yapılır. Ölçülen uçlar arasındaki direnç değerlerine göre uç tespiti yapılır. Step motoru düzgün çalıştırabilmek için tespi edilen bobin uçları doğru sırada bağlanmalıdır. Kablo sıralaması, bobin uçlarına besleme geriliminden gelen enerji uygulanarak deneme yanılma yoluyla tespit

edilebilir. Bobin uçları hatalı bağlanırsa step motor düzgün çalışmaz motorda titremeler oluşabilir [11].

12.4 Step Motorun Çalışması

Step motorlar çalışırken sürücü devrelerine ihtiyaç duyarlar [4]. Sürücüler yardımıyla motorun bobinlerine giriş palsi uygulandığında motor belirli bir açıda döner ve durur. Motorun dönüş miktarı, motorun kendi yapısına göre belirli bir açı ile sınırlandırılmıştır. Step motorların adım açısı motorun yapısına bağlı olarak 90°, 45°,18°, 7.5°, 1.8° veya daha değişik açılarda olabilir. Motor sürücüsüne uygulanan pals adedi kadar rotor hareket ettirilerek motorun dönüşü sağlanır [13]. Step motorlar tek fazlı ve çift fazlı çalıştırılarak hareket ettirilebilir. Motorun tek fazlı çalıştırılmasında bobinin orta ucuna pozitif besleme, diğer bobin uçlarına sırasıyla negatif sinyal uygulanır. Step motor çalışmalarında torkun yüksek olması ve durma karakteristiklerinin iyi olması sebebiyle çift fazlı çalışma metodu daha çok tercih edilir. Çift fazlı çalışma metodunda bobinlerin orta ucuna pozitif gerilim, diğer bobin uçlarına ikili olarak negatif pals uygulanır. Tek fazlı ve çift fazlı çalışmada kullanılan lojik sinyaller Tablo 1-a ve b'de görülmektedir [11].

Tablo 12.1.Step motor çalışma modları [11]

Tek Faz Çalışma

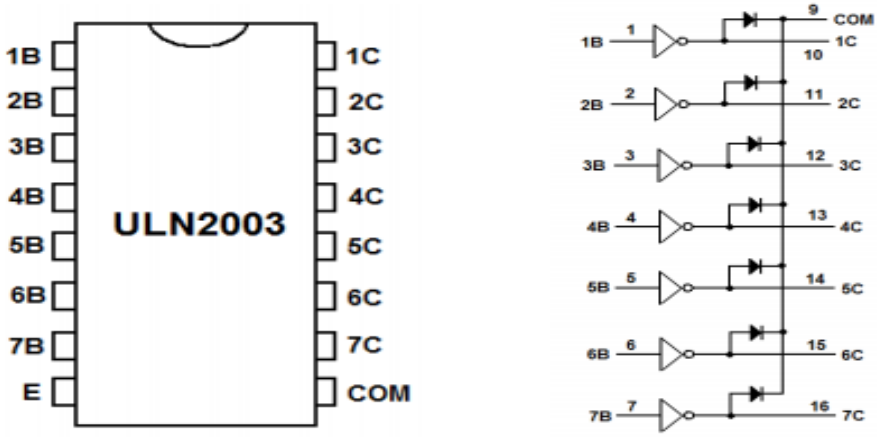
Adım	A1	B1	A2	B2
1	0			
2		0		
3			0	
4				0

Çift Faz Çalışma

Adım	A1	B1	A2	B2
1	0			0
2	0	0		
3		0	0	
4			0	0

12.5 Step Motor Sürücü (ULN2003)

ULN2003 entegresi kendi içerisinde 7 adet darlington NPN transistör ve katotları ortak bağlı koruma diyotlardan oluşmaktadır. Entegre yapısında bulunan darlington transistörler 50 V çıkış gerilimi ve 500 mA kollektör akımında çalışabilecek özelliğe sahiptirler. CMOS teknolojisi ile üretilen bu entegre 5 V ile çalışabildiğinden motor sürme devrelerinde, led ve lamba sürme devrelerinde, yüksek güçlü tampon devrelerde sıklıkla kullanılmaktadır. Şekil 12.4'te ULN2003 entegresinin iç yapısı ve bacak bağlantıları görülmektedir [14].



a) Bacak Bağlantısı b) ULN2003 İç Yapısı

Şekil 12.4. ULN2003 bacak bağlantısı ve iç yapısı [14]

12.6 Mikrodenetleyici ile Fonksiyon Oluşturma

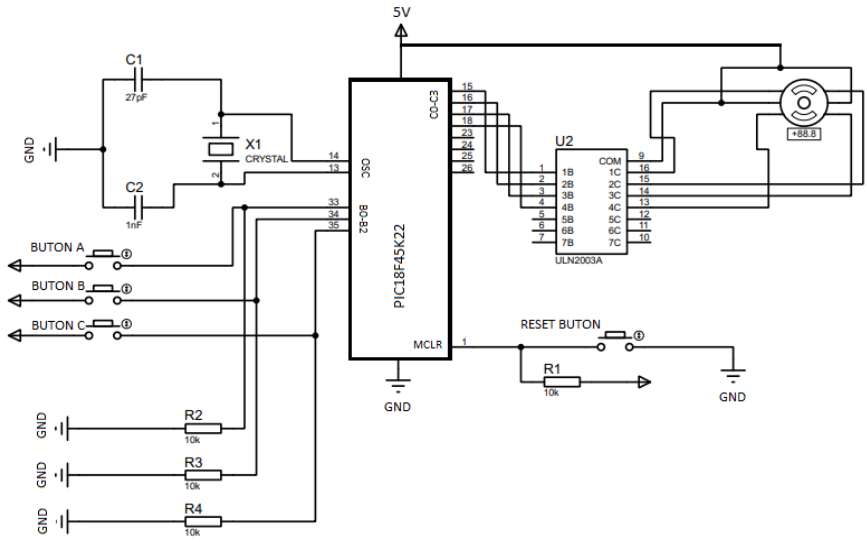
Fonksiyonlar, aldıkları parametrelere bağlı olarak bazı görevleri yerine getiren alt programlardır. Program yazarken birden fazla kez tekrarlanan işlemleri tekrar tekrar yazmak yerine fonksiyonlar kullanılarak bu işlemler kolay bir şekilde program içerisinde kullanılabilir. Fonksiyonlar giriş parametrelerindeki değerleri işleyerek geriye değer döndürebildikleri gibi fonksiyon içinde bulunan değerler ile değişiklikler yapıp geriye değer döndermeyebilir [11]. C dilinde hazır bulunan kütüphanelerdeki fonksiyonlar kullanılabilir veya kullanıcı yapacağı işe göre kendi fonksiyonlarını oluşturabilir. Oluşturulan fonksiyonlar istenildiğinde programın içerisinde kullanılabilir [15].

12.7 Mikrodenetleyici ile Step Motor Kontrol Uygulaması

Uygulamada 6 uçlu unipolar step motor ve motoru sürmek için ULN2003A sürücü kullanılmıştır (Şekil 12.5). Step motor tek fazlı çalışma prensibinde çalıştırılmaktadır. Step motorun kontrolü için C portunun RC0,RC1,RC2,RC3 pinleri kullanılmıştır. Başlangıçta adımlar arasında 100 ms'lik bir gecikme süresi konmuştur. Mikrodenetleyicinin B portuna bağlanan A, B ve C butonları ile gecikme süresi ayarlanarak motorun çalışma hızı değiştirilebilmektedir. Motorun ileri ve geri adımları oluşturulan ileri geri fonksiyonlar ile yapılmaktadır.

```
/* STEP MOTOR KONTROLÜ */
#define butonA portb.rb0
#define butonB portb.rb1
#define butonC portb.rb2
int i;
int sure;
bekle(int ssure)
{
 Vdelay_ms(ssure);  //ADIM SÜRESİ AYARLAMA
 if(butonA==1) sure=sure+10; //BUTON A'YA SÜRE ARTTIR
 if(butonB==1) sure=sure-10;//BUTON B'YE SÜRE AZALT
 if(butonC==1) sure=100; //BUTON C'YE SÜREYİ 100ms yap
 if(sure<50) sure=50; // SÜRE 50 DEN KÜÇÜKSE 50 YAP
 if(sure>1000) sure=1000; // SÜRE 1000'DEN BÜYÜKSE 100 YAP
}

ileri(short adim)
{
for(i=0;i<adim;i++)
 {
 PORTC=1;bekle(sure);
 PORTC=2;bekle(sure);
 PORTC=4;bekle(sure);
 PORTC=8;bekle(sure);
 }
}

geri(short adim)
{
for(i=0;i<adim;i++)
 {
 PORTC=8;bekle(sure);
 PORTC=4;bekle(sure);
 PORTC=2;bekle(sure);
 PORTC=1;bekle(sure);
 }
}

void main()
{
 ANSELB=0; // B PORTU DİJİTAL AYARLANDI.
 ANSELC=0; // C PORTU DİJİTAL AYARLANDI.
 TRISB=7; // B PORTU B0,B1,B2 PİNLERİ GİRİŞ AYARLANDI.
 TRISC=0;  // C PORTU ÇIKIŞ OLARAK AYARLANDI.
 sure=100; // BEKLEME SÜRESİ 100ms OLARAK AYARLANDI.
```

```
BASLA:
 ileri(10); // 10*4=40 ADIM İLERİ GİT.
 geri(10); // 10*4=40 ADIM GERİ GİT.
 goto BASLA; // BASLA DEĞİŞKENİNE GERİ DÖN.
}
```



Şekil 12.5. PIC 18F45K22 mikrodenetleyici step motor sürücü devresi

Kaynaklar

[1] Süzen, A. A., Ceylan, O., Çetin, A., & Ulusoy, A. Arduino Kontrollü Çizim Robotu. *Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *8*(Özel (Special) 1), 79-87, 2017.

[2] Demirtaş, M. Bilgisayar kontrollü güneş takip sisteminin tasarımı ve uygulaması. *Politeknik Dergisi*, *9*(4), 247-253, 2006.

[3] Altun, Y., Öztürk, Z., & Özüberk, H. Bulanık Mantık ve Arduino Kullanarak Step Motorun Hız Kontrolü. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, *4*(2), 2016.

[4] Şimşek, E., Bilgili, M., & Küçükatay, O. PROGRAMLANABİLEN MANTIKSAL DENETLEYİCİ (PLC) İLE ÇALIŞAN SPLİT KLİMA TASARIMI. *Engineer & the Machinery Magazine*, (675), 2016.

[5] Çayıroğlu, İ., & Şimşir, M. PIC ve Step Motorla Sürülen Bir Mobil Robotun Uzaktan Kamera Sistemi ile Kontrolü. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi, 24(1), 1-16, 2008.

[6] Chirikjian, G. S., & Stein, D. Kinematic design and commutation of a spherical stepper motor. IEEE/ASME Transactions on mechatronics, 4(4), 342-353, 1999.

[7] Bodson, M., Chiasson, J. N., Novotnak, R. T., & Rekowski, R. B. High-performance nonlinear feedback control of a permanent magnet stepper motor. IEEE Transactions on Control Systems Technology, 1(1), 5-14, 1993.

[8] Koç, S. Rulmanlı vidalı mil tahrikli robot tasarım ve imalatı/Bearing ball screw driven robot design and manufacturing (Doctoral dissertation, 2018.

[9] Step motor nasıl çalışır? Erişim Tarihi: 26/10/2019 https://www.motorobit.com/blog/icerik/step-motor-nedir-nasil-calisir

[10] Çayıroğlu, İ., & Şimşir, M. PIC ve Step Motorla Sürülen Bir Mobil Robotun Uzaktan Kamera Sistemi ile Kontrolü. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi, 24(1), 1-16, 2008.

[11] Şahin, H., & Dedeoğlu, K. S. MikroC ve PIC18F4550. Altaş Yayıncılık, 2012.

[12] Sensor modules, Erişim Tarihi: 26/10/2019 https://www.electronicwings.com/sensors-modules/stepper-motor

[13] Çayıroğlu, İ., & Şimşir, M. PIC ve Step Motorla Sürülen Bir Mobil Robotun Uzaktan Kamera Sistemi ile Kontrolü. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi, 24(1), 1-16, 2008.

[14] Öztürk, R. Adım motorlarının telefon hatları aracılığı ile uzaktan kontrolü/Remote control of stepper motors over telephone lines, 2009.

[15] Çotuk, H. PIC mikrodenetleyiciler için gerçek zamanlı işletim sistemi (Master's thesis, TOBB Ekonomi ve Teknoloji Üniversitesi-Fen Bilimleri Enstitüsü-Bilgisayar Mühendisliği Anabilim Dalı), 2008.

BÖLÜM 13

13. MİKRODENETLEYİCİ İLE GERÇEK ZAMAN SAATİ (RTC) UYGULAMASI

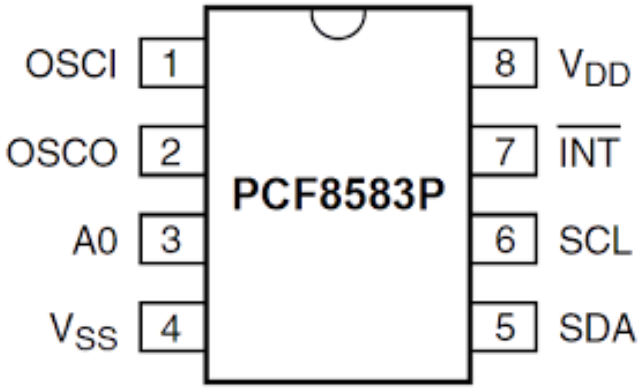
13.1. GİRİŞ

Mikrodenetleyici ile gerçekleştirilen uygulamalarda gerçek zaman parametresini bilmek çok fazla önem taşımaktadır. Örneğin; sıcaklık, basınç, nem, yağış ve hava olaylarının meteorolojik aletler ile ölçüldüğü ve ölçümün yapıldığı tarih ve saat bilgisini gerçek saate uygun ve doğru olarak verilmesi gerekmektedir [1,2]. Mikrodenetleyici'nin dahili saat devresi ile ürettiği saniye, dakika, saat, gün, ay, yıl birimleri ile gerçek zaman saati elde edilmektedir. Ancak dahili saat devresinin hatası ile bir yıl boyunca ulaşacağı birikmiş hata miktarıdır. Hatta geliştirilen özel amaçlı Philipsin PCF8583 entegresinin bile gecikme hataları olabilir. Bu nedenler en uygun uygulama özel entegreyi kendi güç kaynağı ile birlikte kullanmak olur ki bu durumda ölçülen zaman gerçek zamana eşit olmaktadır. PCF8583 entegresinin zaman doğruluğu (Accuracy) 1 saniye/gün'dür. Dolayısıyla bir yıllık sapma 6 dakikayı geçmez ki, bu da birçok gerçek zaman uygulamasının tolare ettiği bir hata büyüklüğüdür.

13.2. Gerçek Zaman Saati (Real Time Clock/RTC)

RTC (Real Time Clock); gerçek zaman saati olarak Türkçeleştirilebilir. Elektronik sistemlerde zamanın doğru bir şekilde algılanmasını sağlar ve algılanan bu zamana göre istenilen işlemlerin yapılmasına imkân verir. Gün, hafta, yıl, ay, saat, dakika ve saniye olmak üzere 7 farklı zamanı desteklemektedir [3]. Fabrika veya iş merkezlerinde gündelik olarak yapılması istenen işlemler (ışıkların açılıp kapatılması, geçiş durumlarının kontrol edilmesi, fabrika veya iş merkezinin ısıtma ve soğutma işlemlerinin yapılması.) RTC yardımıyla yaptırılabilir. Bu sayede insanların günlük olarak yapması gereken işlemler otomatikleştirilebilirken, sadece otomatik olarak kontrol edilebilirliğin yanında verimliliği de arttırmaktadır. Gerçek zaman saati uygulamalarında kullanılan elektronik devre elemanları farklılıklar gösterebilir. Genel olarak DS1302, DS1307, DS3231 ve PCF8583 entegreleri kullanılmaktadır. Kullanılan entegrelerin farklılıklar göstermesinin sebebi zaman içinde meydana gelen zaman sapmasıdır [4]. Genel olarak zaman sapmasının en az olduğu yer olan internet üzerinde çalışan kişilerin oluşan bu zaman sapmalarını göz önünde bulundurması gerekmektedir [4]. Bu entegreler arasında bulunan, zaman sapmasının 1-2 dakika olduğu DS1307

entegresinin 2100 yılına kadar geçerli olan saniye, dakika, saat, ayın günü, haftanın günü, ay ve yıl bilgilerini vermesinin yanında CLK sinyallerinin gönderildiği bir mikrodenetleyici de bulunmalıdır. RTC (Real Time Clock)'lerin "saat" gibi çalışabilmeleri için 32,768 kHz hızında kristal osilatöre sahiptir [6]. Bu entegrelerden PCF8583 gerçek zaman saati entegresi; 256 word 8 bitten oluşan 2048 bitlik CMOS RAM'a sahip olmakla birlikte adres ve veri bilgilerini işlemciye Philips firmasının geliştirmiş olduğu I^2C (Inter-Integrated Circuit) haberleşme protokolü sayesinde iki hattan gönderir [7].



Şekil 13.1. PCF8583P Gerçek Zaman Saati Entegresi (RTC)

Şekil 13.1'de elektronik devrelerde gerçek zaman saati (RTC) olarak kullanılan PCF8583P entegresine ait pin yapıları gösterilmiştir. Görüldüğü üzere 8 adet pini bulunmaktadır. OSCI ve OSCO pinleri osilatör giriş ve çıkış pinleridir ve 32.768kHz kristal bağlanır. I^2C iletişiminde sadece SDA (Serial Data Line) ve SCA (Serial Clock Line) olmak üzere iki hat bulunmakta ve bu iki hat da pull-up dirençlerine ihtiyaç duymaktadır [8].

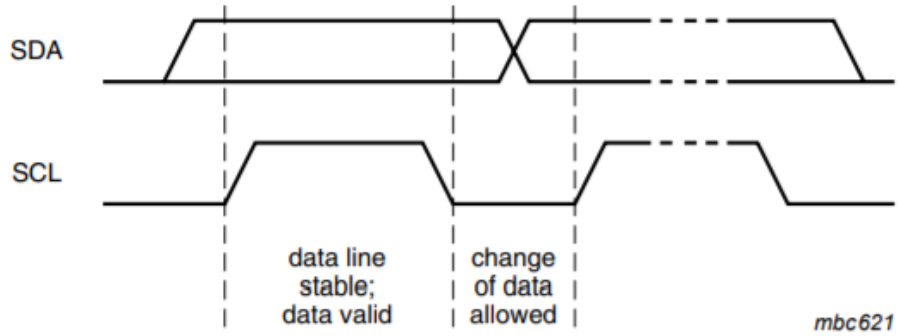
13.3. PCF8583P Entegre Gerçek Zaman Saati Özellikleri

PCF8583 entegresi elektronik devrelerde kullanılan gerçek zaman saati entegrelerinden birisidir. Tablo 13.1'de de görüldüğü üzere 8 adet pini bulunmaktadır. Gerçek tarih ve zamana ek olarak üzerinde alarm devresinin yanı sıra 256 bayt RAM bellek de bulunmaktadır. Güç kaynağı olan durumlarda +5V ile çalışabilirken güç kaynağının bulunmadığı durumlarda tarih ve saatin bozulmasını önleyebilmek adına pil ile çalıştırmak da mümkündür. Harici bir pil yardımıyla beslenip saat ve zaman bilgilerini saklayabilen entegreler enerjisinden bağımsız olarak yıllarca saat bilgisini saklayabilirler [9].

Tablo 13.1. PCF8583P Gerçek Zaman Saati Entegresinin Pinleri ve Özellikleri

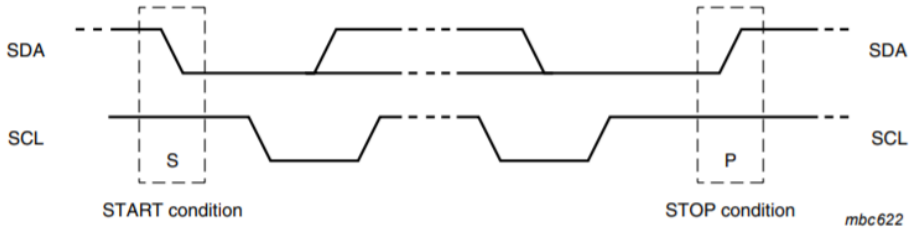
Pin Numarası	Sembol	Açıklama
1	OSCI	Osilatör Girişi
2	OSCO	Osilatör Çıkışı
3	A0	Adres Girişi
4	VSS	Toprak Besleme Gerilimi
5	SDA	Seri Veri Hattı
6	SCL	Seri Saat Hattı
7	INT	Açık Tahliye Kesme Çıkışı
8	VDD	Besleme Gerilimi

Her saat darbesi sırasında bir veri biti aktarılır. SDA hattındaki veri saat darbesinin 'HIGH' olduğu anda sabit kalmalıdır, çünkü bu sırada veri hattındaki değişiklikler kontrol sinyali olarak yorumlanır [7].



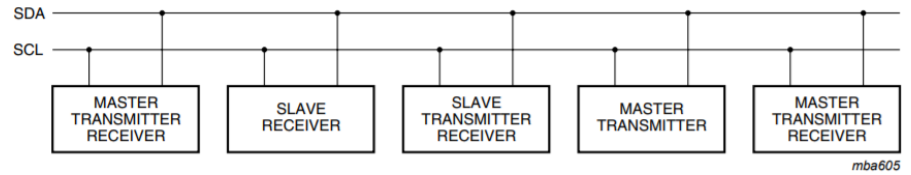
Şekil 13.2 I^2C Üzerinde Bit Transferi

Veriyolu meşgul olmadığında hem veri hattı SDA hem de saat hattı SCL 'HIGH' seviyesinde kalır. Saat hattı SCL 'HIGH' iken veri hattı SDA'nın HIGH-LOW geçişi Başla işlemini, SCL 'HIGH' iken SDA'nın LOW-HIGH geçişi Dur işlemini belirtir. (Şekil 13.3.)



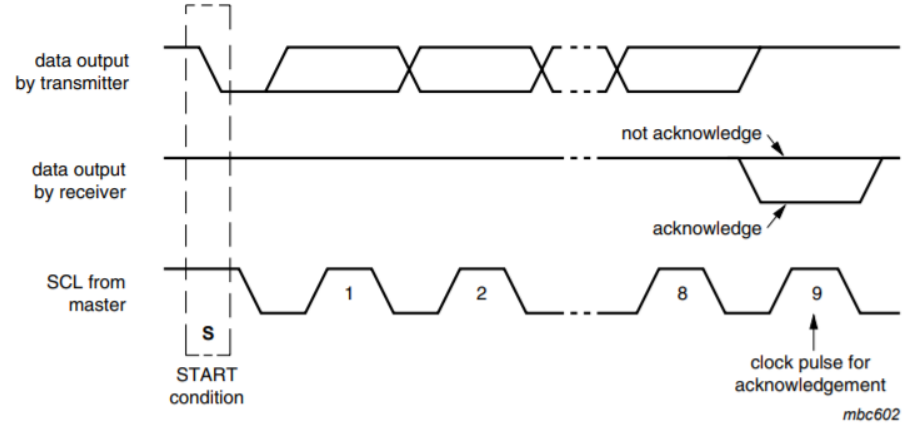
Şekil 13.3 I²C Üzerinde Start-Stop İşlemleri

I2C Bus üzerinde mesajı üreten gönderici (transmitter), mesajı alan da alıcı (receiver) durumunda bulunmaktadır. Mesajı kontrol eden sistem yönetici (master) iken yöneticinin kontrol ettiği birimler de köle (slave) konumundadır.

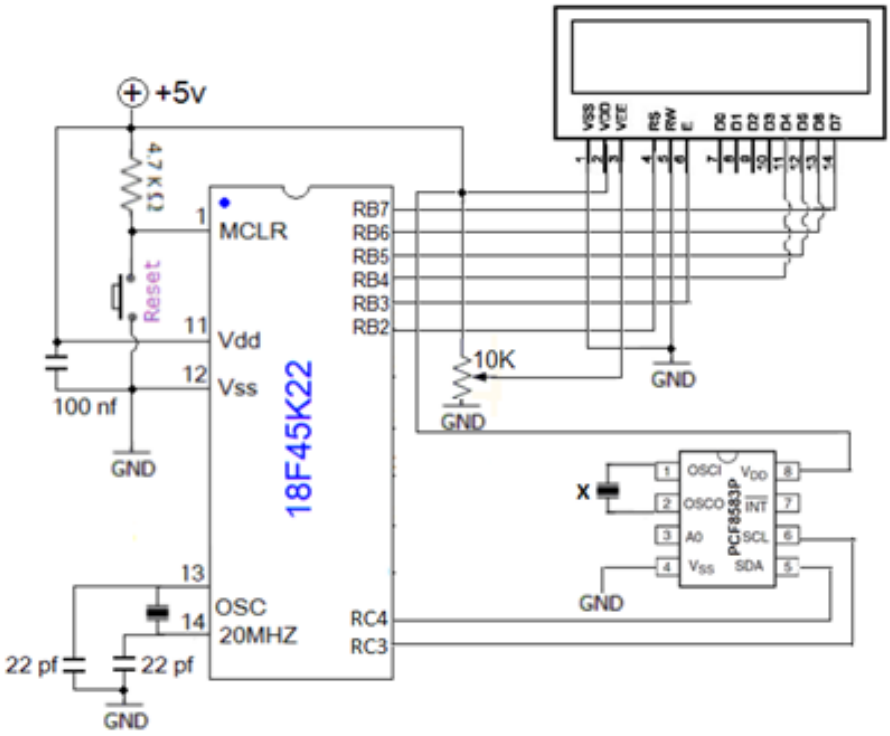


Şekil 13.4 I²C Bus Sistem Konfigürasyonu

Başla-Dur olarak belirtilen zamanlar arasında göndericiden (transmitter) alıcıya (receiver) gönderilen veri byte'ı sayısı sınırsızdır. 8 bitlik her veri byte'ı sonunda 1 bitlik 'Haberleşme' biti vardır.



Şekil 13.5. I²C Bus üzerinde haberleşmesi



Şekil 13.6. 18F45K22 PIC ile DS1307 RTC elektronik devre bağlantı şeması

Şekil 13.7'de gösterilen PIC ailesinden olan 18F45K22 mikrodenetleyicisinin DS1307 gerçek zaman entegresi ile kullanımına ait elektronik devre tasarımı görülmektedir. İlgili kodlar oluşturulup programın çalışması sağlandığında kullanılan bir adet LCD ekran üzerinde gün, ay ve yıl bilgisi görülecektir.

18F45K22 Mikrodenetleyicisi ve DS1307 Gerçek Zaman Saati Entegresi ile ilgili uygulama MikroC kodları aşağıda verilmiştir.

```
/* Mikrodenetleyici ile RTC Uygulaması */
char salise,saniye,dakika,saat,gun,ay;//zaman değişken
unsigned int sene;
unsigned int modul_adres, sene_simdiki=2012; // RTC
// modülünün bağlı olduğu adres ve yıl bilgisi
char deneme_sayar=0; // LCD modül bağlantıları
sbit LCD_RS at LATB4_bit;
sbit LCD_EN at LATB5_bit;
sbit LCD_D4 at LATB0_bit;
sbit LCD_D5 at LATB1_bit;
```

```
sbit LCD_D6 at LATB2_bit;
sbit LCD_D7 at LATB3_bit;
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// LCD modül bağlantılarını sonlandır.
void i2c_yaz(char veri,int adres)   //i²c hattı
// kullanılarak ilgili datayı ilgili adrese yazar.
{
  I2C1_Start();   //  I2C1 başlangıç sinyali gönderilir.
  I2C1_Wr(modul_adres);  //  ( entegre adresi + 'W')
  I2C1_Wr(adres);   // EEPROM'un işlem yapılacak adresi
  I2C1_Wr(veri);      // adrese yazılacak değer
  I2C1_Stop();      // I2C1 durdurma sinyali gönderilir.
  Delay_ms(100);
}
    void Baslangic_yapilandirmasi() {
 sene_simdiki  = 2017;
 modul_adres   = 0xA0;
    I2C1_Init(10000);       //  Soft I2C bağlantısı
  Lcd_Init();               // LCD ilklendirme fonksiyonu
  Lcd_Cmd(_LCD_CLEAR);     // LCD ekranını temizler.
  Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor ekranda gösterme
  LCD_Out(1,1,"Tarih:");    // LCD hazırlanır.
  Lcd_Chr(1,9,'/');
  Lcd_Chr(1,12,'/');
  LCD_Out(2,1,"Zaman:");
  Lcd_Chr(2,9,':');
  Lcd_Chr(2,12,':');
}
//RTC_yapilandirma için PCF8583 datasheet inceleme
void rtc_yapilandirma() { // RTC'nin başlangıç zamanını
  char yazilacak_data, yazilacak_adres;
  yazilacak_data=0x80;   //saymayı durdur.
  yazilacak_adres=0x00; //kontrol ve durum kaydı.
  i2c_yaz(yazilacak_data,yazilacak_adres);
  //salise
  yazilacak_data=0x00;
  yazilacak_adres=0x01;
  i2c_yaz(yazilacak_data,yazilacak_adres);
  //saniye
  yazilacak_data=0x31;
```

```
  yazilacak_adres=0x02;
  i2c_yaz(yazilacak_data,yazilacak_adres);
  //dakika
  yazilacak_data=0x22;
  yazilacak_adres=0x03;
  i2c_yaz(yazilacak_data,yazilacak_adres);
  //saat
  yazilacak_data=0x18;
  yazilacak_adres=0x04;
  i2c_yaz(yazilacak_data,yazilacak_adres);
  //yıl ve gun yapilandirmasi
  yazilacak_data=40;
  0x47;
  yazilacak_adres=0x05;
  i2c_yaz(yazilacak_data,yazilacak_adres);
  //ay ve gun yapilandirmasi
  yazilacak_data=0x04;
  yazilacak_adres=0x06;
  i2c_yaz(yazilacak_data,yazilacak_adres);
  //yapılandırma tamamlandı!!!
  //yapılandırma sonrası sayma tekrar başlat
  yazilacak_data=0x00;   //saymayı tekrar başlatır.
  yazilacak_adres=0x00;  //kontrol ve durum kaydı
  i2c_yaz(yazilacak_data,yazilacak_adres);
}
//---- PCF8583 RTC 'den tarih ve zaman bilgisi okunur.
void RTC_oku_PCF8583() {
  char okunan_data;
  I2C1_Start();  //  I2C1 başlangıç sinyali gönderilir.
  I2C1_Wr(modul_adres);  //( entegre adresi + 'W') yazma
  I2C1_Wr(2);     // EEPROM'un işlem yapılacak adresi
//(1.byte'tan başlanıyor.)
  I2C1_Stop();   // I2C1 durdurma sinyali gönderilir.
  I2C1_Start();  // I2C1 başlatma sinyali gönderilir.
  I2C1_Wr(modul_adres+1);  //(entegre adresi +'R') okuma
  okunan_data = I2C1_Rd(1);                                    //
2. byte okunur. (saniye)
  saniye = ((okunan_data & 0xF0) >> 4)*10 + (okunan_data &
0x0F); // saniye çevirimi
  okunan_data = I2C1_Rd(1);                                    //
3. byte okunur. (dakika)
  dakika =  ((okunan_data & 0xF0) >> 4)*10 + (okunan_data &
0x0F); // dakika çevirimi
  okunan_data = I2C1_Rd(1);                                    //
4. byte okunur. (saat)
```

```
  saat =   ((okunan_data & 0xF0) >> 4)*10 + (okunan_data &
0x0F);   // saat çevirimi
  okunan_data = I2C1_Rd(1);                                    //
5. byte okunur. (sene ve gun(rakamla))
  sene = sene_simdiki + ((okunan_data & 0xC0) >> 6);
// sene çevirimi
  gun = ((okunan_data & 0x30) >> 4)*10 + (okunan_data &
0x0F);
// gün çevirimi
  okunan_data = I2C1_Rd(0);                                     //
6. byte okunur. (ay ve gun(adıyla))
  ay =   ((okunan_data & 0x10) >> 4)*10 + (okunan_data &
0x0F);
 // ay çevirimi
  I2C1_Stop();    // I2C1 durdurma sinyali gönderilir.
}
//--------------------- LCD üzerinde görüntüleme
void LCD_goruntuleme() {
  Lcd_Chr(1, 7, (gun / 10)   + 48);
  Lcd_Chr(1, 8, (gun % 10)   + 48);
  Lcd_Chr(1,10, (ay / 10)  + 48);
  Lcd_Chr(1,11, (ay % 10)  + 48);
  Lcd_Chr(1,13, ((sene / 1000) % 10) + 48);
  Lcd_Chr(1,14, ((sene / 100)  % 10) + 48);
  Lcd_Chr(1,15, ((sene / 10)   % 10) + 48);
  Lcd_Chr(1,16, (sene % 10)          + 48);
  Lcd_Chr(2, 7, (saat / 10)   + 48);
  Lcd_Chr(2, 8, (saat % 10)   + 48);
  Lcd_Chr(2,10, (dakika / 10) + 48);
  Lcd_Chr(2,11, (dakika % 10) + 48);
  Lcd_Chr(2,13, (saniye / 10) + 48);
  Lcd_Chr(2,14, (saniye % 10) + 48);
}
//-----------
void main() {
  ANSELB = 0; //PORTB pinlerini dijital yapılandırır.
  ANSELC = 0; // PORTC pinlerini dijital yapılandırır.
  Baslangic_yapilandirmasi();
  LCD_goruntuleme();
  Delay_ms(2000);
//aşağıdaki comment'i kaldırırsanız rtc_yapilandirma
fonksiyonu
      //içindeki değerler rtc entegresinin ilgili
kayıtçılarına yüklenir.
```

```
      // böylece istediğiniz değere entegrenizi kurmuş
olursunuz.
      // comment kalkarsa kod her çalışmaya başladığında
      // rtc_yapilandirma fonksiyonu işleyeceği için buradaki
değerlerden başlar.
      //bu yüzden entegerenizi bir kere dogru degere
kurduktan sonra bu satiri comment'leyiniz.
      //rtc_yapilandirma();   // rtc'nin başlangıç zamanını
ayarlar.
  while (1)            // sonsuz döngü
  {
    RTC_oku_PCF8583();    //RTC'den veri okuma
    LCD_goruntuleme();
    Delay_ms(1000);
       if (deneme_sayar==0)
         deneme_sayar=1;
       else
         deneme_sayar=0;
    //Lcd_Chr(2,16, deneme_sayar+48);
  }
}
```

Kaynaklar

[1] Kaya, E. C., Çoruh, B., Bayrak, T., Koçoğlu, A., & Koçak, O. Tıbbi Amaçlı Sıcaklık, Nem, Basınç Ölçme ve İzleme Sistemi Tasarımı Measurement and Monitoring System Design of Temperature, Humidity and Pressure for Medical Usage.

[2] BETİ Mikrodenetleyici Eğitim Seti Kullanım ve Deney Kitabı,

[3] Yuklemeler, Erişim Tarihi: 26/10/2019 http://files.beijerelektronik.com.tr/yuklemeler/15.pdf

[4] Gerçek zaman saati, RTC, Erişim Tarihi: 26/10/2019 https://lezzetlirobottarifleri.com/ds1307-gercek-zaman-saati-rtc/

[5] Aydoğan, H., Aras, F., & Karakaş, E. Çok Noktalı Sıcaklık Sensör Sisteminin Tasarımı ve Gerçekleştirilmesi,

[6] Bakır, H. Fotovoltaik sistem entegreli akıllı şebeke için ZigBee aygıtları ile enerji kontrolü ve izlemenin gerçekleştirilmesi (Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, 2016.

[7] PCF8583 Zaman ve Tarih Entegresi Katalog Bilgileri, Erişim Tarihi: 26/10/2019 http://datasheetcatalog.com

[8] Gerçek zaman saati, Erişim Tarihi: 28/10/2019 http://esersenturk.blogspot.com/2015/07/gercek-zamanli-ayarlanabilir-saat-tarih.html

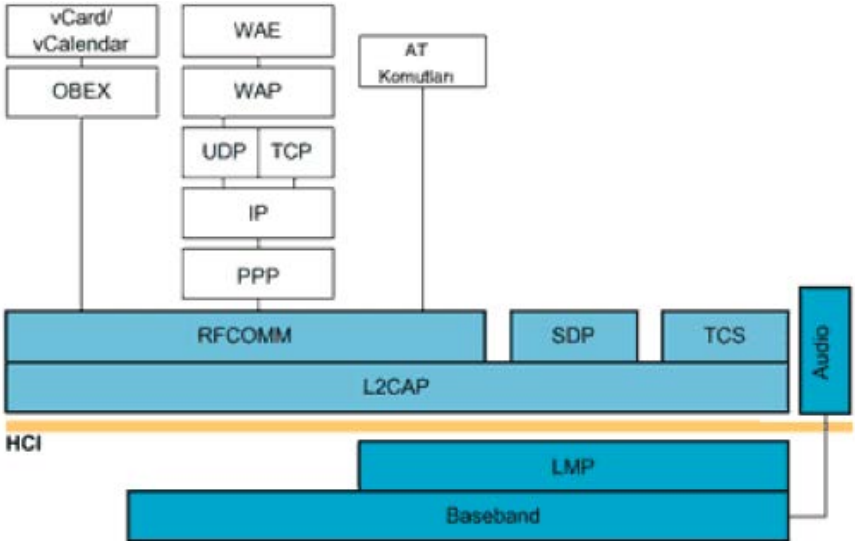
[9] Bakırcılar, S., & Özcerit, A. Programlanabilir CPLD tabanlı akıllı mikrodenetleyici eğitim seti tasarımı ve uygulaması. Sakarya University Journal of Science, 19(2), 123-133, 2015.

BÖLÜM 14

14. MİKRODENETLEYİCİ İLE KABLOSUZ HABERLEŞME

14.1. Kablosuz Haberleşme

Kablosuz haberleşme ağları, son yüzyıllarda iletişim kaynağı olarak vazgeçilmez bir parça haline gelmiştir. Askeriye ve acil servisler de kablosuz haberleşmeyi ilk olarak kullanmışlardır. Kablosuz haberleşmenin temeli iki veya daha fazla nokta arasında kablo kullanılmadan ses, görüntü ve datanın taşınması esasıdır [1]. Bir cihazdan diğer cihazlara iletişim sağlayabilmek için kablo yerine bluetooth kullanılmaktadır ve hem sesin hem de verinin eş zamanlı bir şekilde iletilmesini sağlamaktadır [1,2]. Bluetooth genel anlamda bir RF alıcı ve vericisinden oluşur. Bluetooth kendi içerisinde zamanlayıcıya sahip olmakla birlikte kendi içerisinde sekronizasyon sağlamak amacıyla senkronizeye kaynaklık eden bir cihaz master ve senkron hale gelmesi ile slave olarak adlandırılmaktadır [3]. Günümüzde akıllı cihazlar yaygın olarak kullanılmakta ve işlem kapasiteleri yüksek olmasından dolayı farklı iletişim teknolojilerine sahiptirler. Bu iletişim teknolojilerinden çevresinde bulunan cihazlarla iletişim halinde bulunmaktadırlar. Bluetooth az miktarda güç tüketimi, herhangi bir lisans gerektirmeyen, maliyetinin ucuz olması ve iletişim mesafesi olarak yaklaşık 1-100 metre arasında etkin bir şekilde veri alış-verişi yapabilme kabiliyetine sahiptir [4-5]. Bluetooth haberleşme protokolü 2.40-2.48 GhzISM (Industrial, Scientific, Medical) bandından haberleşme sağlayan bir protokoldür [6-8]. Buletooth, IEEE 802.15.1 standardına dayalı kısa mesafeli olarak çevresinde bulunan el cihazları, kişisel bilgisayarlar ve cep telefonları gibi aygıtlarla iletişim sağlayabilmektedir [2]. Şekil 14.1'de Bluetooth protokol yapısı gösterilmektedir.



Şekil 14.1 Bluetooth protokol yapısı [9]

Bluetooth versiyonlarının gelişimiyle veri iletişim hızı olarak en yüksek 1 Mbps kadar destek veren Bluetooth 1.1 ve 1.2 versiyonları tanınmıştır. 2004 yılında veri hızını artırmak için Bluetooth 2.0 piyasaya sürülmüştür. 2007 de ise Bluetooth versiyon 2.1 benimsenmiş ve yeni işletim özelliklerini kapsamaktadır. Bu özellik ile veri iletişim hızı maksimum olarak 3 Mbps 'a çıkarılmıştır. Bluetooth 3.0 + ile veri iletim hızını 24 Mbps 'a kadar çıktığı belirtilmiştir. Bluetooth 4.0 ile tanıtımı yapılmıştır. Aynı zamanda Bluetooth 3.0 100x olarak bilinmekte ve veri iletim hızı olarak 200-400 Mbps'a kadar çıktığı gözlemlenmektedir [2,10-12]. Tablo 14.1'de bluetooth versiyonları gösterilmektedir.

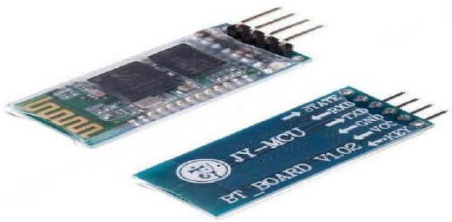
Tablo 14.1. Bluetooth Versiyonları [13]

Bluetooth Sürümü	Çıkış Yılı	Aktarım Hızı	Etkili Mesafe	Yazılım Sürümü
1.B	1999	1Mbps	10 m	LMP 0.x
1.1	2002	1 Mbps	10 m	LMP 1.x
1.2	2003	1 Mbps	10 m	LMP 2.x
2.0+EDR	2004	2 Mbps	10 m	LMP 3.x
2.1+EDR	2007	3 Mbps	10 m	LMP 4.x
3.0+HS	2009	24 Mbps	10 m	LMP 5.x
4.0	2010	24 Mbps	10 m	LMP 6.x
4.1	2013	24 Mbps	10 m	LMP 7.x
4.2	2014	24 Mbps	10 m	LMP 8.x
5.0	2016	48 Mbps	50 m	LMP 9.x

Bluetooth haberleşme protokolünde güvenlik olarak 3 temel sistem tanımı yapılmıştır. Bunlar gizlilik, kimlik doğrulama ve yetkilendirme olarak isimlendirilmektedir. Ayrıca bluetooth sistemlerinde 4 farklı güzenlik modu tanımlanmış ve güvenlik modu1, güvenlik modu2, güvenlik modu3, güvenlik modu4 olarak adlandırılmaktadır. Kimlik doğrulaması SAFER+ algoritmasına dayalı olarak bluetooth ağlarda yaygın olarak kullanılmaktadır [1-2,12,14].

14.2. HC-06 Bluetooth Modülü

HC-06 Bluetooth modülü, bluetooth versiyonlarından 2.0'ı desteklemektedir. Ayrıca 2.4 GHz frekans bandından haberleşme sağlamaya imkân tanımaktadır. Kapalı bir ortamda yaklaşık olarak 10 metreye kadar iletişim sağlamakta, açık alanda ise yaklaşık olarak 30 metreye kadar haberleşme sağlamaya imkân sunmaktadır. Modülün senkronizasyon hızı 1Mbps/1Mbps, asenkronizasyon hızı 2.1 Mbps/160Kbps iletişim hızına sahiptir. Ayrıca modülün üretilmesiyle birlikte olarak gelen şifreleme ve kimlik doğrulama yazılımı içerisinde vardır. HC06 Bluetooth modülü 1.8V ile 3.6V arasındaki gerilim değerlerinde ile besleme yapılmalıdır ve 50 mA akım çekmektedir. Boyut olarak 43X16X7 ölçülerindedir. Şekil 14.2' de HC-06 Bluetooth modülü gösterilmektedir [16].

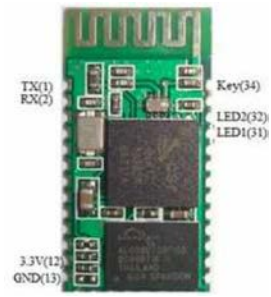


Şekil 14.2 HC-06 Bluetooth modülü

HC-06 Bluetooth modülü herhangi bir bluetooth haberleşme sağlayan cihazlara kolaylıkla bağlantı sağlamaktadır. Seri haberleşmenin en kolay yollarından biri olması sebebiyle haberleşmelerde yaygın olarak kullanılmaktadır [17-18].

14.3. HC-05 Bluetooth Modülü

HC-05 sivil düzeyde kullanılmakta olan bluetooth seri arayüz modülüdür. Fakat endüstriyel düzeyde kullanımı yoktur. Modül seri portu bluetooth'a dönüştürmek için kullanılmaktadır. Genelde iki moda sahiptirler. Bunlar ana ve bağımlı cihaz olarak adlandırılmaktadır. HC-05 kullanıcıları, cihazın ana veya bağımlı moduna AT komutlarıtla ayarlama yapabilme avantajına sahiptirler. Bluetooth seri modül temel anlamda seri bağlantı noktalarında bulunan hattan değişim yapmaktır. Cihazların eşleşme yapması durumunda aralarında bağlantı kurabilmektedir. Bluetooth bağlantısında, birbirleri ile iletişim sağlamak için seri modülü kullanabilir ve alınan ve iletilen sinyalleri içeren bir seri port hat bağlantısı eşdeğer durumdadır. Şekil 14.3'de HC-05 Bluetooth modülü gösterilmektedir [19].



Şekil 14.3 HC-05 Bluetooth Modülü

14.4. HC-05 Ve HC-06 Ortak Özellikleri

- 2,4 GHz haberleşme frekansı (ISM)
- Hassasiyet: ≤-80 dBm
- Çıkış gücü: ≤+4 dBm
- Asenkron hız: 2,1 MBps / 160 KBps
- Senkron hız: 1 MBps / 1 MBps
- Çalışma gerilimi: 1,8 - 3,6 V (Önerilen 3,3 V
- Akım: 50 mA
- Kimlik doğrulama ve şifreleme [20]

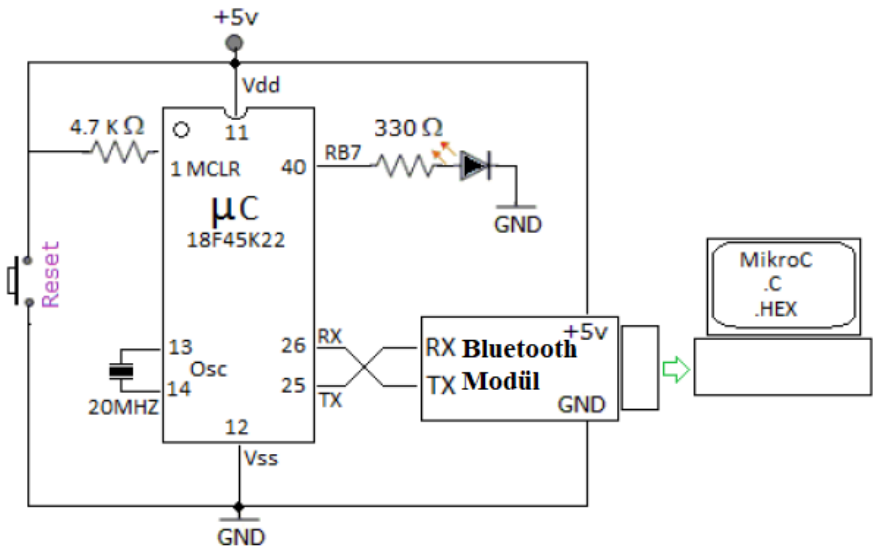
18F45K22 PIC devresinde bluetooth modül bağlantısı yapılmaktadır. Daha sonra Mikro C devresinde PIC kodları yazılıp, PIC kodlarının ".HEX" dosyası 18F45K22 PIC'in içerisine gönderilir. Bilgisayar ile Bluetooth modülü birbirleri ile haberleşme sağlaması için bağlantısı kurulmalıdır. Haberleşme seri olarak gerçekleşir. Haberleşme sağlamak amacıyla PIC kodları yazılırken UART kodları ile bütünleşik olarak yazılır.

Aşağıda yer alan MikroC uygulama, bilgisayar ile bluetooth modül arasında ASCII kodları ile haberleşme sağlayarak led yakıp, söndürme işlemidir. Mikro C programında RB7 portuna direnç ve led bağlantısı olduğu ve bluetooth haberleşme ile gelen koda göre ledin yanması veya sönmesini sağlamaktır. RX ve TX bluetooth modül ile bağlantısı yapılır.

```
#define LED_ON PORTD.B7=1
#define LED_OFF PORTD.B7=0
unsigned short i;
char txt[6];
char DRX; //UART DEĞİŞKENİ RECEIVE
void main()
{
ANSELA=0; //PORTA DIJITAL AYARLANDI !!!!
ANSELD=0; //PORTD DIJITAL AYARLANDI !!!!
ANSELB=0; //PORTB DIJITAL AYARLANDI !!!!
ANSELC=0; //PORTC DIJITAL AYARLANDI !!!!
ANSELE=0; //PORTE DIJITAL AYARLANDI !!!!
TRISA = 255;
TRISB = 0;
TRISC = 0;
TRISD = 0b00001100;
TRISE = 0;
Uart1_Init(9600); //UART 9600 BAUD AYARLANDI
```

```
PORTD.B0=1; PORTD.B1=1; PORTD.B7=1;
Delay_ms(2000);
PORTD.B0=0; PORTD.B1=0; PORTD.B7=0;
Uart1_Write(65+66);
Uart1_Write_Text("\r\n SERI HABERLESME v1.1\r\n");
Uart1_Write_Text("HAZIR >\r\n");

while(1)
{
  if(UART1_Data_Ready()==1)
   {
      DRX=UART1_Read();
      UART1_Write('#');
}
   if(DRX=='1') { PORTD.B7=1;Uart1_Write_Text("LED ON
\r\n");}
   if(DRX=='2') { PORTD.B7=0;Uart1_Write_Text("LED
OFF\r\n");}
    if (PORTD.b2==1)
{
Uart1_Write_Text("B2 BUTON.\r\n");
Delay_ms(1000);
}
if (PORTD.b3==1)
{
Uart1_Write_Text("B3 BUTON.\r\n");
Delay_ms(1000);
}
DRX=0;
}
}
```

18F45K22 PIC devresinde bluetooth modül bağlantısı ile LED yanıp söndüren uygulamanın elektronik devre şeması Şekil 14.4'de gösterilmiştir.



Şekil 14.4. 18F45K22 PIC devresinde bluetooth modül bağlantısı

Kaynaklar

[1] S. Rackley, *Wireless Networking Technology*. 2007.

[2] K. V. S. S. S. S. Sairam, N. Gunasekaran, and S. Rama Reddy, “Bluetooth in Wireless Communication,” *IEEE Commun. Mag.*, vol. 40, no. 6, pp. 90–96, 2002.

[3] Bluetooth SIG. A look at the basics of bluetooth wireless technology, Erişim Tarihi: 26/10/2019 https://www.bluetooth.com/what-is-bluetooth-technology/

[4] Android Mobile Phone Controlled Bluetooth Robot using ARM7 Microcontroller. International Journal of Scientific Engineering and Technology Reseach, 14-17,2014.

[5] P. Oliveira and P. J. Matos. "A Code Generator for Bluetooth Low Energy Services". Lecture Notes on Software Engineering, Vol. 4, No. 1, 2014.

[6] Mikrodenetleyicilerde Açık Kaynak Deneyimi: Aurdiuno ve Köprülü Vinç Kablosuz Kontrol Uygulaması Akademik Bilişim, 2015.

[7] Mishra, Sanjeev Kumar, et al. "A compact dual-band fork shaped monopole antenna for Bluetooth and UWB applications." IEEE Antennas and Wireless Propagation Letters 10,627-630, 2011.

[8] Bray, J., Senese, B., McNutt, G., Munday, B., &Kammer, D. BlueTooth application developer's guide. Syngress publishing, 2001.

[9] Sokullu, R., & Poyraz, U. Kanal Kalitesine Bağlı Bluetooth İletişim Performansının İyileştirilmesi,2016.

[10] Bluetooth SIG. Bluetooth technology basics, Erişim Tarihi: 26/10/2019 https://www.bluetooth.com/what-is-bluetoothtechnology/bluetooth-technology-basics

[11] J. Padgette, K. Scarfone, and L. Chen. Guide to Bluetooth Security: Recommendations of the National Institute of Standards and Technology (Special Publication 800-121 Revision 1). CreateSpace Independent Publishing Platform, USA, 2012. ISBN 147816896X, 9781478168966.

[12] N. Sriskanthan, F. Tan, and A. Karande, “Bluetooth based home automation system,” Microprocess. Microsyst., vol. 26, no. 6, pp. 281–289, 2002.

[13] Sordum, Erişim Tarihi: 06.10.2019 https://www.sordum.net/45298/bluetooth-surumu-nasil-bulunur/

[14] Bluetooth SIG. Security, classic. URL https://developer.bluetooth.org/TechnologyOverview/Pages/Security.aspx

[15] Robolink Teknoloji, Erişim Tarihi: 06.10.2019 http://320volt.com/ccs-PIC-c-derleyici-programi-c-dili-ve-ccs-temel-esaslari/

[16] Tezel, C. Elektrik Tahrikli Mobil Kontrollü Tank Robotun Tasarımı ve Gerçekleştirilmesi (Master's thesis, İstanbul Gelişim Üniversitesi Fen Bilimleri Enstitüsü), 2017.

[17] User instruction bluetooth, Erişim Tarihi: 26/10/2019 http://www.rcscomponents.kiev.ua/datasheets/hc_hc-05-user-instructionsbluetooth.Pdf

[18] Orig, Erişim Tarihi: 26/10/2019 http://en.ardumotive.com/uploads/1/2/7/2/12726513/967404309_orig.jpg?179

[19] Krishna, B. M., Nayak, V. N., REDDY, K., Rakesh, B., KUMAR, P., & Sandhya, N. Bluetooth Based Wıreless Home Automatıon System Usıng Fpga. Journal of Theoretical & Applied Information Technology, 77(3), 2015.

[20] Gelecegi Yazanlar, Erişim Tarihi: 06.10.2019 https://gelecegiyazanlar.turkcell.com.tr/konu/arduino/egitim/arduino-201/bluetooth-ile-iletisim

Özgeçmişler

Öğr. Gör. Seyit Ahmet İNAN

1973 yılında Konya'da doğdu. İlk, ortaokulu, lise eğitimini Konya'da tamamladı. Lisans eğitimini 1997 yılında Süleyman Demirel Üniversitesi Fen Fakültesi Fizik bölümünde tamamladı. Öğr. Gör. olarak Isparta Süleyman Demirel Üniversitesinde devam etmektedir. Endüstriyel otomasyon, Elektronik,Uzaktan Data Transfer Yöntemleri ve RF Veri iletimi (TELEMETRİ), Enerji İzleme ve Kontrol Sistemleri,ISO9001 Kalite Yönetim Sistemi, ISO EN/IEC 17025 Akreditasyon,Tarimsal Mekanizasyon ve Sulama Sistemleri, Mikroişlemci Programla, PLC Programlama,GSM Kontrol Sistemleri,Proses geliştirme & PID Kontrol,UHF Band Telsiz ile Uzaktan Kontrol ve izleme,,Web Tabanlı Programlama (PHP), Su Kaynakları uzaktan izleme ve kontrol sistemleri üzerinde çalışmalar yapmaktadır.

Dr. Bekir AKSOY

1974 yılında Gönen-Isparta'da doğdu. İlköğretimi Gönen, Orta ve lise eğitimini ise Isparta'da tamamlaladı. Lisans eğitimini 1997 yılında Fırat Üniversitesi Bilgisayar öğretmenliğinde tamamladı. Yüksek Lisans ve doktora eğitimini Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsünde sırasıyla 2001 ve 2016 yıllarında tamamladı. 1997 yılında Süleyman Demirel Üniversitesi Senirkent Meslek Yüksekokulunda Öğretim görevlisi olarak göreve başladı. 2017 yılında Süleyman Demirel Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliğinde Yrd. Doç. Dr. Olarak göreve başladı halen Dr. Öğr. Üyesi olarak Isparta Uygulamalı Bilimler Üniversitesinde devam etmektedir. Görüntü işleme, büyük veri ve yapay zeka konularında araştırmalar yapmaktadır. SCI kapsamındaki dergilerde yayınlanmış 2 makalesi ve bu makalelerden 1 atıf almış olup h-indeksi 2'dir.

Dr. Koray ÖZSOY

1981 yılında Dinar-Afyonkarahisar'da doğdu. İlk, ortaokulu Dinar-Afyonkarahisar'da, lise eğitimini Isparta'da tamamladı. Lisans eğitimini 2004 yılında Gazi Üniversitesi Elektrik öğretmenliği ve 2013 yılında Süleyman Demirel Üniversitesi Makine Mühendisliğinde tamamladı. Yüksek Lisans ve doktora eğitimini Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsünde sırasıyla 2008 ve 2016 yıllarında tamamladı. 2004 yılında Öğretim görevlisi oldu ve halen görevine Öğr. Gör. Dr. olarak Isparta Uygulamalı Bilimler Üniversitesinde devam etmektedir. Otomatik Kontrol, Mekatronik, Eklemeli İmalat teknolojileri, 3B baskı teknolojileri, Ergiyik Biriktirme Yöntemi (FDM), konularında araştırmalar yapmaktadır. SCI kapsamındaki dergilerde yayınlanmış 3 makalesi ve bu makalelerden 10 atıf almış olup h-indeksi 3'dir. Ulusal dergilerde yayınlanmış 6 makalesi vardır. Uluslararası ve ulusal kongrelerde sunulmuş ve bildiri kitabında basılmış sırasıyla 9 ve 7 bildirisi vardır. Ulusal ve Uluslararası yayınevinde basılmış birer kitabı vardır. Değişik idari görevlerde bulunmuş olup Makine Mühendisleri Odasına kayıtlı üyedir.

Kitap, mesleki ve teknik eğitim veren orta ve yükseköğretim kurumlarındaki öğrenciler için kaynak olması düşünülerek hazırlanmıştır. Kitapta mikrodenetleyicilerin temelleri ve uygulama örnekleri akademik bir dille ele alınmıştır. Kitap, mikrodenetleyici ve programlama konusunda daha önceden hiçbir bilgisi olmayan bireyleri başlangıç düzeyinden alarak ileri düzeye kadar getirecek biçimde hazırlanmıştır. Kitap elektrik-elektronik, mekatronik ve bilgisayar teknolojilerinde eğitim gören öğrenciler ve bu alanlarda çalışan mühendisler için hazırlanmıştır.

Kitabın okuyucularına, akademiye ve bilime yararlı olacağını umuyoruz.

Saygılarımızla…

978-625-7029-34-6

İKSAD
Publishing House