T.C.
Niğde Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Ana Bilim Dalı

PIC MİKRODENETLEYİCİSİ KULLANARAK DENEYSEL BİR ENDÜSTRİYEL SİSTEMİN KONTROL EDİLMESİ

Halil Fikret Turgutlu

Yüksek Lisans Tezi

Danışman: Yrd. Doç. Dr. Murat Uzam

Şubat 2002

Fen Bilimleri Enstitüsü Müdürlüğüne,

Bu çalışma jürimiz tarafından ELEKTRİK-ELEKTRONİK ANABİLİM DALI'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Başkan : Doç. Dr. Saadetdin HERDEM (Niğde Üniversitesi).....

Üye : Yrd. Doç. Dr. Murat UZAM (Niğde Üniversitesi).....

Üye : Yrd. Doç. Dr. Murat AKSOY (Çukurova Üniversitesi).....

ONAY:

Bu tez 18../02./2002 tarihinde, Fen Bilimleri Enstitüsü Yönetim Kurulunca belirlenmiş olan yukarıdaki jüri üyeleri tarafından uygun görülmüş ve Enstitü Yönetim Kurulu'nun kararıyla kabul edilmiştir.

05../04./2002

Enstitü Müdürü

Doç. Dr. Aydın TOPÇU

ÖZET

PIC MİKRODENETLEYİCİSİ KULLANARAK DENEYSEL
BİR ENDÜSTRİYEL SİSTEMİN KONTROL EDİLMESİ

TURGUTLU, Halil Fikret

Niğde Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Ana Bilim Dalı

Danışman: Yrd. Doç. Dr. Murat UZAM

Şubat 2002, 135 sayfa

Bu çalışmada PIC16F84 mikrodenetleyicisi kullanarak bir deneysel endüstriyel sistem kontrol edilmiştir. Çalışmada kullanılan deneysel endüstriyel sistem parça tanıma, ayırma ve birleştirme işlemlerinin gerçekleştirilebildiği bir sistemdir. Deneysel endüstriyel sistemin farklı çalışma senaryoları için dört ayrı program geliştirilmiştir. Bu dört ayrı program, sonuçta tek bir entegre devre (PIC16F84) içerisine yüklenmiştir Endüstriyel sistem kontrolü için bir PIC mikrodenetleyicisinin kullanılmasıyla, tasarımı kolay ve maliyeti oldukça düşük bir kontrol sistemi elde edilmiştir.

Anahtar sözcükler : PIC16F84 mikrodenetleyicisi, Otomasyon ve Kontrol sistemleri.

SUMMARY

THE CONTROL OF AN EXPERİMENTAL INDUSTRIAL SYSTEM BY USING A PIC MICROCONTROLLER

TURGUTLU, Halil Fikret

Niğde University
Institute of Natural Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assist. Prof. Dr. Murat UZAM

February 2002, 135 pages

In this study, an experimental industrial system is controlled by using a PIC16F84 microcontroller. The experimental industrial system considered, represents component inspection, sorting and assembly processes. For different senarious of the experimental industrial system four programs are developped to control the system. These four programs are then implemented in one integrated circuit (PIC16F84). As aresult, an easy to design and very cheap control system is obtained, for the experimental industrial system by using a PIC microcontroller.

Key Words : PIC16F84 microcontroller, Automation and Control systems.

TEŞEKKÜR

Bu çalışmayı yöneten, endüstriyel sistem uygulamaların yürütülmesinde ve tez yazımı sırasında değerli katkılarını esirgemeyen danışmanım Yrd. Doç. Dr. Murat UZAM'a ve Niğde Üniversitesi Mühendislik Mimarlık Fakültesi Elektrik-Elektronik Mühendisliği Bölümü öğretim elemanları ve çalışanlarına şükranlarımı sunarım.

Ayrıca bu çalışma boyunca verdiği manevi desteklerinden dolayı sevgili anneme teşekkür ederim

İÇİNDEKİLER

ÇİZELGELER DİZİNİ

ŞEKİLLER DİZİNİ

SİMGE VE KISALTMALAR

ms,	Mili saniye
µs,	Mikro saniye
µF,	Mikro farad
A,	Amper
V,	Volt
<,	Küçük
>,	Büyük
≤,	Küçük eşit
≥,	Büyük eşit
Є,	Elemanıdır
Mhz	Mega hertz
Khz	Kilo hertz
Kb,	Kilo bayt
F,	File register
W,	Working register
C,	Komut saykıl sayısı
DKF,	Dahili komut frekansı
DKS,	Dahili komut saykılı
KİS,	Komut işleme süresi
TOS,	Top Of Stack, Stack registerin üst hafiza bölümü
PC,	Program sayıcı
k,	Sabit sayı veya etiket
d,	Destination, Komut ardından işlem sonucunun kaydedileceği register
b,	Bir file registerin herhangi bir biti
°C	Santigrad derece
± %	Hata payı

BÖLÜM I

GİRİŞ

Günümüzde el değmeden üretim veya başka bir deyimle otomasyon teknolojisinin hızla gelişmekte olduğunu görmekteyiz. Bu teknolojinin rağbet görmesinde, seri üretim, maliyet (işgücü)'in azalması, kalitenin artması gibi faktörler önemli rol almaktadır.

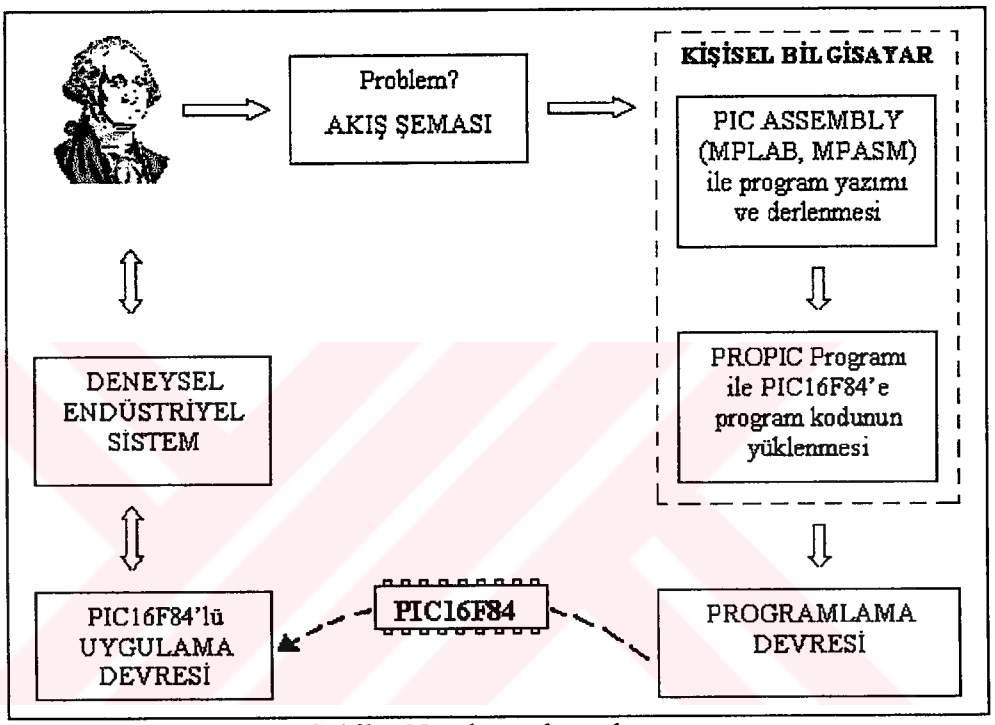
Bir otomasyon sisteminin en önemli kısmını, elektronik beyin olarak ta adlandırabileceğimiz, sisteme göre programlanabilen elektronik cihazlar oluşturmaktadır. Bu cihazların başında da yirmiyi aşkın firma çeşidiyle PLC (Programlanabilir Lojik Denetleyici) cihazları yer almaktadır. PLC'ler kolay programlanabilmeleri, simülasyon yapabilme imkanı ile programlarda hata riskini azaltmaları gibi özellikleri ile ilgiyi kendilerine çekmektedirler. PLC'lerden daha ucuz bir çözüm yolu olarak, mikroişlemci temelli sistemleri görmekteyiz.

Bu tez çalışmasında, bir endüstriyel sistemin kontrolünde, PLC cihazı veya mikroişlemcili başka bir sistemin yerine, bir PIC16F84 mikrodenetleyicisinin (www.microchip.com) kullanılması incelenmiştir. Bu çalışmada, bir PIC (Peripheral Interface Controller) mikrodenetleyicisi için program hazırlanması, hazırlanan programın mikrodenetleyiciye yüklenmesi ve uygulama devresiyle birlikte çalışmasına kadarki aşamalar, gerekli program ve malzemeler tek tek tanıtılmış, PIC programlamaya yeni başlayanların da faydalanabileceği bir çalışma ortaya çıkmıştır.

PIC, mikrodenetleyicilerinin böyle bir uygulamada kullanılması ile birlikte, PLC'lere göre çok daha ucuz ve diğer mikroişlemcili sistemlere göre çok daha sade ve az yer kaplayan, kullanılması kolay bir yapı meydana gelmektedir. Bu çalışmada bir PIC mikrodenetleyicisi için program yazmak üzere "PIC ASSEMBLY" dili kullanılmış olmakla birlikte, PIC mikrodenetleyicileri için program geliştirmek üzere, "PIC BASIC" (www.rentron.com), "PIC C" (www.ccsinfo.com) gibi yüksek seviyeli programlama dilleri veya PARSIC (www.parsic.de), PICBIT (Turgutlu, 2001) gibi görsel paket programlarda bulunmaktadır.

Şekil 1'de bu tezde yapılan çalışmaların özeti görülmektedir. Yapılan bu çalışmalarda öncelikle deneysel endüstriyel sistem için 4 farklı problem tanımlanmış ve bu problemlerin

çözümüne ilişkin akış şemaları çizilmiştir. Çizilen akış şemalarına göre, bilgisayar ortamında MPLAB ve MPASM programları kullanılarak PIC16F84 için PIC ASSEMBLY dilinde programlar hazırlanmıştır. Hazırlanan her program, PROPIC yazılımı ile çalışan programlama devresi kullanılarak PIC16F84 içerisine yüklenmiştir. Programlanan PIC16F84 mikrodenetleyicisi, Şekil 35'te verilen uygulama devresine taşınarak deneysel endüstriyel sistemin kontrolü sağlanmıştır.



Şekil 1 Yapılan çalışmalar

Bu tez Şekil 2'de görülen sıra dahilinde sunulmuştur. Buna göre;

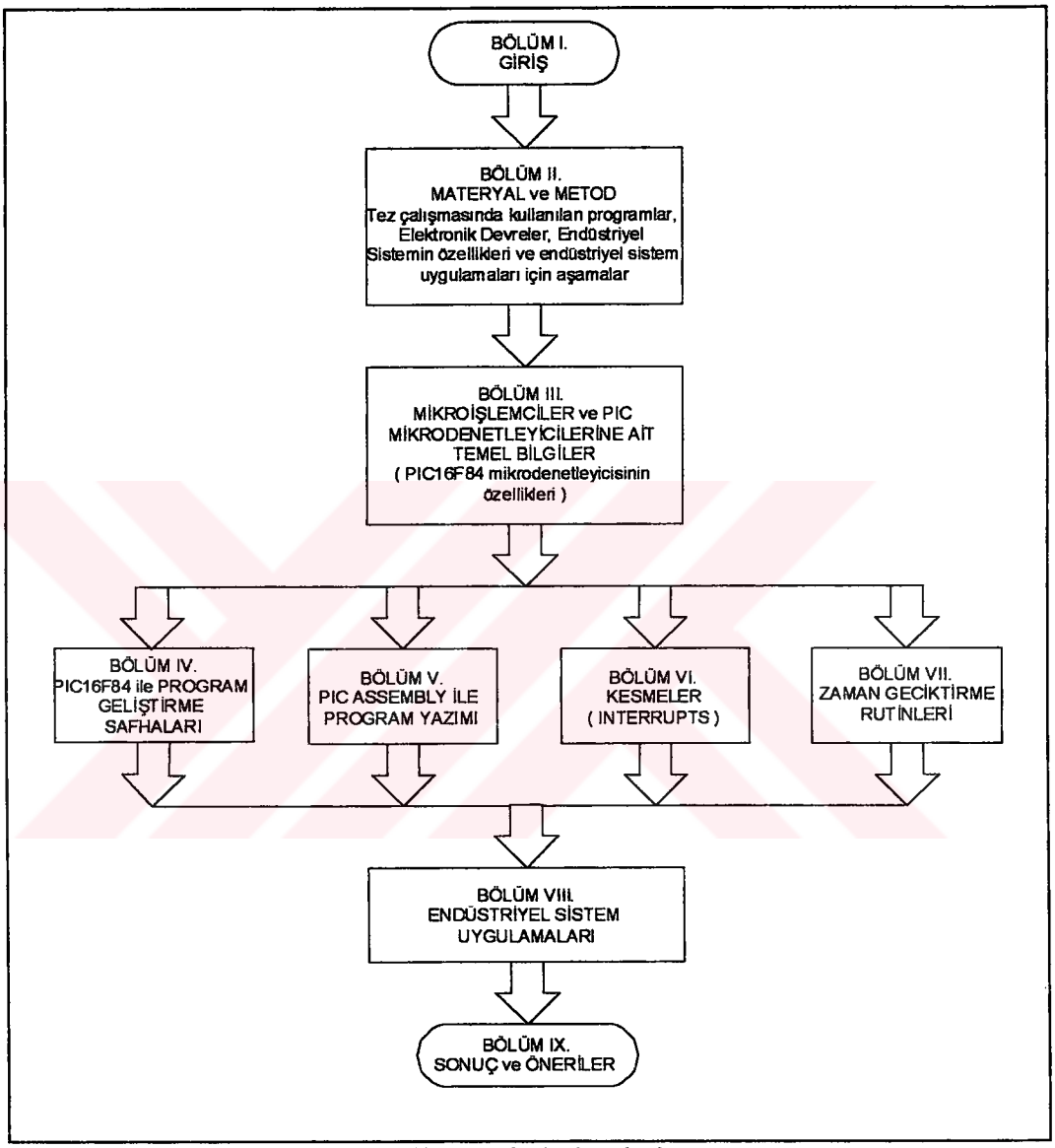
BÖLÜM II'de tez çalışmasında kullanılan programlar, elektronik devreler, endüstriyel sistemin özellikleri ve çalışmasına ait açıklamalar verilmiştir.

BÖLÜM III'de mikroişlemciler ve PIC mikrodenetleyicilerine ait temel bilgiler verilmiş ve PIC16F84 mikrodenetleyicisinin özellikleri hakkında detaylı açıklamalar yapılmıştır.

BÖLÜM IV, V, VI ve VII'de bir PIC mikrodenetleyicisi için program hazırlanmasından, PIC içerisine yükleninceye kadar geçen bütün aşamalar tek tek ele alınmıştır.Bu bölümlerde PIC ASSEMBLY dili yazım kuralları, komutların tanıtılması, kesme programlarının oluşturulması, zaman geciktirme programlarının hazırlanması gibi, program geliştirmek için gerekli bilgiler verilmiştir.

BÖLÜM VII'de deneysel endüstriyel sistem üzerinde yapılan değişiklikler ve bu sistem için 4 farklı problemin tanımlanmasına ve çözümüne ilişkin açıklamalar verilmiştir.

BÖLÜM IX ise sonuç ve öneriler bölümüdür.



Şekil 2 Tezin bölümleri

BÖLÜM II

MATERYAL ve METOT

Bu tez çalışmasında deneysel bir endüstriyel sistemin kontrolü için kullanılmış olan bütün materyalleri kısaca şu şekilde sıralayabiliriz;

1) PIC mikrodenetleyicisi (PIC16F84),
2) Programlama Birimi,
    a) Program geliştirme yazılımı (MPLAB, MPASM),
    b) Programlama yazılımı (PROPIC),
    c) Programlama devresi,
    d) Kişisel bilgisayar,
3) Uygulama devresi,
4) Deneysel endüstriyel sistem,
5) 4 farklı çalışma problemi.

Yukarıda sayılan materyalleri sırasıyla kısaca inceleyecek olursak;

2.1 PIC Mikrodenetleyicisi

Endüstriyel kontrol işlemini gerçekleştirmek için PIC16F84 mikrodenetleyicisi kullanılmıştır. Bu mikrodenetleyicinin tercih edilmesinin sebeplerinden bazıları şunlardır;

- Kontrol edilecek olan endüstriyel sistem için yeterli sayıda (13 adet) I/0 (giriş/çıkış) ucuna sahiptir.
- 1000 defaya kadar tekrar programlanabilen EEPROM (Flash) belleğe sahiptir.
- TMR0 yazılım kesmesine sahiptir.
- 4-10 Mhz arası yüksek bir çalışma frekansına sahiptir.

2.2 Programlama Birimi

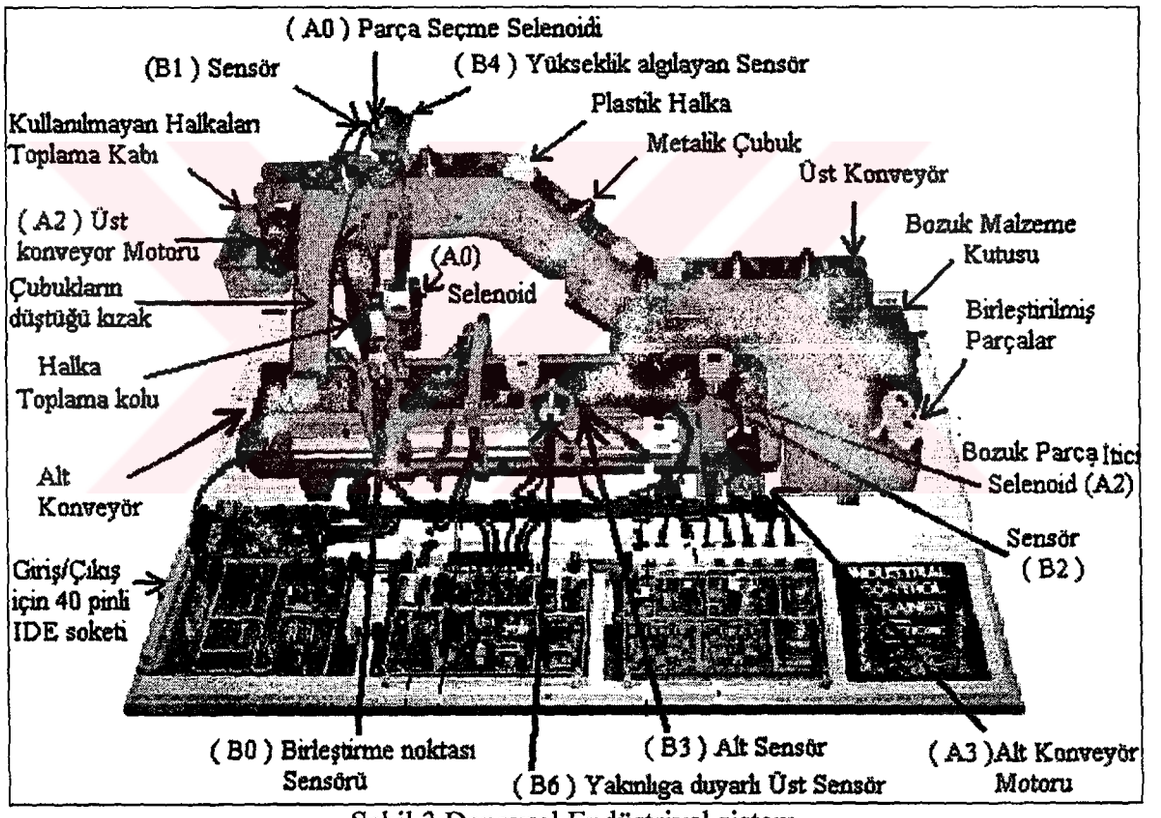
PIC mikrodenetleyicisi için programların hazırlanmasında "PIC ASSEMBLY" dili kullanılmıştır. Bu dilde program geliştirmek için gerekli olan MPLAB ve bu programla birlikte gelen MPASM programı http://www.microchip.com internet adresinden ücretsiz olarak temin edilebilmektedir.

Hazırlanan PIC programlarının PIC16F84'ün içerisine yüklenebilmesi için http://jaichi.virtualave.net/electr-e.htm internet adresinden yine ücretsiz olarak temin edilebilen PROPIC programı ve bu programla uyumlu olarak çalışabilen Şekil 20'de verilen, paralel port programlayıcı devresi kullanılmıştır.

2.3 Uygulama Devresi

Uygulama devresi, enerji ve osilatör girişleri gibi PIC'in çalışması için gerekli olan gereksinimleri üzerinde bulunduran ve PIC ile endüstriyel sistem arasındaki bağlantıyı sağlayan Şekil 35'te görülen, tarafımızdan gerçekleştirilen bir devredir.

2.4 Deneysel Endüstriyel Sistem



Şekil 3 Deneysel Endüstriyel sistem

Kontrol edilecek olan deneysel endüstriyel sistem, Şekil 3'te görülen ICT (Industrial Control Trainer), Endüstriyel Kontrol Eğiticisi olup, Niğde Üniversitesi, Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü Mikroişlemci Laboratuarı'nda bulunmaktadır.

Şekil 3'te görülen sistem parça tanıma, ayırma ve birleştirme işlemlerinin yapılması amacıyla geliştirilmiştir. Üst konveyör ve alt konveyör sırasıyla üst konveyör motoru (A2)

ve alt konveyör motoru (A3) ile çalıştırılmaktadır. Metal çubuklar ve plastik halkalar rastgele bir şekilde üst konveyör tarafından seçme işleminin yapılacağı seçme bölgesine taşınırlar. Metal çubuk ve plastik halkayı tanımlayıp birbirinden ayırt etmek için iki sensör (B1: proximity sensor, B4: infra-red reflective sensör) kullanılmaktadır. Parça seçme selenoidi (A0) ile plastik halkalar, halka toplama kolu üzerine aktarılırlar. Halka toplama kolu üzerine maksimum beş tane halka yerleştirilebilir. Metalik çubuklar ise üst konveyörün sonuna doğru yerleştirilmiş olan bir engele çarparak üst konveyörden alt konveyöre, kızak üzerinden aktarılırlar. Kullanılmayan ve üst konveyörün sonuna kadar giden plastik halkalar bu engele çarpmadan üst konveyör sonundaki bir kutuya düşerler. B0 sensörü (infra-red emitter/detector) birleştirme noktasının boş olup olmadığını tespit etmek için kullanılır. Eğer birleştirme noktası boş ise birleştirme selenoidi (A1) halka besleme kolundan birleştirme noktasına bir plastik halka aktarmak için kullanılır. Birleştirme noktası alt konveyörün biraz üzerine yerleştirilmiştir ve bir metalik çubuk alt konveyör üzerinden bu noktadan geçtiği anda burada bulunan plastik halka ile birleşir. Böylece metalik çubuk ile plastik halka birleştirilmiş olur. Birleştirilmiş iki parça üst üste gelerek birbirini tamamlar. Alt konveyör birleştirilmiş parçaları toplama kabına iletir. Birleştirilmiş veya birleştirilmemiş parçalar alt konveyör tarafından, birleştirilmiş parçaları toplama kabına giderken bu parçaların birleştirilmiş (sağlam) ya da birleştirilmemiş (hatalı) olduğu, test edilebilir. Bu iş için iki sensör (B3 ve B6) kullanılır. Eğer hatalı parça tespit edilmiş ise biraz ilerideki B2 sensörü ve A2 itici selenoidi kullanılarak reddetme işlemine tabi tutularak bozuk malzeme kutusuna aktarılır. Bu sayede toplama kabında sadece birleştirilmiş (sağlam) parçalar toplanır.

Mevcut olan bu sistemde bazı değişiklikler yapılmış ve sistem daha karmaşık kontrol uygulamaları için yeniden düzenlenmiştir. Yapılan ilk değişiklik, üst konveyörde parçaların ayrıldığı kısmın biraz önüne konulan metal algılayıcı sensördür. Ayrıca metal çubuk ile birlikte plastik çubuk, plastik halka ile birlikte metal halka sisteme ilave edilmiştir. Bu sayede sistemdeki birleştirilecek olan parçaların birleşme ihtimalleri artmış ve buna göre çeşitli programlar geliştirilerek değişik birleştirme uygulamaları yapılmıştır.

2.5 Farklı Çalışma Problemleri

Deneysel endüstriyel sistemde yapılan değişiklikler den sonra BÖLÜM VIII.'de verilen 4 farklı çalışma problemi için uygulamalar yapılmıştır.Endüstriyel sistem uygulamalarında yazılan her program için şu işlemler tekrarlanmıştır.

a) Problemin tanımlanması.

b) Programın hazırlanması,

c) Hazırlanan programın PIC'e yüklenmesi,

d) Programlanan PIC'in uygulama devresine takılarak, endüstriyel sistemin kontrol edilmesi.

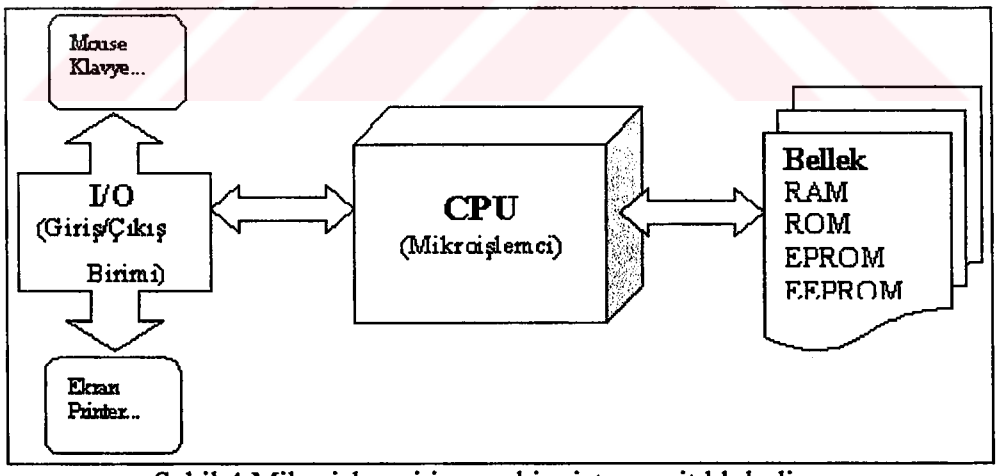
BÖLÜM III

MİKROİŞLEMCİLER ve PIC MİKRODENETLEYİCİLERİNE AİT TEMEL BİLGİLER

3.1 Mikroişlemciler

Bilgisayar donanımının vazgeçilmez parçaları olan mikroişlemciler, ön belleğine yazılan programı işleterek istenilen çıkışlara yönlendiren birimdir. Bir mikroişlemcinin görevlerini yerine getirebilmesi veya bir sayısal bilgisayar oluşturabilmek için bazı ek yardımcı elemanlara ihtiyaç vardır. Bunlar ;

- I/O (Input/Output) Giriş/Çıkış Birimi,
- Memory (Bellek),
- Çevresel Üniteler (monitör, printer, klavye, modem vb.) olarak sınıflandırılabilir.



Şekil 4 Mikroişlemci içeren bir sisteme ait blok diyagram

Şekil 4'te mikroişlemcili bir sisteme ait blok diyagram verilmiştir. Her bir temel kısım en basitten en karmaşığa kadar çeşitlilik göstermektedir.

I/O (Giriş/Çıkış) Birimi: Sayısal, Anolog ve özel fonksiyonlar olmak üzere üç farklı gruba ayrılabilir. I/O birimi mikroişlemcinin dış dünya ile ilişkisini sağlar. Mikroişlemciye verilen ve işlemlerden alınan veriler I/O hatları üzerinden sağlanır.

CPU (Central Proccesing Unit-Merkezi işlem birimi): Sistemde aritmetik ve lojik işlemleri gerçekleştiren ve bütün verileri idare eden en önemli kısımdır. 4, 8, 16, 32, ... bitlik veri formatında çalışabilir.

Bu birimlerin birbirleri ile haberleşmesi ise BUS adı verilen yollarla olmaktadır. Bir mikroişlemcide temelde kullanılan üç yol vardır (Özkan,1995).

Veri Yolu (Data Bus): Bu yol ; işlemci, bellek ve çevre birimleri arasında veri iletmek için kullanılır.

Adres Yolu (Address Bus): Bu yol, işlemcinin program komutlarına ve veri saklama alanlarına erişimi sağlayan bellek adreslerini, ROM ve RAM belleklerine göndermek için kullanılır.

Denetim yolu (Control Bus): Bu yol, RAM belleğe veri yazıldığı veya ondan veri okunduğuna dair bilgi vermek gibi, denetim amaçları için kullanılır. Bu yol aynı zamanda kesmelerin (interrupt) kullanımına olanak tanıyan bağlantıları da içerir. Tipik bir mikroişlemci komutunun yürütülmesi her üç yolunda kullanımını gerektirebilir. Böylelikle kullanılan ek devreler artarak maliyet yükselir ve tasarım çok karmaşık hal alır. İşlemci komuta erişmek için ilk olarak, komut adresini adres yoluna koyar. İkili kodlardan oluşan bu adres, buna karşılık gelen bellek konumu tarafından tanınır ve bu konum istenen komutu veri yolundan işlemciye gönderir. Örneğin eğer bu komut, verinin işlemciden gönderilmesini ve bir RAM konumunda saklanmasını gerektiriyorsa işlemci, adres yolunu istenen konumu belirtmek, veri yolunu veriyi iletmek ve denetim yolunu da RAM'a yazıyor olduğunu belirtmek için kullanılır.

Mikroişlemciler temel olarak iki görevi yerine getirirler (Çelik,1997). Bunlar ;

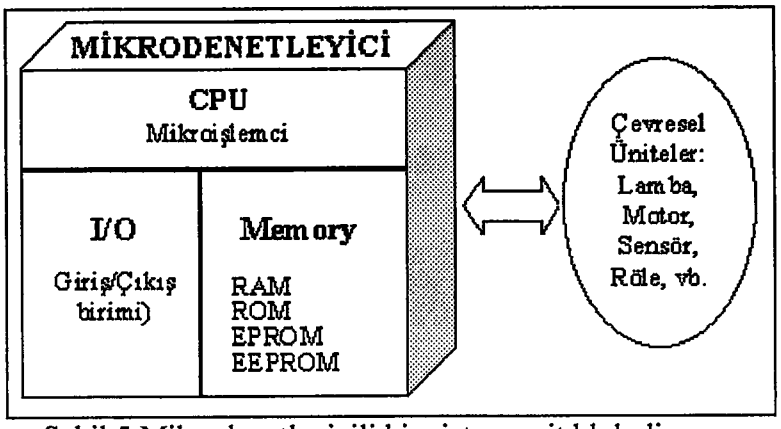
- Fetch : Komutların ana bellekten alınması,
- Execution : Komutların icra edilmesidir.

Mikroişlemcilerin komut icra birimi (CPU) olması sayesinde, yazılım yoluyla yönlendirilebilen bir donanım elde edilebilmekte ve çok çeşitli uygulama alanları bulabilmektedir.

3.2 Mikrodenetleyiciler

Bir mikroişlemcili sistemin (Ör: Kişisel Bilgisayar) içerisinde bulunması gereken temel bileşenlerin (CPU, RAM, I/O) tek bir chip içerisinde üretilmiş biçimine mikrodenetleyici veya mikrokontrolör (Microcontroller) denilmektedir. Günümüzde mikrodenetleyiciler, otomobillerde, kameralarda, cep telefonlarında, fax-modem cihazlarında, fotokopi, radyo,

televizyon, bazı oyuncaklar ve daha bir çok bilgisayar teknolojisi gerektiren alanlarda kullanılmaktadırlar. Şekil 5'te bir mikrodenetleyiciye ait blok diyagram verilmiştir.



Şekil 5 Mikrodenetleyicili bir sisteme ait blok diyagram

3.2.1 Mikrodenetleyiciler hakkında genel bilgiler

Mikroişlemci üreten çeşitli firmalar tarafından birbirinden farklı mikrodenetleyiciler de üretilmektedir. Bu denetleyiciler yapısal olarak aralarında küçük farklar olmasıyla birlikte genel olarak aynı işlemleri yapabilmektedirler. Bu firmalar arasında en fazla rağbet gören, Microchip (PIC serisi mikrodenetleyiciler) ve Intel (MCS-51 ailesi mikrodenetleyiciler) firmaları sayılabilir.

Bir uygulama için işimizi görebilecek bir mikrodenetleyicinin seçiminde bazı özellikleri önceden tespit etmek gerekir. Bu özellikler (Altınbaşak, 2000) şunlardır:

- Programlanabilir dijital paralel giriş/çıkış,
- Programlanabilir Anolog giriş/çıkış,
- Seri giriş/çıkış,
- Motor veya servo kontrol için pals sinyali çıkışı,
- Harici giriş vasıtası ile kesme,
- Timer vasıtası ile kesme,
- Harici bellek arabirimi,
- Harici bus arabirimi (PC ISA gibi),
- Dahili bellek tipi seçenekleri (ROM, PROM, EPROM, EEPROM),
- Dahili RAM seçeneği,
- Kayan nokta hesaplaması.

3.2.2 Mikrodenetleyicilerin avantajları

Mikrodenetleyicilerin mikroişlemcilere olan üstünlükleri oldukça fazladır. Mikroişlemcili bir sistem yapıldığında mikroişlemcinin yanı sıra hafızalar (RAM, ROM EPROM, EEPROM), I/O (giriş/çıkış) birimi ve buna benzer birçok sisteme ihtiyaç duyulmaktadır. Bu karışık sistemin hem tasarlanması ve yapımı zordur hem de maliyeti oldukça yüksektir. Mikrodenetleyicili bir sistemin çalıştırılabilmesi için yalnızca bir mikrodenetleyici ve osilatör devresi yeterli olmaktadır. Sistemde gerekli olan ön bellek ve I/O birimi mikrodenetleyiciler içinde bir yonga halindedir.

3.3 PIC Mikrodenetleyicileri

PIC ismi, "*Peripheral Interface Controller*" kelimelerinin baş harflerinin kısaltılmasından oluşmuştur. Bu ifadeler "*çevresel birimleri denetleyici*" manasına gelmektedir. PIC bu anlama uygun olarak, lamba, motor, röle, LCD display, ısı ve ışık sensörü gibi giriş/çıkış elemanlarının denetimini yapabilecek şekilde dizayn edilmiştir. İlk olarak 1994 yılında 16 bitlik ve 32 bitlik büyük işlemcilerin, giriş ve çıkışlarındaki yükü azaltmak ve denetlemek amacıyla çok hızlı ve ucuz bir çözüme ihtiyaç duyulduğu için geliştirilmiştir.

3.3.1 PIC mikrodenetleyicilerinin tercih edilmesinin sebepleri

a) Lojik uygulamalarının hızlı olması,
b) Fiyatının oldukça ucuz olması,
c) 8 bitlik mikrodenetleyiciler olması ve bellek ve veri için ayrı yerleşik bus'ların kullanılması,
d) Veri ve belleğe hızlı olarak erişimin sağlanması,
e) PIC'e göre diğer mikrodenetleyicilerde veri ve programı taşıyan bir tek bus bulunması, dolayısıyla PIC'in bu özelliği ile diğer mikrodenetleyicilerden iki kat daha hızlı olması,
f) Herhangi bir ek bellek veya giriş/çıkış elemanı gerektirmeden sadece 2 kondansatör ve bir direnç ile çalışabilmeleri,
g) Yüksek frekanslarda çalışabilme özelliği,
h) Standby durumunda çok düşük akım çekmesi,
i) İntterrupt kapasitesi ve 14 bit komut işleme hafızası,
j) Kod sıkıştırma özelliği ile aynı anda birçok işlem gerçekleştirebilmesi,
k) Yazılımının Microchip'ten veya internetten parasız olarak elde edilebilmesi,
l) Çok geniş bir kullanıcı kitlesinin bulunması,
m) Kolay ve ucuz olarak elde edilebilmesi,
n) Çok basit reset, clock sinyali ve güç devreleri gerektirmeleri.

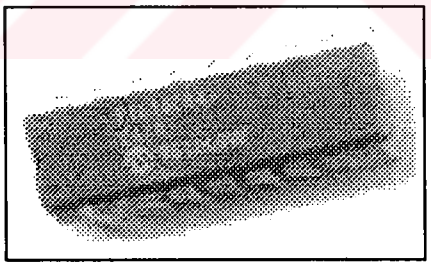
3.4 PIC16F84 Mikrodenetleyicisi

PIC mikrodenetleyicileri çeşitli özelliklerine göre PIC16C6X, 16C7X, 16C5X gibi gruplara ayrılmaktadırlar. Bu seriler, giriş çıkış sayısı, EEPROM (Flash) belleğe sahip olması, hafıza büyüklüğü, analog sayısal çeviriciye sahip olması, çalışma frekansı, vb. özellikleri ile farklılık göstermektedir. Bu tez çalışmasında PIC16F84 (PIC16F84A) mikrodenetleyicisi kullanıldığından bu mikrodenetleyiciye ait detaylı bilgiler verilecektir. Bazı küçük farklılıklar dışında anlatılanlar bütün PIC'ler için geçerli özellikler olmaktadır.

PIC'ler özellikle de PIC16F84 yüksek hızlı otomobillerden, motor kontrolü uygulamaları, düşük enerji sarfiyatlı uzaktan çalışan sensörler, elektronik kilitler, asansör kumandaları, güvenlik aygıtları ve akıllı kartlara kadar bir çok uygulamalarda kullanılmaktadırlar. EEPROM teknolojisi, uygulama programlarını son derece hızlı ve uygun hale getirmektedir. Düşük maliyet, düşük enerji sarfiyatı, yüksek performans, kullanım kolaylığı ve I/O esnekliği gibi özellikler bu mikrodenetleyicilerin kullanım alanlarını genişletmektedir. Zaman fonksiyonlu uygulamalar, seri haberleşme, PWM uygulamaları bunlardan bazılarıdır.

3.4.1 PIC16F84 mikrodenetleyicisinin iç yapısı

Şekil 6'da 18 bacaklı PIC16X8X mikrodenetleyicilerinin dış görünümü verilmiştir.

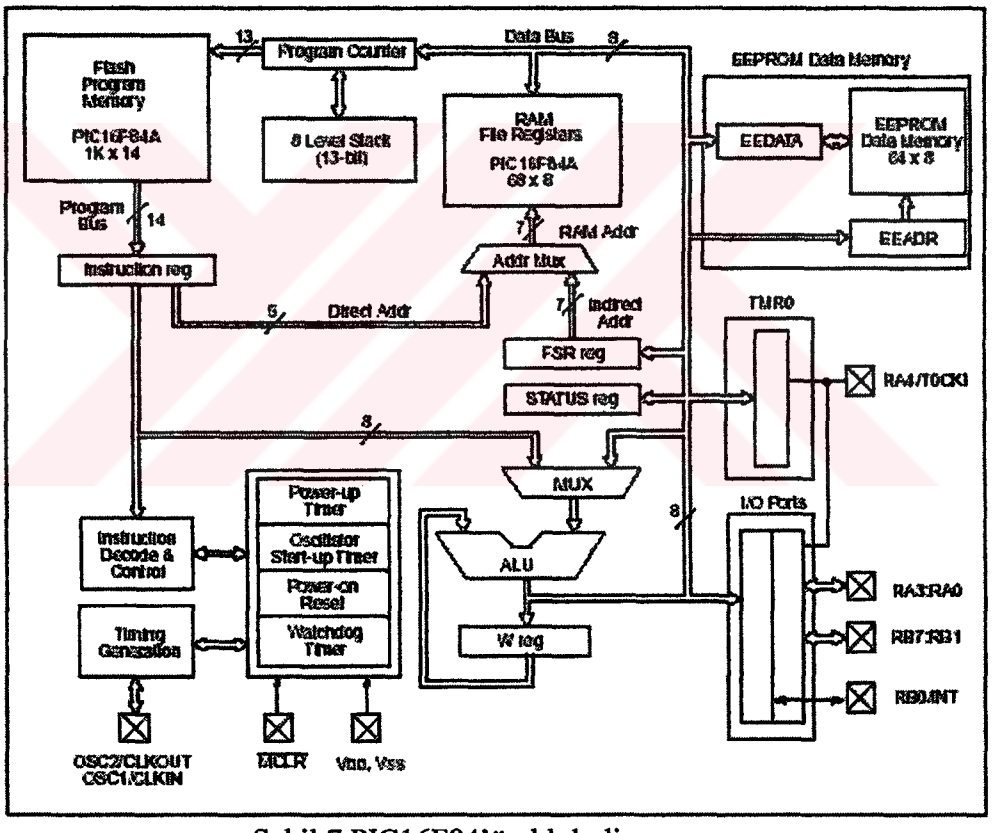


Şekil 6 PIC16X8X mikrodenetleyicilerinin dış görünümü

Çizelge 1'de PIC16F8X ailesinin özellikleri ve Şekil 7'de PIC16F84 mikrodenetleyicisine ait blok diyagram verilmiştir. PIC16F84 için maximum çalışma frekansı 10 Mhz ve kullanılabilir EEPROM (elektriksel olarak silinebilir ve yazılabilir) program hafızası 1 Kb' tır (1Kbx14 Bit). Ayrıca çalışan bir program içerisindeki verileri sürekli saklama özelliği olan 64 Kb (64x8 Byte)'lık ikinci bir EEPROM hafızaya sahiptir. Bu hafızaya yazılım aracılığıyla bilgi yüklenebilir ve okunabilir. Bu özellik istenmeyen bir durumda enerji kesilmesi gibi durumlarda istenilen verilerin saklanmasını sağlar. Bu belleğin yetersiz kaldığı durumlarda, ayrıca iki port ucu ile haberleşebilen seri EEPROM (24CXX vb.) hafızaları da PIC'e bağlanabilmektedir.

Çizelge 1 PIC16X8X ailesi özellikleri

		PIC16F83	PIC16CR83	PIC16F84	PIC16CR84
Saat palsi (Clock)	Maximum çalışma frekansı (MHZ)	10	10	10	10
Hafıza	EPROM program hafızası	512	—	1K	—
	EEPROM program hafızası	—	—	—	—
	ROM program hafızası	—	512	—	1K
	Veri hafızası (byte)	36	36	68	68
	Veri EEPROM'u (byte)	64	64	64	64
Aritbirim Özellikler	Zamanlama modülü	TMR0	TMR0	TMR0	TMR0
	Kesme kaynakları	4	4	4	4
	I/O pinleri	13	13	13	13
	Voltaj aralığı	2.0-6.0	2.0-6.0	2.0-6.0	2.0-6.0
	Paket	18-pin DIP, SOIC	18-pin DIP, SOIC	18-pin DIP, SOIC	18-pin DIP, SOIC



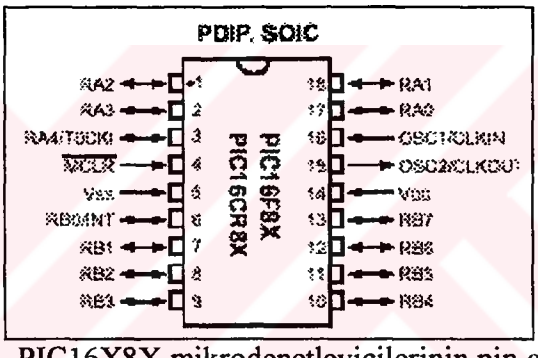
Şekil 7 PIC16F84'ün blok diyagramı

Mikrodenetleyicinin dış dünya ile haberleşmesini sağlayan I/O portlarıdır. Veri yolu 8 bitliktir. Programın işlemesi sırasında o anki program adresini tutan PC (program counter) ise 13 bitlik veri tutar. PIC16F84 içerisinde yığın bellek olarak kullanılan ve 8 adet 13 bitlik veri saklamaya yarayan bir STACK register mevcuttur. Stack register genellikle bir alt programdan veya kesme alt programlarından dönüş adreslerini saklamaya yarar. PIC16F84 14 bitlik komut işleme kapasitesine sahiptir. Şekil 7'de görülen diğer registerler ve özellikler ilerki bölümlerde detaylı olarak anlatılacaktır.

3.4.2 PIC16F84 mikrodenetleyicisinin pin görünüşü ve I/O (giriş/çıkış) portları

PIC16F84 mikrodenetleyicisinde 18 adet pin bulunmaktadır. Bu pinlerden 13 tanesi Giriş/Çıkış noktası olarak kullanılmaktadır. Bu I/O noktalarından 5 tanesi A portunu (RA0–RA4), 8 tanesi de B portunu (RB0–RB7) oluşturur. 13 I/O noktasından her biri giriş yada çıkış olarak kullanılabilir. Portların giriş/çıkış yönlendirilmesi PIC içerisindeki TRISA ve TRISB registerleri aracılığıyla yapılmaktadır. (Katzen, 2001)

Bu portlardan girilen dijital sinyaller vasıtasıyla PIC içerisinde çalışan programa veri girişi yapılır. Program verileri değerlendirerek, portları kullanmak suretiyle dış ortama dijital sinyaller gönderir. Dış ortama gönderilen bu sinyallerin kullanılacağı yerde, yeterli olmadığı durumlarda yükselteç devreleri kullanılarak (Transistör, Röle, vb.), kumanda edilecek cihaza uygun akım ve gerilim verilir.



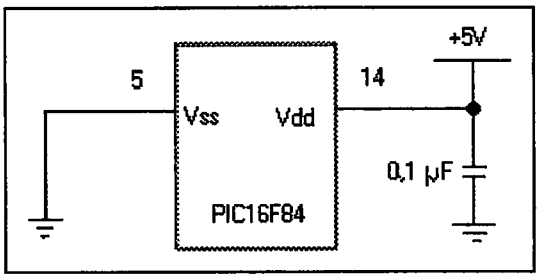
Şekil 8. PIC16X8X mikrodenetleyicilerinin pin görünüşü

I/O portlarından geçen maksimum "sink akımı" 25 mA ve "source akımı" 20 mA'dir. Bu değerlerden anlaşılacağı gibi çıkış olarak alınan bir sinyal LED'leri doğrudan sürebilir. PIC16F84'ün çektiği akım besleme gerilimine, clock girişine uygulanan sinyalin frekansına ve I/O portlarındaki yüke bağlı olarak değişir. Örnek olarak, 4 MHz'lik clock frekansında çektiği akım yaklaşık olarak 2 mA kadardır. Bu akım uyuma modunda (sleep mode) yaklaşık olarak 40 μA'e kadar düşer. CMOS entegrelerinde olduğu gibi entegredeki giriş uçları mutlaka bir yere bağlanmalıdır. Bu nedenle kullanılmayan tüm girişler besleme geriliminin +5 V'luk ucuna bağlanmalıdır. (Altınbaşak, 2000)

3.4.3 PIC16F84' ün besleme uçları ve beslenmesi

PIC16F84' ün besleme gerilimi 5 ve 14 numaralı pinlerden uygulanmaktadır. 14 numaralı Vdd ucu +5 V'a ve 14 numaralı Vss ucu ise, toprağa bağlanır. PIC' e ilk defa enerji verildiği anda meydana gelebilecek gerilim dalgalanmaları nedeniyle, oluşabilecek istenmeyen arızaları önlemek amacıyla Vdd ile Vss arasına 0.1 μF'lık dekuplaj

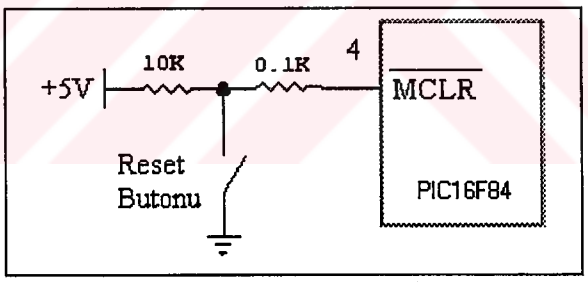
kondansatörü bağlamak gerekir. PIC'ler CMOS teknolojisi ile üretildiklerinden 2 ila 6 volt arasında çalışabilmektedirler. +5 V'luk bir gerilim ise ideal bir değer olmaktadır. PIC16F84 için bu bağlantılar Şekil 9'da gösterilmiştir.



Şekil 9 PIC16F84'ün beslenmesi

3.4.4 PIC16F84'ün reset uçları ve reset devresi

Kullanıcının programı kesip kasti olarak başlangıca döndürebilmesi amacıyla PIC'in 4 numaralı ucu (MCLR) kullanılmaktadır. MCLR ucuna 0 Volt uygulandığında programın çalışması başlangıç adresine döner. Programın ilk başlangıç adresinden itibaren tekrar çalışabilmesi için aynı uca +5 V gerilim uygulanmalıdır. Buna ilişkin reset devresi Şekil10'da verimiştir.



Şekil 10 Reset devresi

3.4.5 PIC16F84' ün Clock Uçları ve Osilatör Tipleri

PIC16CXX mikrodenetleyicilerinde 4 çeşit osilatör tipi bulunmaktadır. Kullanıcı bu 4 çeşitten birini seçerek iki konfigürasyon bitini (FOSC1 ve FOSC2) programlayabilir. Bu osilatör çeşitleri Çizelge 2'de verilmiştir. İstenilen osilatör tipinin seçimi ve OSC1 (16. uç) ile OSC2 (15. uç) olan clock uçlarına uygulanması aşağıda detaylı olarak anlatılmıştır.

Hazırlanacak olan PIC programlarında kullanılan osilatör tipi PIC programının çalışma hızını ve hassasiyetini etkileyeceğinden dolayı amaca uygun bir osilatör devresi kullanılmalıdır. Çizelge 2'de farklı osilatör çeşitleri ve özellikleri görülmektedir. Osilatör tipinin seçiminde dikkat edilmesi gereken bir başka nokta ise, seçilecek olan osilatörün

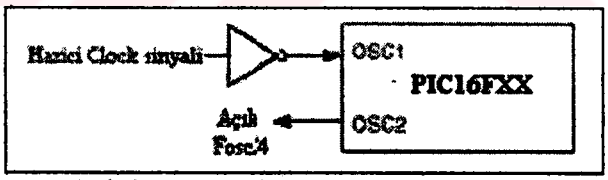
kullanılan PIC'in özelliğine uygun olarak seçilmesidir. Örnek verecek olursak, 10 MHz çalışma frekansına sahip bir PIC16F84 için 20 Mhz'lik bir osilatör devresi kullanılamaz. Fakat daha düşük bir frekans değeri ile çalışan bir osilatör devresi kullanılabilir. Kullanılacak olan PIC türünün de bu özelliklerine bakılarak seçilmesi gerekir. Bunun için microchip firmasının her PIC türü için hazırlamış olduğu data sheet'lere bakılmalıdır.

Çizelge 2 Osilatör çeşitleri

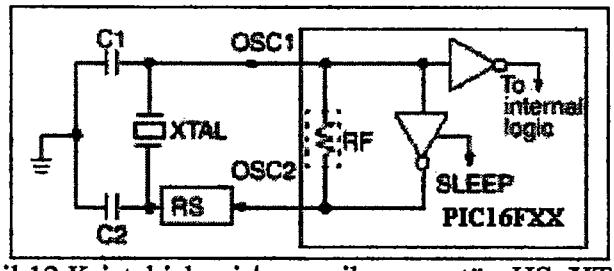
Osilatör Tipi	Tanımı	Özelliği	Frekans
LP	Kristal osilatör veya seramik rezonatör	asgari akım	40Khz
XT	Kristal osilatör veya seramik rezonatör	genel amaçlı	4Mhz
HS	Kristal osilatör veya seramik rezonatör	yüksek hız	20 Mhz
RC	Direnç/Kapasitör zaman sabitli	düşük maliyet	4Mhz

3.4.5.1 Kristal osilatör / seramik rezonatör

XT, LP ve HS modları, RC osilatörlere nazaran çok daha hassastırlar. Bu modlar, kristal osilatör veya seramik rezonatörlerin, OSC1/CLKIN ve OSC2/CLKOUT uçlarına bağlanmalarıyla kurulur. XT, LP ve HS modlarında OSC1/CLK1 dışardan sürülebilir. Şekil 12'de bu tür bir osilatör devresinin PIC16FXX serisi mikrodenetleyicilere bağlanması görülmektedir.Çizelge 3'te bazı frekanslar için örnek kristal ve kapasitör seçimleri verilmiştir (Microchip, 2000' de belirtildiği gibi).



Şekil 11 Harici clock girdi işlemleri



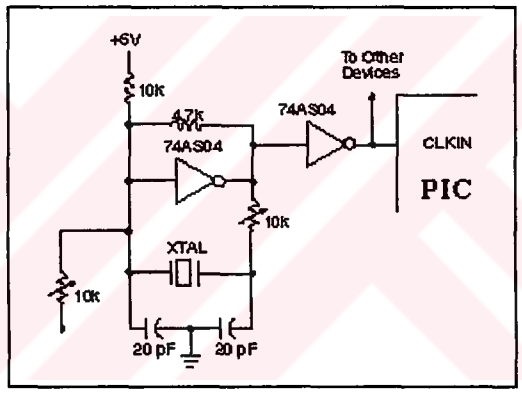
Şekil 12 Kristal işlemi / seramik rezonatör–HS, XT, LP

Çizelge 3 Kristal ve kapasitör değeri seçimleri

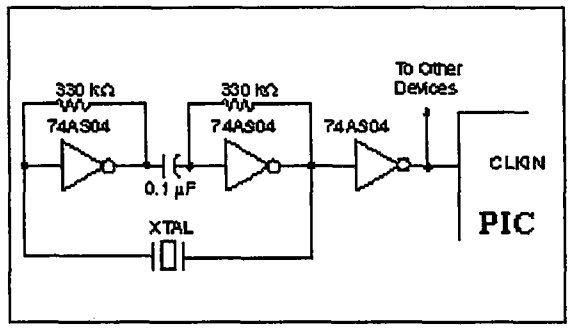
	Osilatör Tipi	Frekans	OSC1/C1	OSC2/C2
Kristal Osilatör için	LP	32 kHz	68 - 100 pF	68 - 100 pF
	XT	100 kHz	100 - 150 pF	100 - 150 pF
		2 MHz	15 - 33 pF	15 - 33 pF
		4 MHz	15 - 33 pF	15 - 33 pF
	HS	4 MHz	15 - 33 pF	15 - 33 pF
		10 MHz	15 - 33 pF	15 - 33 pF
		20 MHz	15 - 33 pF	15 - 33 pF
Seramik Rezonatör için	XT	455 kHz	47 - 100 pF	47 - 100 pF
		2.0 MHz	15 - 33 pF	15 - 33 pF
		4.0 MHz	15 - 33 pF	15 - 33 pF
	HS	8.0 MHz	15 - 33 pF	15 - 33 pF
		10.0 MHz	15 - 33 pF	15 - 33 pF

3.4.5.2 Dış kristal osilatör devresi

İki tip kristal osilatör devresi mevcuttur. Birisi seri rezonanslı ve diğeri de paralel rezonanslı osilatördür.



Şekil 13 Dış paralel rezonanslı kristal osilatör devresi



Şekil 14 Dış seri rezonanslı kristal osilatör devresi

Şekil 13'de paralel rezonanslı osilatör devresi görülmektedir. Devre, kristalin temel frekansını kullanmak için tasarlanmıştır. 74AS04 çevirici, paralel osilatörün gerektirdiği 180 dereceli faz kaymasını yürütmektedir. 4.7 K direnci kararlılık için negatif geri besleme

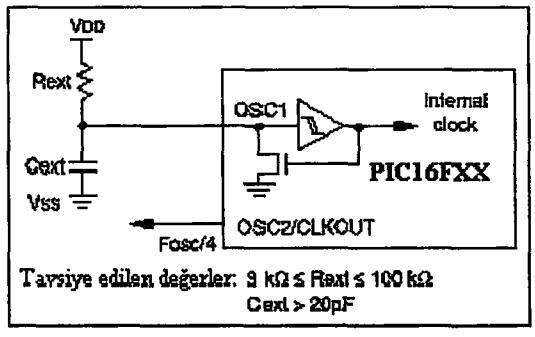
sağlamaktadır. 10 K'lık potansiyometre 74AS04'ü lineer bölgede çalıştırır. Bu devre osilatör tasarımı içinde kullanılabilir.

Şekil 14'te seri rezonanslı osilatör devresi görülmektedir. Bu devre kristalin esas frekansını kullanmak için tasarlanmıştır. Çevirici 180 derece faz kaymasını yürütmektedir. 330K'lık direnç, çeviricilerin lineer bölgelerinde çalışmaları için negatif geri besleme sağlamaktadır.

3.4.5.3 RC osilatörü

Zamanlamanın çok hassas olmasının gerekli olmadığı uygulamalarda, RC (Direnç-Kondansatör) ikilisi osilatör kaynağı olarak kullanılmaktadır. RC osilatör, maliyetin azaltılmasını sağlamaktadır. Kullanıcı dış R ve C elemanlarının toleransı nedeniyle meydana gelen değişiklikleri de dikkate almalıdır. Şekil 15'de RC kombinasyonunun PIC16F84'e nasıl bağlandığı görülmektedir. Rext direncinin değeri, 3 ila 100 Kohm arasında seçilmelidir.1 Mohm gibi yüksek direnç değerleri osilatörü gürültü ve nem gibi çevresel etkilere duyarlı hale getirir. 2,2 Kohm değerinin altında ise, osilatör işlemi kararsız hale gelebilir hatta tamamıyla durabilir. Osilatör RC modunda iken, OSC1 pini dış saat devresi ile sürülmemelidir. Aksi taktirde mikrodenetleyici kararsız çalışır. (Yalçın,1998) (Matic ve Ark., 2000)

RC osilatör kullanıldığında OSC2/CLKOUT ucunda OSC1/CLKIN ucuna uygulanan osilatör frekansının ¼'ü alınabilmektedir.

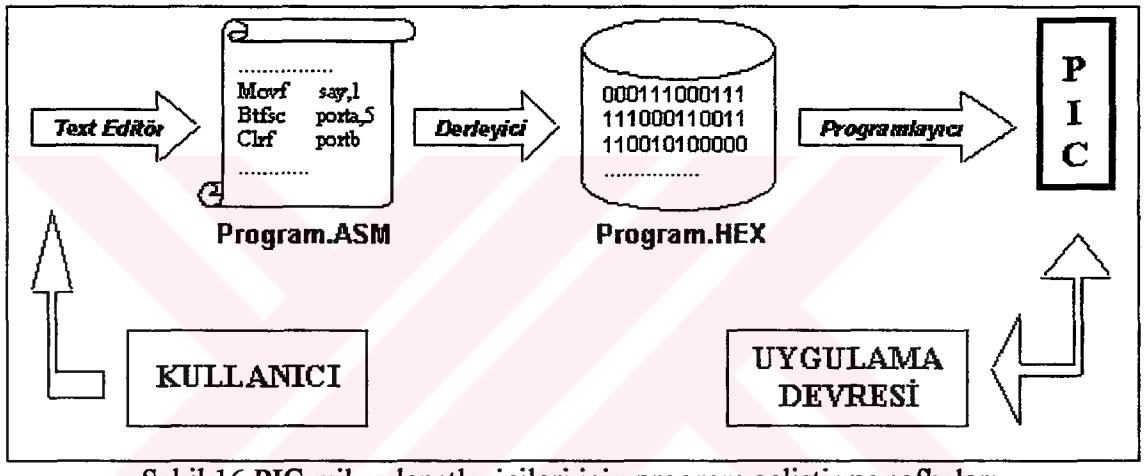


Şekil 15 RC osilatör modu

BÖLÜM IV

PIC16F84 İÇİN PROGRAM GELİŞTİRME SAFHALARI

Bu bölümde bir PIC mikrodenetleyicisi için program yazımından, mikrodenetleyiciye yüklenmesine kadar gecen aşamalar ve gerekli materyaller ele alınmıştır. Şekil 16'da bir PIC mikrodenetleyicisi için program geliştirme safhaları özet olarak verilmiştir.



Şekil 16 PIC mikrodenetleyicileri için program geliştirme safhaları

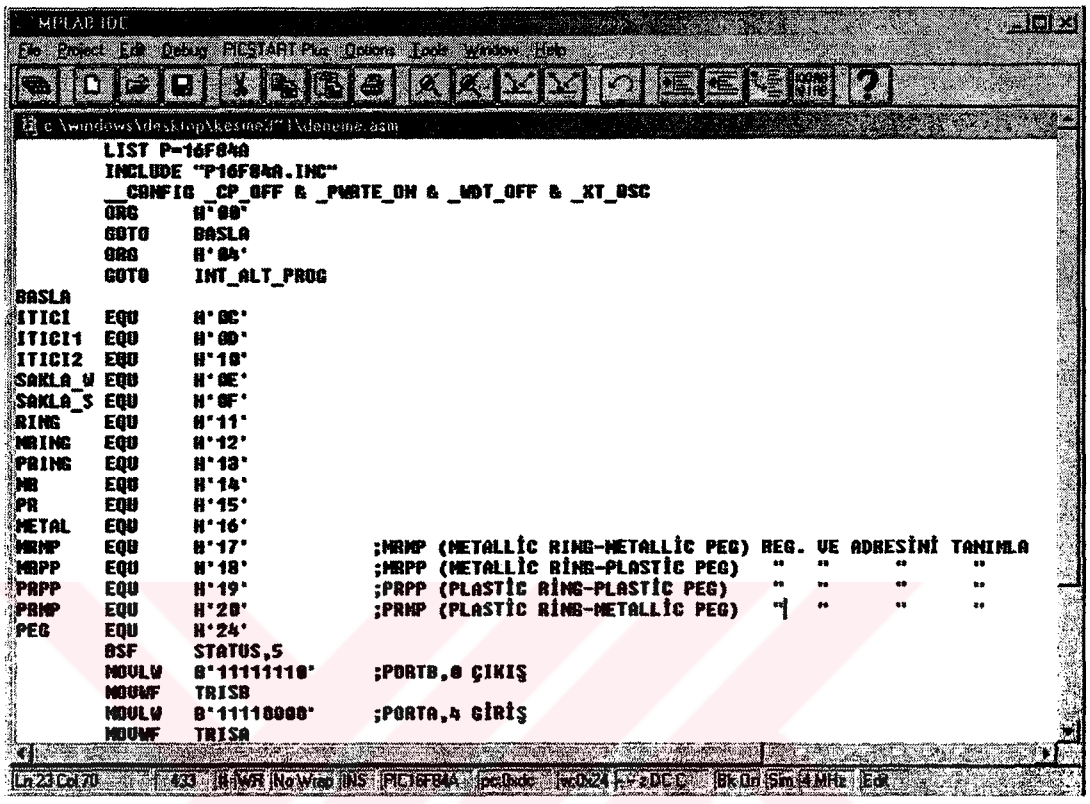
4.1 Text Editör

PIC assembly dili ile program yazmak için microchip firmasının ücretsiz olarak dağıttığı Şekil 17'de görülen MPLAB programı kullanılmaktadır. MPLAB programı içerisinde, yazılan PIC programının derlenmesi için gerekli olan MPASM programı da birlikte gelmektedir. Ayrıca bir text editöründe de program yazılabilmektedir. Ör: windows işletim sisteminde NOTPAD veya DOS işletim sisteminde EDIT gibi. Bu programlar kullanıldığında, yazılan PIC programı .ASM uzantılı olarak kaydedilmeli ve derlenmesi için MPASM kullanılmalıdır.

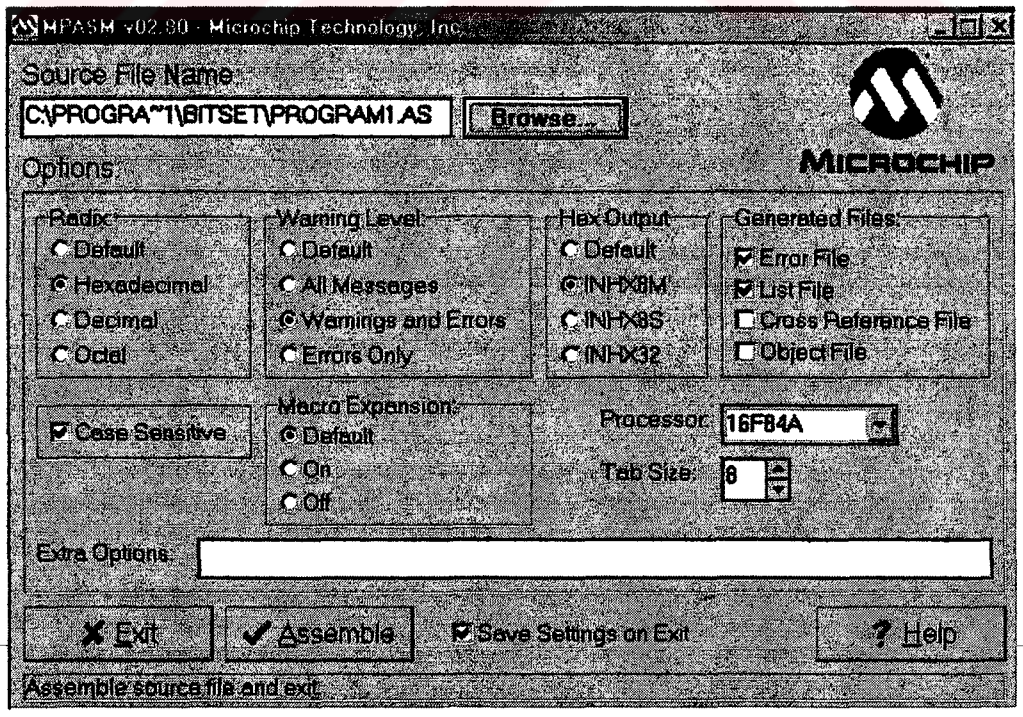
4.2 Derleyici

Derleyici PIC için yazılan ..ASM uzantılı programların PIC içerisine yazılabilecek formaca dönüştüren programdır. Microchip firmasının MPASM adını verdiği program bu işi yapmaktadır. MPASM programı, .ASM uzantılı PIC programını PIC' in anlayabileceği

.HEX uzantılı programa çevirmektedir.MPASM programı açılınca karşımıza Şekil 18'deki ekran gelecektir.



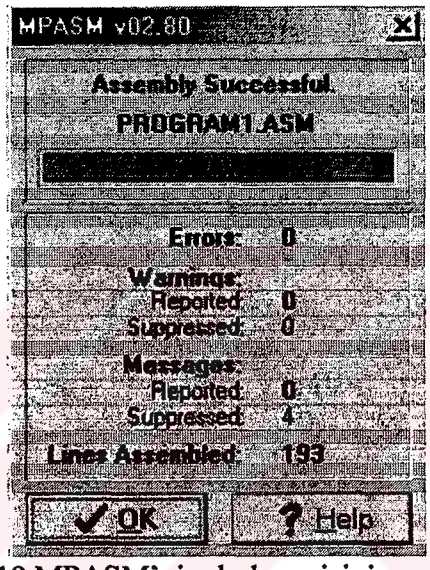
Şekil 17 MPLAB programının görünüşü



Şekil 18 MPASM derleyicisinin görünümü

Gerekli ayarlamalar yapıldıktan sonra., "Browse" tuşundan daha önceden hazırlanan .ASM uzantılı dosyayı seçip, "Assemble" tuşuna basmak yeterli olmaktadır. Bu işlem Şekil 19'daki pencerenin açılmasını sağlar. Açılan penceredeki "Assembly Successful" mesajı işlemin başarıyla tamamlandığını belirtir ve %100 yazan bölge yeşil renkli görünür. Aksi durumda kırmızı rengi alacak ve pencerede "Errors Found" hata mesajı görünecektir. Bu durumda programdaki hatalar (bugs) incelenmelidir.

Elde edilen .HEX uzantılı dosya .ASM uzantılı dosyanın bulunduğu klasöre otomatik olarak kaydedilecektir.



Şekil 19 MPASM'nin derleme işini yapması

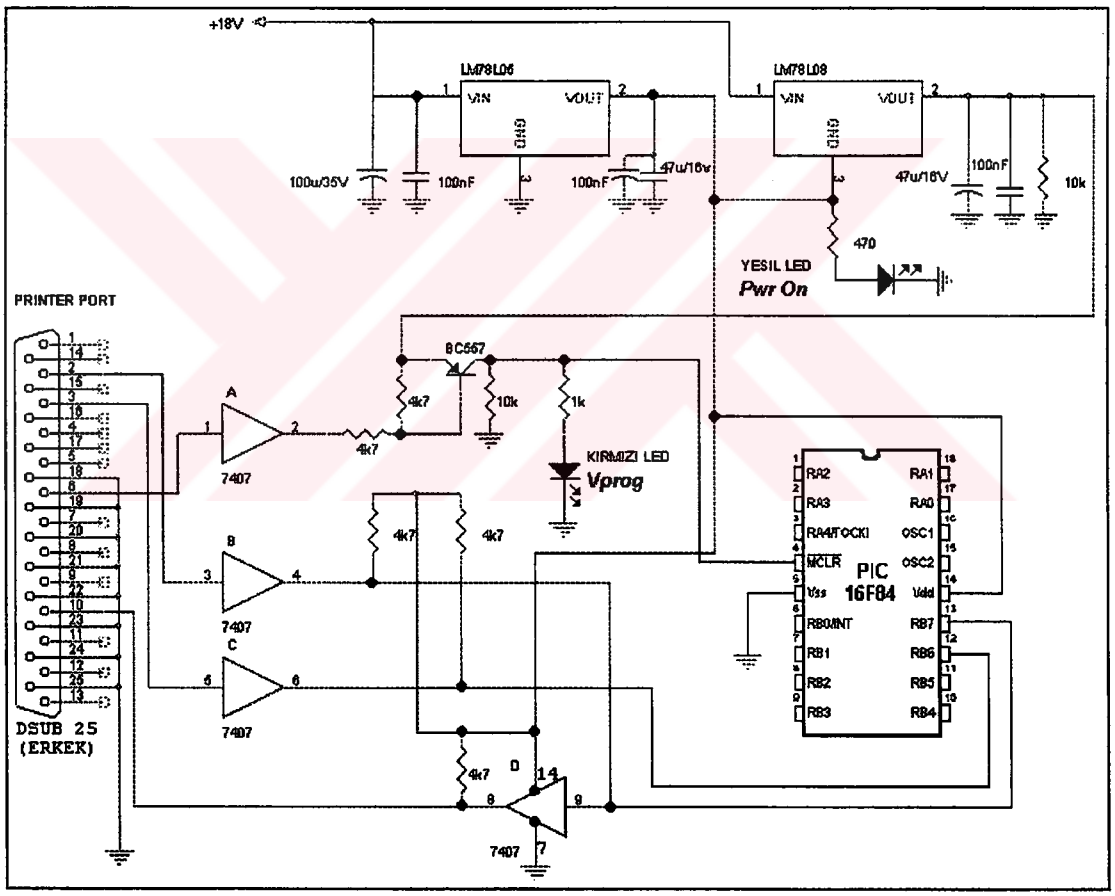
MPASM programı, MPLAB programı içerisinden direkt olarak çalıştırılabilmektedir. Bunun için, program .ASM uzantılı olarak kaydedildikten sonra Project → Build Node (veya ALT+F10 kısa yol tuşu) seçeneği kullanılır. Bu şekilde derlenen programlarda hata varsa, bu hatalar başka bir ekranda açılarak gösterilmektedir. Böylece hata bulmak kolaylaşmaktadır.

4.3 Programlayıcı Devre ve yazılımı

MPLAB ile hazırlanarak derlenen bir (.HEX uzantılı) programın bilgisayar ortamından PIC içine aktarılması için paralel veya seri port'tan haberleşen çeşitli donamımlar ve yazılımlar bulunmaktadır. Bir programlayıcı devre satın alınabileceği gibi internet ortamından çok rahatlıkla amaca uygun bir programlayıcı devre ve yazılımı bulunabilmektedir.Bu tez çalışmasında LPT (printer) portunu kullanan Şekil 20'deki

programlayıcı devre ve bu devre ile uyumlu olarak çalışabilen PROPIC programı kullanılmıştır.

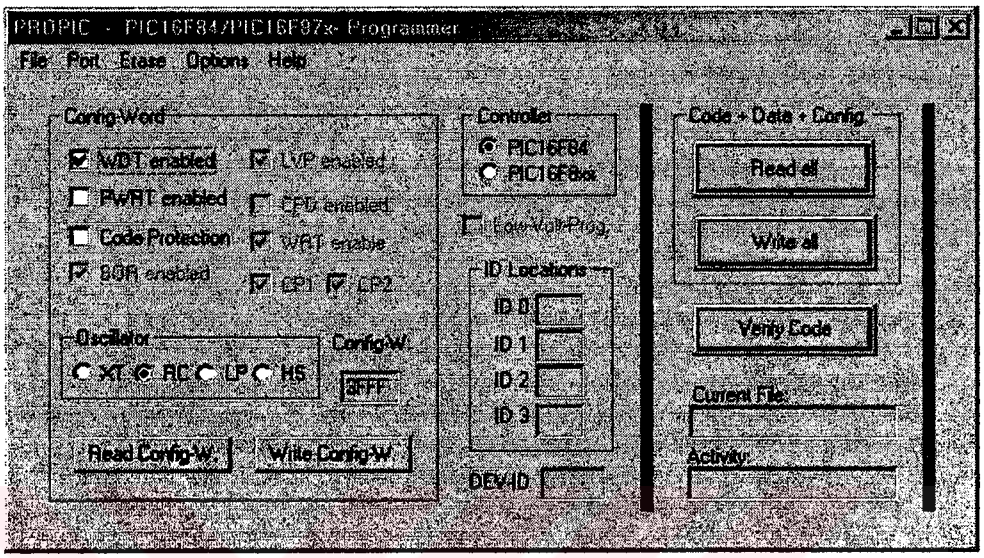
Şekil 20'de görülen devre kurulup bilgisayar bağlantısı yapıldıktan ve yaklaşık DC 18 volt beslendikten sonra PROPIC isimli program ile yükleme işlemi gerçekleştirilir. Programı kullanmadan önce bir defaya mahsus olmak üzere kullandığımız bilgisayarın LPT (paralel port) adresinin tanıtılması gerekmektedir. Bilindiği gibi bilgisayar paralel portları LPT1, LPT2 gibi isimlerle anılmakta ve adresleri de genellikle 378h veya 278h olmaktadır. Şekil 22'de görüldüğü gibi paralel port adresi işaretlenerek ayarlamalar yapılır. OK tuşuna basıldığında Şekil 23'teki mesaj çıkıyorsa, ya programlayıcı devre takılı değildir ya da devrede bir hata vardır.



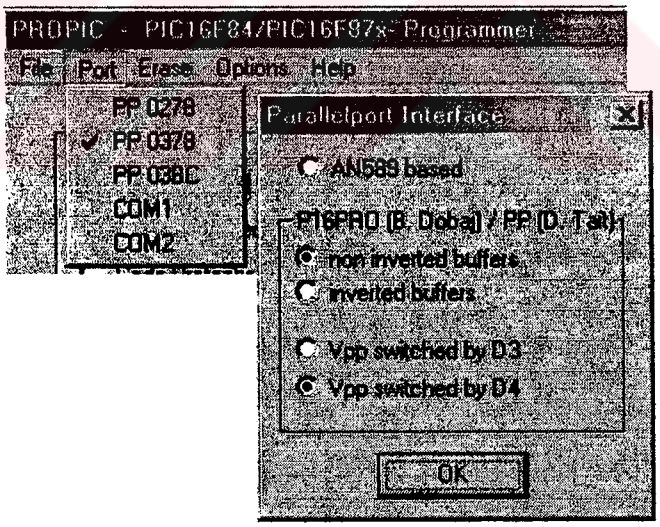
Şekil 20 Programlama devresi

Yeşil LED devreye 18 volt bağlandığında yanar, kırmızı LED ise programlama esnasında yanar ve programlama bitiminde söner. Önce File menüsünden .HEX uzantılı dosya seçilerek gerekli konfigürasyon ayarlamaları yapıldıktan sonra "Write all" tuşuna basılarak program yüklenir. Programın yüklenmesinde "Code protection" seçeneği işaretlenmiş ise PIC üzerine yalnızca bir defaya mahsus program yazılabilir. Eğer kod koruma konulması

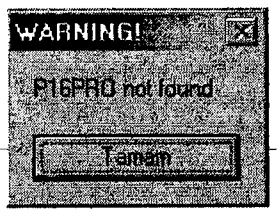
istenmiyorsa bu seçenek kesinlikle işaretlenmemelidir. Diğer konfigürasyon seçenekleri yazılan programa göre değişmektedir. Eğer Assembly programı yazarken konfigürasyon seçenekleri belirtilmiş ise PROPIC programı, bu ayarları kendisi otomatik olarak bulmaktadır.



Şekil 21 PROPIC programının görünüşü



Şekil 22 Programlayıcı port ayarının yapılması



Şekil 23 PROPIC programı hata mesajı

Programlayıcının kullanılması esnasında bilgisayar, PIC ve devrenin sağlığı açısından bazı noktalara dikkat etmek gerekmektedir. Bu noktalar şöyle özetlenebilir:

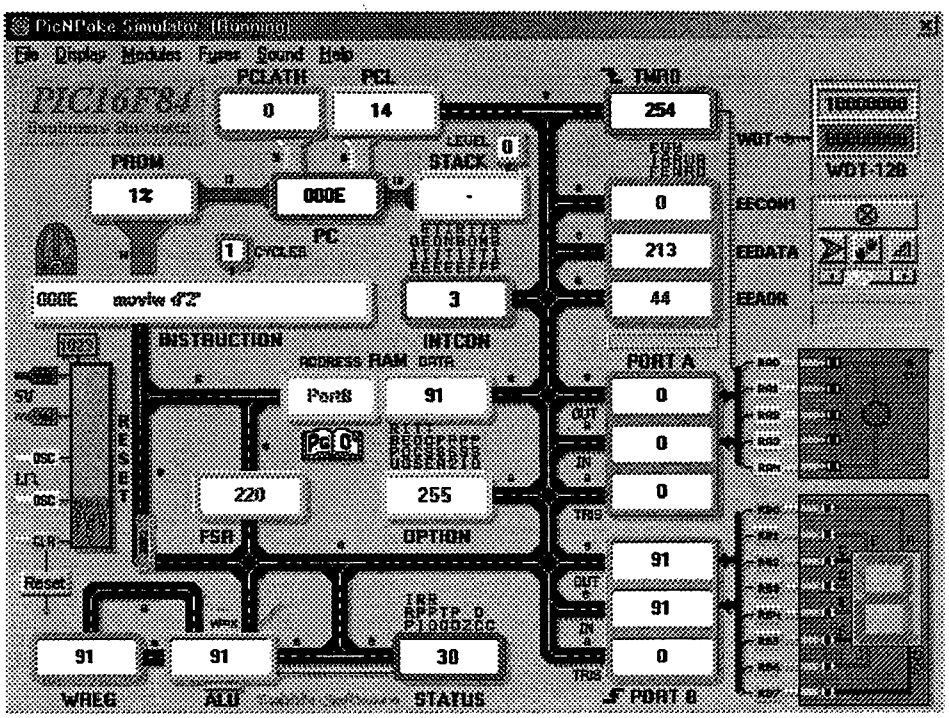
- PIC'in devreye doğru takıldığından emin olmak,
- Kırmızı LED yanarken PIC'i söküp takmaya çalışmamak,
- Programlayıcı devrede voltaj varken ve PC açıkken LPT portu bağlantısını takıp çıkarmamak,
- LPT bağlantısı takılı ve devrede 18 V gerilim bağlı ve kırmızı LED yanmazken PIC sökülüp takılabilir, ancak bu durumda PIC bacaklarını ve devre elemanlarını kısa devre etmemek,
- PC açıldığında programlayıcı devrede enerji varsa kırmızı LED, LPT portu initialize edilmediğinden yanık kalır bu durumda programlayıcı devre üzerinde PIC bulunuyorsa zarar görebilir. Bu durumda program bir kere çalıştırılıp kapatıldığında kırmızı LED söner. PIC'in zarar görmemesi için ya devre üzerinde PIC bırakılmamalıdır veya bilgisayarın her açılışında devreye enerji verilmeden program bir defa çalıştırılıp kapatılmalıdır.

Bu programlayıcı donanımı ve yazılımı aynı zamanda PIC16F8XX serisi PIC'leri de programlayabilmektedir. Devrede yalnızca PIC16F84 için ayak bağlantıları verilmiştir. PIC16F8XX serisi PIC'leri programlayabilmek için, ilgili PIC için VDD, MCLR, RB6, RB7, VSS isimli programlama bacakları uygun şekilde bağlanmalıdır.

4.4 Simülatör Programları

PIC programlarının yazılmasından sonra doğru çalışıp çalışmadığını test edebilmek için geliştirilen çok sayıda simülatör programı mevcuttur. Bunlardan ilki microchip firmasının geliştirmiş olduğu MPSIM programıdır. MPLAB programı ile birlikte ücretsiz olarak gelmektedir. Bu programda hazırlanan bir .ASM programının çalışması esnasında, register içerikleri, port durumları vb. incelenebilmekte ve yapılan bir hata varsa bulunması kolaylaşmaktadır. Bunun dışında internetten demo sürümleri ücretsiz alınabilen PICNPOKE, PROTEUS, BITSET gibi, çeşitli firmaların geliştirmiş olduğu ve PIC programı ile beraber, PIC'e çeşitli devre elemanlarının (LED, LCD display vb.) bağlanmasına izin veren ve görsel simülasyon yapabilme imkanı tanıyan programlar geliştirilmiştir. Bazı programlar için internet adresleri kaynaklar bölümünde verilmiştir.

Şekil 24'te PICNPOKE simülatör programından örnek bir ekran görülmektedir. Bu programda, çalışan bir PIC programının hem register içerikleri izlenebilmekte hem de PIC bacaklarına çeşitli devre elemanları bağlanarak simülasyon yapılabilmektedir.



Şekil 24 PICNPOKE simülatör programından örnek bir ekran

BÖLÜM V

PIC ASSEMBLY İLE PROGRAM YAZIMI

Bu bölümde PIC ASSEMBLY dili komutları tanıtılmış, PIC assembly dili ile program hazırlamak için gerekli diğer bilgiler bazı program örnekleri verilmiştir.

5.1 Sayı Sistemlerinin Gösterilişi

PIC programlama esnasında, onlu (desimal), ikili (binary), ve onaltılı (hexadecimal) sayı sistemlerinden herhangi birisi veya aynı program içerisinde farklı program satırlarında farklı sayı sistemleri, yazım kurallarına uyularak kullanılabilmektedir. Bu sayı sistemlerinin gösterilmesine ait örnekler aşağıda verilmiştir.

Onlu (Decimal) sayılar: sayının önünde d veya D harfi bulunmalıdır. Ör: decimal 25 sayısı D'25' veya d'25' şeklinde gösterilir.

İkili (Binary) sayılar: sayının önünde b veya B harfi bulunmalıdır. Ör: binary 10010111 sayısı B'10010111' veya b'10010111' şeklinde gösterilir.

Onaltılı (Hexadecimal) sayılar: sayının önünde 0x, 0, h veya H bulunmalıdır. Ör: hexadecimal 4F sayısı 0x4F, 04F, h'4F' veya H'4F' şeklinde gösterilir.

ASCII Karakterler: karakter kesme işaretleri arasına yazılmalıdır. Ör : 'A', 'B'.

5.2 PIC Assembly Komutları

PIC16F84 mikrodenetleyicilerini programlayabilmek için 35 adet komut bulunmaktadır. Programlamada bu komutların yanı sıra mikrodenetleyici içerisinde bulunan ve komutların kullanımından etkilenen özel register (Special Function Register)'lerin özelliklerinin bilinmesi gerekmektedir. Komutların ve özel registerlerin işlevleri sırayla verilecektir. Komutlar açıklanırken bazı terimlerin ve kısaltmaların yeri geldiğinde daha detaylı olarak açıklanacak olmasına rağmen kabaca ne anlama geldiğinin bilinmesi faydalı olacaktır.

Bit : İkilik sayı sistemine göre 0 ve 1 değerlerinden her birisini ifade eder.

Byte : 8 bitin bir araya gelmesi ile oluşan data (veri) kümesini ifade eder.

File register : Kullanıcının program içerisinde kullanmak üzere, PIC mikrodenetleyicisinin RAM belleği içerisinde kendisinin tanımladığı 8 bitlik veri işleyip saklayabilen register lardan herhangi birisini ifade eder.

Komut: Programlarda kullanılan emir veya direktiflerin herhangi birisini ifade eder.

W → (Working register), geçici olarak veri saklamaya yarayan registerdir. Bazı PIC komutlarının kullanımında, bir file register içeriği veya sabit bir sayı ile ancak W register içerisine yüklenerek işlem yapılabilmektedir. Örneğin, PORTB içeriği ile H'03' sayısı toplanıp yine PORTB içerisine yazılmak isteniyor ise, önce H'03' sayısı W registere yazılır, daha sonra PORTB ile toplanır.

```
MOVLW    H ' 03 '        ; H ' 03 ' sayısı W register' e yüklenir.
ADDWF    PORTB , 1       ; PORTB + W → PORTB
```

d → Destination, bazı komutların işletilmesinin ardından, işlem sonucunun kaydedileceği yeri ifade eder. İki ihtimali vardır. d ∈ [0,1]

d = 0 ise işlem sonucu W register içine yazılır.

d = 1 ise işlem sonucu file register içine yazılır.

b → Herhangi bir register (W, F) veya bir portun herhangi bir bitini ifade eder.

k → Sabit bir sayı veya bir etiketi ifade eder.

F → Herhangi bir file registeri veya portu ifade eder. File register PIC içerisinde mevcut özel registerlerden (STATUS, INTCON vb.) herhangi birisi olabileceği gibi kullanıcının program içerisinde kendisinin tanımladığı bir register de olabilir.

C → Cycle, komut saykıl sayısı

S → Komut işlemesi esnasında STATUS registerin etkilenen flag'i (biti)

TOS → (Top Of Stack), PIC içerisinde bulunan yığın (stack) belleğin üst hafıza bölümünü ifade eder.

PC → (Program Counter) Program sayıcı; yazılan programın işletilmesi sırasında, işletilen komutun o anda nerede (hangi adreste) olduğunu gösterir.

PCL → PC içeriğinin ilk 8 bitini tutan registerdir. PCL Ram belleğin h'02' adresinde bulunan özel bir registerdir.

PCLATH → PC içeriği 8 bitten fazla ise üst bitleri buradan yüklenir. PCLATH RAM belleğin h'0A' adresinde bulunan bir özel registerdir. PCLATH içeriği PC içerisine

yazılabilir fakat, PC içeriği PCLATH içerisine asla yazılamaz. Bu registerin ilk 3 biti PC içeriğinin 9, 10 ve 11. bitlerini tutar, 4. ve 5. biti ise 2 KB tan fazla belleğe sahip PIC'lerde bank verilerini tutmaya yarar. Son 3 biti ise kullanılmaz.

Microchip firması PIC komutlarını 3 ana gurup altında toplamaktadır(Microchip, 2000). Bunlar :

1) Byte yönlendirmeli ve file register ile birlikte çalışan komutlar :

Bu komutlar 1 byte (8 bit) uzunluğundaki W veya herhangi bir file register içeriğinin tümünü etkileyebilecek özelliğe sahip komutlardır. Aşağıdaki örnek ring isimli register içeriğinin tümünü (1 byte) sıfırlamaktadır.

```
CLRF    RING            ; RING registerinin 8 biti de silinmektedir.
```

2) Bit yönlendirmeli ve file register ile birlikte çalışan komutlar :

Bu komutlar W veya herhangi bir file register içeriğinin bir bitini etkileyen veya test eden komutlardır. Aşağıdaki örnekte ring registerinin 7. biti sıfırlanmaktadır.

```
BCF     RING , 7        ; RING registerinin sadece 7. biti silinmektedir.
```

3) Sabit ve kontrol komutları :

Bu komutlar ya tek başına veya sabit bir sayı ile birlikte kullanılırlar. Sabit bazen bir etiket olabilir. Aşağıdaki örneklerimizden birincisinde W registerinin içeriği onaltılı 34 sayısı ile toplanmaktadır. İkinci örneğimizde ise program akışı TOPLA etiketine dallandırılmaktadır.

```
ADDLW   H'34'           ; W + H'34' → W     , Sonuç yine W registere yazılır.
GOTO    TOPLA           ; Program TOPLA  etiketinin bulunduğu satıra dallanır.
```

Bu bölümün bundan sonraki kısmında, komutların fonksiyonları bakımından benzer olanları guruplara ayrılarak anlatılmıştır.

5.2.1 Atama komutları

MOVF (Move F) **MOVLW** (Move Literal to W)
MOVWF (Move W to F)

Bu komutlar W register veya file register gibi hafıza elemanlarına bir sabit sayı yada bir başka register içerisindeki bilgiyi yüklemeye yarayan komutlardır. Bu komutlara ait açıklamalar Çizelge 4'te verilmiştir.

Çizelge 4 Atama komutları

YAZILIŞI	İŞLEVİ	C	S	AÇIKLAMA
MOVF F, d	F → d	1	Z	File Register içeriğini d (destination) içerisine yükler. ($0 \leq F \leq 127$)
MOVLW k	k → W	1	–	8 bitlik sabit bir sayıyı W register içerisine yükler. ($0 \leq k \leq 255$)
MOVWF F	W → F	1	–	W register içeriğini File register içerisine yükler ($0 \leq F \leq 127$)

Örnekler :

```
MOVF    ABC , 0        ;ABC isimli register içeriği W register' e yüklenir.
MOVLW   H ' 3D '       ;H '3D' sayısı W register'e yüklenir.
MOVWF   INTCON         ;W register içeriği INTCON register' e yüklenir.
```

5.2.2 Test etme ve dallanma komutları

BTFSC (Bit Test F, Skip if Clear) **BTFSS** (Bit Test F, Skip if Clear)
DECFSZ (Decrement f, Skip if Zero) **INCFSZ** (İncrement f, Skip if Zero)
GOTO (Go to Label) **CALL** (Call Subroutine)
RETURN (Return from subroutine) **RETLW** (Return with Literal in W)
RETFIE (Return From Interrupt)

Bu komutlar program içerisinde test etme ve karar verme gibi mantıksal işlemleri yerine getirirler. Ayarıca program akışını da değiştirebilen bu komutlara ait açıklamalar Çizelge 5'te verilmiştir

Örnekler :

```
BTFSC   PORTA, 2       ;PORTA nın 2. bitine bakar, 0 ise bir komut atlar, 1 ise
                       ;hiçbir işlem yapmaz .
BTFSS   STATUS, 0      ;STATUS registerinin 0. bitine bakar, 1 ise bir komut atlar,
                       ;0 ise hiçbir işlem yapmaz .
DECFSZ  SAY, 1         ;SAY registerinin içeriğini 1 azaltır sonucu yine aynı register
                       ;içerisine yazar, eğer sonuç 0 ise bir komut atlar,1 ise hiçbir işlem
                       ;yapmaz .
INCFSZ  SAY, 0         ;SAY registerinin içeriğini 1 artırır sonucu W register
                       ;içerisine yazar, eğer sonuç 0 ise bir komut atlar, 1 ise hiçbir işlem
                       ;yapmaz .
GOTO    KONTROL        ;KONTROL etiketinin bulunduğu satıra gider.
```

Çizelge 5 Test etme ve dallanma komutları

YAZILIŞI	İŞLEVİ	C	S	AÇIKLAMA
BTFSC F, b	b = 0 ise bir komut atla	1(2)	--	F registerinin "b" bitini test eder, 1 ise sıradaki komutu işletir, 0 ise bir komut atlar. **(b = 0 ise C = 2, b = 1 ise C = 1)** **(0 ≤ F ≤ 127 , 0 ≤ b ≤ 7)**
BTFSS F, b	b = 1 ise bir komut atla	1(2)	--	F registerinin "b" bitini test eder, 0 ise sıradaki komutu işletir, 1 ise bir komut atlar. **(b = 1 ise C = 2, b = 0 ise C = 1)** **(0 ≤ F ≤ 127 , 0 ≤ b ≤ 7)**
DECFSZ F, d	F - 1 = 0 ise bir komut atla	1(2)	--	F registerinin içeriğini bir(1) azalttıktan sonra, File register içeriğine bakar, eğer F içeriği 0 olmuş ise bir komut atlar, F içeriği 0 değilse sıradaki komut işletilir. Komuttan sonra işlem sonucu d (destination) ye yazılır. **(0 ≤ F ≤ 127)**
INCFSZ F, d	F + 1 = 0 ise bir komut atla	1(2)	--	F registerinin içeriğini bir(1) artırdıktan sonra, File register içeriğine bakar, eğer F içeriği 0 olmuş ise bir komut atlar, F içeriği 0 değilse sıradaki komut işletilir. Komuttan sonra işlem sonucu d (destination) ye yazılır. **(0 ≤ F ≤ 127)**
GOTO k	k → PC<10:0>, PCLATH<4:3> → PC<12:11>	2	--	Koşulsuz dallanma komutudur. Program akışını k adresine veya etiketine dallandırır. k içeriği PC içerisine yazılır ve program o adrese yönlendirilmiş olur. **(0 ≤ k ≤ 2047)**
CALL k	PC + 1 → TOS, k → PC<10:0>, PCLATH<4:3> → PC<12:11>	2	--	k adresinde bulunan bir alt programı çağırır. Önce dönüş adresi (PC+1) yığın (stack) registerine kaydedilir. 11 bitlik (k) adresi PC<10:0>'a yüklenir. PC' nin ilk 8 biti PCL' den üst bitleri PCLATH' den yüklenir. **(0 ≤ k ≤ 2047)**
RETURN	TOS → PC	2	--	Bu komut alt programların sonuna yazılır. Alt programın çalışmasından sonra ana programda kalınan adrese dönülmesini sağlar. Daha önce TOS içine yazılan etikete (adrese) dönülür.
RETLW k	k → W TOS → PC	2	--	Return komutu gibi ana programa dönülmesini sağlar. Aralarındaki fark ise bu komutun 8 bitlik k değerini W registere yükleyerek dönmesidir. **(0 ≤ k ≤ 255)**
RETFIE	TOS → PC 1 → GIE	2	--	Kesme alt programlarının sonuna yazılır, ana programa dönülmesini sağlar. Bu komut ana programa dönerken INTCON registerin 7. bitini (GIE) 1 yaparak tüm kesmelerin aktifleşmesini sağlar.

```
CALL    KONTROL   ;Program KONTROL etiketi ile başlayan alt programa gider.
                  ;Giderken o anda bulunulan program adresinin 1 fazlası
                  ;(PC+1 → TOS) stack registere yazılır.
```

```
RETURN          ;Alt programın işletilmesinin ardından bu komut sayesinde ana
                ;programda kalınan adrese geri dönülür. Dönüş adresi stack
                ;registerden alınır.
RETLW   H'CA'   ;W registerine H'CA' sayısını yükledikten sonra ana programa
                ;geri dönülmesini sağlar.
RETFIE          ;Kesme alt programı bitişinde ana programa dönülmesini sağlar.
```

5.2.3 Aritmetik işlem komutları

ADDLW (Add Literal and W) **ADDWF** (Add W with F)

SUBLW (Subtract W From Literal) **SUBWF** (Subtract W From File register)

Toplama, çıkarma gibi aritmetik işlemleri yerine getiren bu komutlara ait açıklamalar Çizelge 6'da verilmiştir.

Çizelge 6 Aritmetik işlem komutları

YAZILIŞI	İŞLEVİ	C	S	AÇIKLAMA
ADDLW k	W + k → W	1	C, Z, DC	8 bitlik sabit bir sayı ile W registerinin içeriği toplanır, sonuç W registerinin içerisine yazılır. ($0 \leq k \leq 255$)
ADDWF F, d	W + F → d	1	C, Z, DC	File Register ile W registerinin içeriğini toplar, sonuç d (destination) ye yazılır. ($0 \leq F \leq 127$)
SUBLW k	k - W → W	1	C, Z, DC	8 bitlik sabit bir sayıdan W registerinin içeriği çıkarılır, sonuç W registerinin içerisine yazılır. ($0 \leq k \leq 255$)
SUBWF F, d	F - W → d	1	C, Z, DC	File Register içeriğinden W registerinin içeriği çıkarılır, sonuç d (destination)'ye yazılır. ($0 \leq F \leq 127$)

Örnekler :

```
ADDLW   0x15    ; W registerinin içeriği işlemden önce 0x10 ise,
                ; komuttan sonra 0x10 + 0x15 = 0x25 olur.
ADDWF   FSR , 0 ; İşlemden önce W = 0x17 ve FSR = 0xC2 ise,
                ; komuttan sonra W ← 0x17 + 0xC2, FSR =0xC2 olur.
SUBLW   0x02    ; Komuttan önce W = 0x01 ise,
                ; komuttan sonra W = 0x02 - 0x01 = 0x01, C = 1, Z = 0
                ; Komuttan önce W = 0x02 ise,
                ; komuttan sonra W = 0x02 - 0x02 = 0x01, C = 1, Z = 1
                ; Komuttan önce W = 0x03 ise,
                ; Komuttan sonra W = 0x02 - 0x03 = 0xFF, C = 0, Z = 0
```

SUBWF REG , 1 ; Komuttan önce **W** = 0x02 ve **REG** = 0x3 ise,

; Sonra **REG** = 0x03 - 0x02 = 0x01, **W** = 0x02 **C** = 1, **Z** = 0

d = 1 olduğundan sonuç REG e yazılır

; Komuttan önce **W** = 0x02 ve **REG** = 0x02 ise,

; Sonra **REG** = 0x02 - 0x02 = 0x00, **W** = 0x02 **C** = 1, **Z** = 1

; Komuttan önce **W** = 0x02 ve **REG** = 0x1 ise,

; Sonra **REG** = 0x03 - 0x02 = 0xFF, **W** = 0x02 **C** = 0, **Z** =0

Örneklerde görüldüğü gibi SUBLW ve SUBWF komutu çalıştığında STATUS registerinin C (Carry) ve Z (Zero) flaglerin içeriği değişmektedir. Bu sonuçlardan Çizelge 7'de görülen genellemeyi çıkarabiliriz

Çizelge 7 Çıkarma işleminde C ve Z flaglerin durumu

Sayıların durumu	C Flag	Z Flag
(Büyük sayı – Küçük sayı) REG > W	1	0
(Eşit iki sayı birbirinden çıkarılırsa) REG = W	1	1
(Küçük sayı – Büyük sayı) REG < W	0	0

5.2.4 Lojik işlem komutları

ANDLW (And Literal with W) **ANDWF** (And W with F)

IORLW (Inclusive OR Literal with W) **IORWF** (Inclusive OR W with F)

XORLW (Exclusive OR Literal with W) **XORWF** (Exclusive OR W with F)

Bu komutlar dijital elektronikte kullanılan lojik kapıların çalışmasına benzer işlevlere sahiptirler. Bazı lojik kapıların doğruluk tablosu Çizelge 8'de verilmiştir.

Çizelge 8 Bazı lojik kapıların doğruluk tablosu

Girişler	AND	OR	XOR
($X_1 - X_2$)	Tüm girişler 1 ise çıkış 1 dir	Girişlerden herhangi birisi 1 ise çıkış 1 dir	Girişler farklı ise çıkış 1 dir
0 – 0	0	0	0
0 – 1	0	1	1
1 – 0	0	1	1
1 – 1	1	1	0

Çizelge 9 Lojik işlem komutları

YAZILIŞI	İŞLEVİ	C	S	AÇIKLAMA
ANDLW k	**W** AND **k → W**	1	Z	8 bitlik bir sayı ile W register içeriğine lojik VE işlemi uygulanır,sonuç W register içerisine yazılır. **(0 ≤ k ≤ 255)**
ANDWF F, d	**W** AND **F → d**	1	Z	File Register ile W register içeriğine lojik VE işlemi uygulanır, sonuç d(destination) ye yazılır. **(0 ≤ F ≤ 127)**
IORLW k	**W** OR **k → W**	1	Z	8 bitlik bir sayı ile W register içeriğine lojik OR işlemi uygulanır, sonuç W register içerisine yazılır. **(0 ≤ k ≤ 255)**
IORWF F, d	**W** OR **F → d**	1	Z	File Register ile W register içeriğine lojik OR işlemi uygulanır, sonuç d(destination) ye yazılır. **(0 ≤ F ≤ 127)**
XORLW k	**W** XOR **k → W**	1	Z	8 bitlik bir sayı ile W register içeriğine lojik XOR işlemi uygulanır, sonuç W register içerisine yazılır. **(0 ≤ k ≤ 255)**
XORWF F, d	**W** XOR **F → d**	1	Z	File Register ile W register içeriğine lojik XOR işlemi uygulanır, sonuç d (destination) ye yazılır. **(0 ≤ F ≤ 127)**

Çizelge 9'da görüldüğü gibi lojik işlem komutları 8 bitlik iki sayı arasına uygulanmaktadır. Bu durumda 8 bitlik iki sayının karşılıklı bitleri (birinci sayının 0. biti ile ikinci sayının 0. biti, 1. biti ile 1. biti ...) lojik işleme tabi tutulur ve 8 bitlik yeni bir sayı elde edilir.

Örnekler :

ANDLW 0x5F ; Komuttan önce W = 0xA3 ise
; komuttan sonra W = 0x03 olur.

(H 'A3' = B '10100011' ve H '5F' = B '01011111')

```
        1 0 1 0 0 0 1 1  <-\   Alt alta olan bitler AND işlemine tabi tutulur.
AND     0 1 0 1 1 1 1 1  <-/
        ---------------
        0 0 0 0 0 0 1 1  ---> H '03' -> W
```

ANDWF FSR , 1 ; Komuttan önce W = 0x17, FSR = 0xC2 ise
; komuttan sonra W = 0x17, FSR = 0x02 olur.

d = 1 olduğu için sonuç FSR register içine yazılır, W içeriği değişmez.

IORLW 0x35 ; Komuttan önce W = 0x9A ise
; Komuttan sonra W = 0xBF ve (Zero flag) Z = 1 olur.

IORWF RES , 0 ; Komuttan önce W = 0x91 ve RES = 0x13 ise
; Komuttan sonra W = 0x93, RES = 0x13 ve Z = 1 olur.

d = 0 olduğundan sonuç W içine yazılır, RES register içeriği değişmez.

```
        1 0 0 1 0 0 0 1  ← RES (H '91') .
OR      0 0 0 1 0 0 1 1  ← W   (H '13')
-------------------------┼
        1 0 0 1 0 0 1 1 ⟶  H '93' → W
```

```
XORLW   0xAF            ; Komuttan önce W = 0xB5 ise
                        ; Komuttan sonra W = 0x1A olur.

XORWF   RES , 1         ; Komuttan önce W = 0xB5 ve RES = 0xAF ise
          ↗             ; Komuttan sonra W = 0xB5, RES = 0x1A olur.
```

d = 1 olduğundan sonuç RES register içine yazılır, W içeriği değişmez.

```
        1 0 1 0 1 1 1 1  ← RES (H 'AF')
XOR     1 0 1 1 0 1 0 1  ← W   (H 'B5')
-------------------------┼
        0 0 0 1 1 0 1 0 ⟶  H '1A'→ RES
```

5.2.5 Register içeriğini değiştiren komutlar

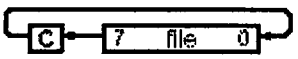
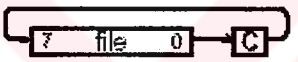
BCF (Bit Clear F)	**BSF** (Bit Set F)
CLRW (Clear W)	**CLRF** (Clear F)
DECF (Decrement F)	**INCF** (Increment F)
RLF (Rotate Left F)	**RRF** (Rotate Right F)
COMF (Complement F)	**SWAPF** (Swap nibbes in F)

Bu komutlar file registerlerinin içeriğini değiştirmeye yarar. Bir file register içeriğinin, tümünü değişrirebilen komutlar olduğu gibi sadece bir bitini değiştiren komutlar da bulunmaktadır. Bu komutlara ait açıklamalar Çizelge 10'da verilmiştir.

Örnekler :

```
BCF     STATUS , 5      ; Komuttan önce STATUS = B'00111111' ise
                        ; Komuttan sonra STATUS = B'00011111' olur.
BSF     STATUS , 5      ; Komuttan önce STATUS = B'00000000' ise
                        ; Komuttan sonra STATUS = B'00100000' olur.
CLRW                    ; Komuttan sonra W = 0x00 olur
CLRF    INTCON          ; Komuttan sonra INTCON = 0x00 olur
DECF    SAYI , 1        ; Komuttan önce SAYI = H'05' ise
                        ; Komuttan sonra SAYI = H'04' olur.
INCF    SAYI , 0        ; Komuttan önce SAYI = H'05'' ise
                        ; Komuttan sonra SAYI = H'05' ve W = H'06' olur.
```

Çizelge 10 Register içeriğini değiştiren komutlar

YAZILIŞI	İŞLEVİ	C	S	AÇIKLAMA
BCF F, b	0 → b	1	--	F registerinin b. bitini 0 yapar. **(0 ≤ b ≤ 7 , 0 ≤ F ≤ 127)**
BSF F, b	1 → b	1	--	F registerinin b. bitini 1 yapar. **(0 ≤ b ≤ 7 , 0 ≤ F ≤ 127)**
CLRW	H'00' → W 1 → Z	1	Z	W registerinin içeriğini siler (W registerinin bütün bitlerini 0 yapar).
CLRF F	H'00' → F 1 → Z	1	Z	F registerinin içeriğini siler (F registerinin bütün bitlerini 0 yapar). **(0 ≤ F ≤ 127)**
DECF F, d	F – 1 → d	1	Z	F registerin içeriğinden 1 eksiltir, sonuç d (destination) içerisine yazılır.**(0 ≤ F ≤ 127)**
INCF F, d	F +1 → d	1	Z	F registerinin içeriğini 1 artırır, sonuç d (destination) içerisine yazılır. **(0 ≤ F ≤ 127)**
RLF F, d		1	C	F registerinin 0...6 arası bitlerini 1 bit sola kaydırır, 7. bit önce C flag içerisine daha sonra 0. bitin yerine yazılır.**(0 ≤ F ≤ 127)**
RRF F, d		1	C	F registerinin 7...1 arası bitlerini 1 bit sağa kaydırır, 0. bit ise önce C flag içerisine daha sonra 7. bitin yerine yazılır. **(0 ≤ F ≤ 127)**
COMF F, d	$\overline{F}$ → d	1	Z	F registerinin içeriğinin tersi alınarak d (destination) içerisine yazılır.**(0 ≤ F ≤ 127)**
SWAPF F, d	F<3:0> → d<7:4> F<7:4> → d<3:0>	1	--	F registerinin ilk 4 biti ile son 4 biti yer değiştirir, sonuç d (destination) içerisine yazılır. **(0 ≤ F ≤ 127)**

```
RLF     FLAG , 1    ; Komuttan önce FLAG = B'00111111' ve C = 0 ise
                    ; Komuttan sonra FLAG = B'01111110' ve C = 1 olur.
RRF     FLAG , 1    ; Komuttan önce FLAG = B'00111111' ve C = 0 ise
                    ; Komuttan sonra FLAG = B'10011111' ve C = 0 olur.
COMF    FLAG , 1    ; Komuttan önce FLAG = B'00111111' ise
                    ; Komuttan sonra FLAG = B'11000000' olur.
SWAPF   REG , 0     ; Komuttan önce REG = B'00111111' ise
                    ; Komuttan sonra REG = B'11110011'olur.
```

5.2.6 Diğer komutlar

NOP (No Operation)

CLRWDT (Clear Watchdog Timer)

SLEEP (Go Into Standby Mode)

END (End)

Bu komutlar program içerisinde aynen yazıldığı gibi kullanılırlar. END komutunun bütün programlarda kullanılması zorunludur. Bu komutlara ait açıklamalar Çizelge 11'de verilmiştir.

Çizelge 11 Diğer komutlar

YAZILIŞI	İŞLEVİ	C	S	AÇIKLAMA
NOP	--	1	Z	Programın 1 komut saykılı beklemesini sağlar. Başka bir işlem yapmaz, hiçbir registeri etkilemez.
CLRWDT	H'00' → WDT 0→WDTprescaler 1 → TO 1 → PD	1	TO PD	Watchdog Timer'ı ve Watchdog Timer'ın prescaler değerini reset eder (sıfırlar). Status registerinin 3. ve 4. bitlerini 1 yapar.
SLEEP	H'00' → WDT 0→WDTprescaler 1 → TO 0 → PD	1	TO PD	Status registerinin enerji düşme biti PD yi siler, zaman çıkış biti TO bitini set eder, Watchdog Timer ve prescaler değerini siler. Böylece PIC uyuma moduna gecerek daha az güç harcar. WDT saymaya devam eder. Prescaler değeri dolduğunda TO = 0 olur, sleep modundan çıkılır. Ayrıca TOCKI pininden bir giriş sinyali veya MCLR ucundan reset ile de sleep modundan çıkılabilir.
END	--	--	--	Program komutlarının sonuna yazılır.

5.3 Status Register

PIC programlarının yazılmasında en fazla dikkate alınan özel registerlerden (SFR) birisi olan STATUS register PIC16F84 RAM belleğinin H'03' adresinde bulunmakta ve PIC RAM belleğinin her iki bankından da, bu register okunabilmektedir. Status register aritmetik ve lojik işlemlerin durumlarını gösteren verileri bitlerinde (flag) tutar. 5. ve 6. bitleri program içerisinde bank seçmek için kullanılır. 3. ve 4. bitleri PIC'in çalışma ve reset olma durumunu kontrol eder. İlk üç biti ise aritmetik ve lojik işlemlerin sonucunda etkilenerek işlemin sonucu hakkında fikir verir. Bütün bitlerin özellikleri ve etkilenme durumları Çizelge 12'de verilmiştir.

Eğer programda Bank1'e geçmek gerekirse PIC16F84 için BSF STATUS,5 komutu kullanılarak RP0 biti 1 (SET) yapılmalıdır. Bank0'a geçmek için BCF komutu kullanılır.

CLRF komutu STATUS registerinin içeriğinin tümünü beklendiği gibi sıfırlamaz. CLRF STATUS komutu bu registerin içeriğini '000uu1uu' şeklinde değiştirebilir. Burada

'u ' değişmeyen, önceki durumunu muhafaza eden bit manasındadır. Z flag (sıfır bayrağı) ise 1 olur.

Çizelge 12 STATUS register

STATUS REGISTER

7	6	5	4	3	2	1	0
IRP	**RP1**	***RP0***	***TO***	***PD***	**Z**	**DC**	**C**
(R/W -0)	(R/W -0)	(R/W - 0)	(R -1)	(R -1)	(R/W -x)	(R/W -x)	(R/W -x)

R = Okunabilir bit W = Yazılabilir bit -n = Bitin POR resetindeki değeri

Bit 7 **IRP :** Register bank seçme biti (Dolaylı adresleme için kullanılır)
Bu bit 16F8X serisi PIC'lerde kullanılmaz, bu PIC'lerde IRP = 0 kalmalıdır.
0 : Bank 0,1 (00h – FFh)
1 : Bank 2,3 (100h – 1FFh)

Bit 6-5 **RP1:RP0 :** Register bank seçme biti (Direkt adresleme için kullanılır)
Her bir bank 128 byte tır. 16F8X için sadece RP0 kullanılır. RP1=0 kalmalıdır

RP1	RP0	Bank	Adres Aralığı
0	0	0	(00h – 7Fh)
0	1	1	(80h – 1FFh)
1	0	2	(100h – 17Fh)
1	1	3	(180h – 1FFh)

Bit 4 **TO :** Zaman aşımı biti
1 : Pic enerjilendikten, CLRWDT veya SLEEP komutundan sonra
0 : WDT zamanlayıcısının süre aşımı sonunda (sayımı her bitirdiğinde)

Bit 3 **PD :** Enerji düşme biti
1 : Pic enerjilendikten veya CLRWDT komutundan sonra
0 : SLEEP komutundan sonra

Bit 2 **Z :** Sıfır biti
1 : Aritmetik yada lojik işlem sonucu " 0 " ise
0 : Aritmetik yada lojik işlem sonucu " 0 " değil ise

Bit 1 **DC :** Sayı taşma / ödünç alma biti (ADDWF ve ADDLW komutları için)
1 : 8 bitlik işlem sonucu 3. bitten 4. bite taşma varsa
0 : 8 bitlik işlem sonucu 3. bitten 4. bite taşma yoksa

Bit 0 **C :** Taşma / ödünç alma biti (ADDWF ve ADDLW komutları için)
1 : 8 bitlik işlem sonucu 7. bitte taşma olursa
0 : 8 bitlik işlem sonucu 7. bitte taşma olmazsa

5.4 PIC Assemly Dili ile Program Yazım Kuralları

Bir PIC mikrodenetleyicisi için assembly programı belirli bir formatta ve bazı kurallara uyularak yazılmaktadır. Program iki bölüme ayrılabilir. İlki PIC türünün tanıtıldığı ve program içerisinde kullanılacak olan file registerlerin atandığı bölüm. İkincisi ise asıl programın yazıldığı kısımdır. Başlangıçta (program hangi PIC için yazılıyor ise) PIC türünün tanıtılması zorunludur. Ardından kullanılacak olan file registerler EQU komutu kullanılarak PIC mikrodenetleyicisinin RAM belleğinde bir adres tayin edilerek tanımlanır. Bu adresler her PIC için RAM büyüklüğüne bağlı olarak farklı sayıda olabilir. PIC16F84 için 0Ch adresinden 4Fh adresine kadar herhangi bir adrese bir file register atanabilir. Özel file registerlerin (SFR, Special Function Register) adresleri sabittir ve mutlaka PIC'in

RAM haritasında belirtilen adrese atanmalıdır. PIC16F84 için bu adresler Şekil 25'te gösterilmiştir. Özel file registerler için microchip firması hazır tanıtma dosyaları hazırlamıştır. Bu dosyalar "include" dosyaları olarak MPLAB programı ile birlikte gelmektedir. .INC uzantılı bu doyalar istenilirse program başlangıcında eklenerek SFR' lerin tek tek tanıtılması zorluğu ortadan kalkar. (Katzen, 2001)

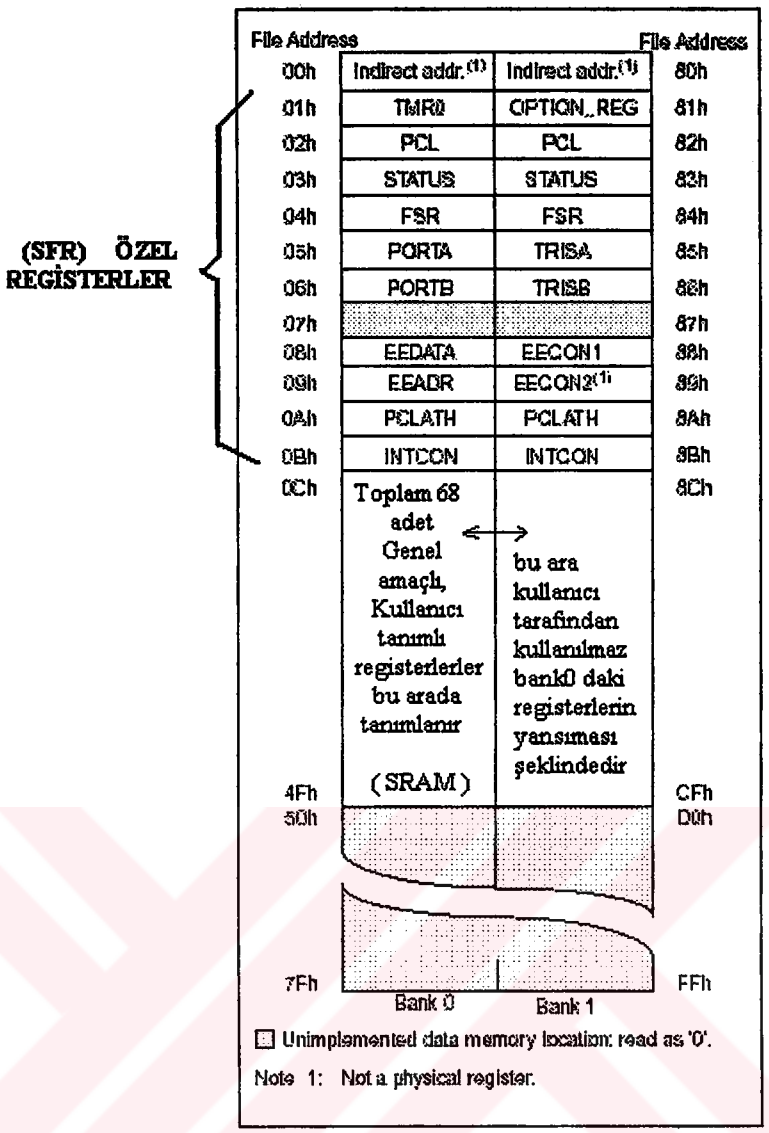
Tanıtma bölümünden sonra, kullanılacak olan asıl program yazılır. Program yazımında üç sütün kullanılmaktadır. Bu sütunlar TAB veya SPACE BAR tuşları ile ayrılabilmektedir. Etiketler kullanılacaksa mutlaka ilk sütuna yazılmalıdır. Komut ikinci sütuna ve komuttan sonra kullanılan komutun işlevi (hexadecimal bir adres, bir sabit veya dallanılacak olan bir etiket ismi) üçüncü sütuna yazılmalıdır. Program sonunda mutlaka END komutu kullanılmalıdır. Bu anlatılanlar aşağıda özet olarak verilmeye çalışılmıştır.

```
; Başlık bölümü, Bu alana istenilirse program ismi, tarihi, yazan vb.
; bilgiler (;) noktalı virgül işaretinden sonra yazılabilir.
;İlk sütun      ikinci sütun    üçüncü sütun
;---------      ------------    ---------------
               LIST            P=16F84         ;PIC türü tanıtılır
               INCLUDE         "P16F84.INC"    ;inc dosyası eklenir.

PORTA          EQU             H'05'           ;File Registerler tanıtılır
XXXX           EQU             H'15'           ;  "     "     "
               .....                           ;
ETIKET                                         ;Etiket ilk sütuna yazılır.
               KOMUT           ISLEVI          ;Komutlar 2. sütuna yazılır
               MOVLW           H'05'           ;Komut işlevleri 3. sütuna yazılır.
               GOTO            ETIKET          ;  "
               .....           .....           ;Program devamı
               .....           .....           ;  "
               END                             ;Program sonu
```

Bir PIC programı yazmak için o PIC'e ait bellek organizasyonunun iyi bilinmesi gerekir. Çünkü bir komuta, bir register ile ilgili bir iş yaptırmak için register hangi bank'ta ise, o banka geçmek gerekmektedir. (Predko, 2000) Ayrıca tanıtılacak olan registerlerin hangi adresten itibaren hangi adrese kadar tanıtılabileceği de bilinmelidir. Aşağıda verilen Program örneklerinde bu ifadeler daha iyi anlaşılacaktır.

Komutların kullanımında SFR (özel register)'lerin ve kullanıcı tanımlı registerlerin isimleri ya da adreslerinin kullanılması arasında bir fark yoktur. Örnek olarak, BSF PORTB,5 ile BSF 0x06,5 yazımı aynı ifadeyi temsil eder.



Şekil 25 PIC16F84 bellek organizasyonu

5.4.1 Başlangıç örnekleri

***Örnek1** : İki girişli VE (AND) kapısının PIC assembly dili ile gerçekleştirilmesi;*

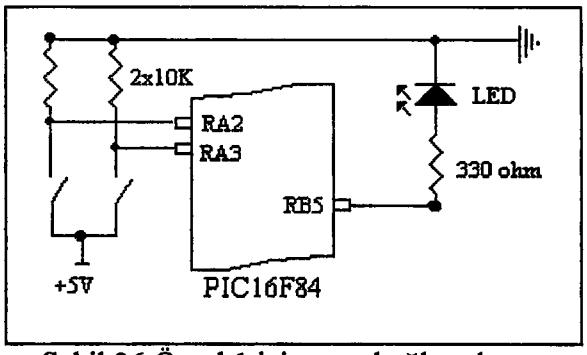
Bu örneğimizde girişler PORTA' nın 2. ve 3. bitleri çıkış ise PORTB' nin 5. biti olacaktır. Bu şekilde çalışan bir program aşağıdaki gibi yazılabilir.

Programın başında, kullanacağımız SFR'lerin PIC'in bellek organizasyonuna göre tanıtımı yapıldıktan sonra, PORTA'yı giriş ve PORTB'yi de çıkış olarak yönlendirildi. Bu işlem için TRISA ve TRISB registerlerine uygun sayıların yüklenmesi gerekmektedir. Bir portun herhangi bir bitini (ucu) giriş olarak yönlendirmek için o porta ait TRIS registerinin ilgili biti '1', çıkış olarak yönlendirmek için '0' yapılmalıdır. Örnek olarak PORTA'nın 3. biti giriş olarak yönlendirilmek istenirse TRISA'nın 3. biti '1' yapılmalıdır. TRIS registerleri Bank1'de olduğundan bu registerlere sayı yüklemek için Bank1'e geçmek gerekir.

```
        LIST    P=16F84         ;16F84' ü Tanıt
PORTA   EQU     H'05'           ;PORTA' yı tanımla
PORTB   EQU     H'06'           ;PORTB' yi tanımla
STATUS  EQU     H'03'           ;STATUS registeri tanımla
TRISA   EQU     H'85'           ;TRISA' yı tanımla
TRISB   EQU     H'86'           ;TRISB' yi tanımla
        CLRF    PORTB           ;PORTB' yi sıfırla
        BSF     STATUS,5        ;BANK1'e geç,Çünkü TRISB ve TRISA orada
        CLRF    TRISB           ;PORTB' yi çıkış yap
        MOVLW   H'FF'           ;W registerine H'FF' sayısını yükle
        MOVWF   TRISA           ;W -->TRISA (PORTA' yı giriş yap)
        BCF     STATUS,5        ;BANK0' a geç, Çünkü PORTA ve PORTB orada
BASLA                           ;Kontrol etmeye başla
        BTFSS   PORTA,2         ;PORTA 2. bit 1 mi?
        GOTO    CIKIS_0         ;Değil ise CIKIS_0 etiketine git
        BTFSS   PORTA,3         ;(PORTA 2. bit 1 ise)PORTA 3. bit 1 mi?
        GOTO    CIKIS_0         ;Değil ise CIKIS_0 etiketine git
CIKIS_1                         ;PORTA 2. ve 3. bit 1 ise
        BSF     PORTB,5         ;PORTB 5. biti 1 yap
        GOTO    BASLA           ;Tekrar kontrol etmeye başla
CIKIS_0                         ;PORTA 2. ve 3. bit 0 ise
        BCF     PORTB,5         ;PORTB 5. biti 0 yap
        GOTO    BASLA           ;Tekrar kontrol etmeye başla
        END                     ;SON
```

Bilindiği gibi AND kapılarının çıkış verebilmesi için girişlerinin hepsi 1 olmalıdır. Bazı kapılara ait doğruluk tablosu Çizelge 9'da verilmiştir. Bu programda dikkate alacağımız iki giriş mevcuttur. Bu girişlerin durumu, karşılaştırma komutları kullanılarak değerlendirilir ve ilgili bitin (RB5) çıkış vermesi sağlanır.

Bu programın çalışması Şekil 26'da verilen devre kurularak izlenebilir.Bu devrede yalnızca RA2 ve RA3 girişleri ve RB5 çıkışı için bağlantılar verilmiş olup PIC'in çalışması için gerekli olan BÖLÜM III'de verilen osilatör devresi, reset devresi ve besleme devresi eklenmelidir.Devrede 10K'lık dirençler RA2 ve RA3 girişinde enerji olmadığı durumlarda, bu girişlere sürekli 0 V gerilim uygulanmasını sağlar.



Şekil 26 Örnek1 için port bağlantıları

Örnek2 : Üç girişli VEYA (OR) kapısının PIC assembly dili ile gerçekleştirilmesi;

Bu örneğimizde girişler PORTA'nın 0., 1. ve 2. bitleri, çıkış ise yine PORTA'nın 3. biti olacaktır. Bu şekilde çalışan bir program aşağıdaki gibi yazılabilir.

```
        LIST   P=16F84              ;16F84' ü Tanıt
        INCLUDE "P16F84.INC"        ;P16F84.INC dosyasını ekle
        CLRF        PORTA           ;PORTB' yi sıfırla
        BSF         STATUS,5        ;BANK1'e geç,Çünkü TRISB ve TRISA orada
        MOVLW       B'11110111'     ;W registerine B'11110111' sayısını yükle
        MOVWF       TRISA           ;W →TRISA (PORTA 3. biti çıkış, diğer bitleri giriş yap)
        BCF         STATUS,5        ;BANK0' a geç, Çünkü PORTA orada
BASLA                               ;Kontrol etmeye başla
        BTFSC       PORTA,0         ;PORTA 0. bit 0 mi?
        GOTO        CIKIS_1         ;1 ise CIKIS_1 etiketine git
        BTFSC       PORTA,1         ;(PORTA 0. bit 0 ise)PORTA 1. bit 0 mı?
        GOTO        CIKIS_1         ;1 ise CIKIS_1 etiketine git
        BTFSC       PORTA,2         ;(PORTA 0.ve 1. bit 0 ise)PORTA 2. bit 0 mı?
        GOTO        CIKIS_1         ;1 ise CIKIS_1 etiketine git
        GOTO        CIKIS_0         ;Girişlerin hepsi 0 ise
CIKIS_1                             ;PORTA 0., 1. veya 2. bit 1 ise
        BSF         PORTA,3         ;PORTA 3. biti 1 yap
        GOTO        BASLA           ;Tekrar kontrol etmeye başla
CIKIS_0                             ;PORTA 0., 1. ve 2. bit 0 ise
        BCF         PORTA,3         ;PORTA 3. biti 0 yap
        GOTO        BASLA           ;Tekrar kontrol etmeye başla
        END                         ;SON
```

Programın çalışmasına ait açıklamalar yanında verilmiştir. Bu programda PORTA hem giriş hem de çıkış olarak kullanılmıştır. Portların bu özelliğiyle kullanılması baskı devrelerde de kolaylık sağlamaktadır. 16F84.INC dosyası eklendiği için SFR'lerin tanıtılmasına gerek kalmamıştır.

Bu programın izlenebilmesi için kullanılan port uçlarına göre Şekil 26'da verilen devrenin benzeri bir devre kurulabilir.

Örnek3 : RS - FLIP FLOP uygulamasının PIC assembly dili ile gerçekleştirilmesi;

Bu örneğimizde RS – FLIP FLOP olarak bilinen ve set girişi ile çıkışını set (1) yapan ve reset girişi ile bu çıkışı reset eden lojik entegrelerin çalışma prensibine uygun olarak çalışan bir program yazılacaktır. S ve R girişleri PORTA' nın 1. ve 2. bitleri, çıkış ise PORTB' nin 6. biti olacaktır.

Bu şekilde çalışan bir program aşağıdaki gibi yazılabilir.

```
        LIST   P=16F84              ;16F84' ü Tanıt
        INCLUDE "P16F84.INC"        ;P16F84.INC dosyasını ekle
        CLRF        PORTB           ;PORTB' yi sıfırla
        BSF         STATUS,5        ;BANK1'e geç,Çünkü TRISB ve TRISA orada
        CLRF        TRISB           ;PORTB' yi çıkış yap
        MOVLW       H'FF'           ;W registerine H'FF' sayısını yükle
        MOVWF       TRISA           ;W -->TRISA (PORTA' yı giriş yap)
        BCF         STATUS,5        ;BANK0' a geç, Çünkü PORTA ve PORTB orada
BASLA                               ;Kontrol etmeye başla
        BTFSC       PORTA,2         ;PORTA 2. bit 0 mı?
        GOTO        RESET           ;Değil ise RESET etiketine git
        BTFSS       PORTA,1         ;(PORTA 2. bit 0 ise)PORTA 1. bit 1 mi?
        GOTO        BASLA           ;Değil ise, Tekrar kontrol etmeye başla
SET                                 ;PORTA 2. ve 3. bit 1 ise
        BSF         PORTB,6         ;PORTB 6. biti 1 yap (Set)
        GOTO        BASLA           ;Tekrar kontrol etmeye başla
RESET                               ;PORTA 2. bit 0 ise
        BCF         PORTB,6         ;PORTB 6. biti 0 yap (Reset)
        GOTO        BASLA           ;Tekrar kontrol etmeye başla
        END                         ;SON
```

Bu programın izlenebilmesi için yine kullanılan port uçlarına göre, Şekil 26'da verilen devrenin benzeri bir devre kurulabilir.

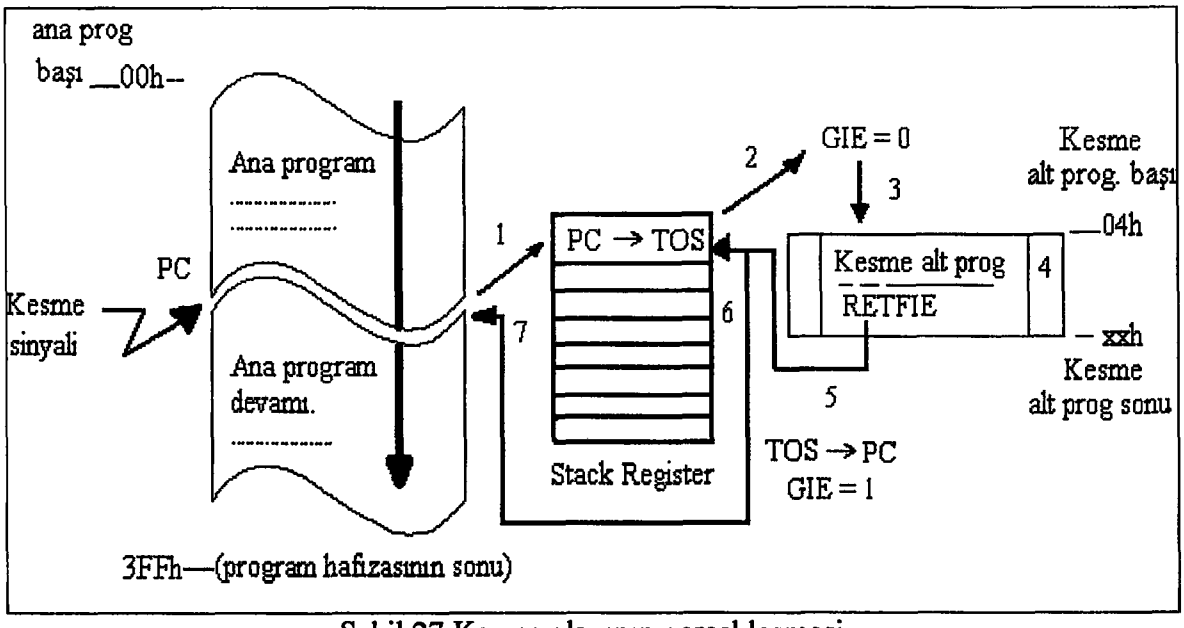
Bu üç örnekten de anlaşıldığı gibi tek bir PIC içerisinde bir çok lojik uygulama aynı anda birbirine bağımlı yada bağımsız bir şekilde gerçekleştirilebilir. Portların giriş yada çıkış olarak yönlendirilebilmesi, ve ayrıca zaman kontrollü uygulamalar, sayıcılar, kesme vb., aşağıda da anlatılacak olan birçok özellik sayesinde, PIC mikrodenetleyicisi ile yapılabilecek kontrol sistemlerinin gücü kendisini göstermektedir.

BÖLÜM VI

KESMELER (INTERRUPTS)

Ana program çalışmasını sürdürürken, mikrodenetleyici özelliğine bağlı olarak, gelen bir ya da daha fazla uyarı sinyali ile, sinyalin geldiği anda ana programın çalışmasını kesmesi ve bir alt programı çalıştırarak kaldığı yere geri dönmesi olayına "kesme" yada "interrupt" denilmektedir. Bu kesme sinyallerinin kaynağı PIC16F84 serisi için 4 çeşittir. (Matic ve Ark., 2000). Dolayısıyla bir programda 4 farklı şekilde kesme oluşturulabilmektedir. Çalıştırılacak olan alt programı interrupt veya kesme alt programı olarak isimlendireceğiz. Kesmelerin istenilen durumlarda program akışını, başka bir fonksiyonun gerçekleştirilebilmesi için değiştirebilme özelliğinden dolayı çok fonksiyonlu programların yazımı da kolaylaşmaktadır.

Şekil 27'de PIC mikrodenetleyicisinde bir kesme olayının gerçekleşme aşamaları özet olarak verilmiştir. Şekil 27'de görülen olaylar ve gerçekleşme sırası bütün kesme çeşitleri için ortaktır.



Şekil 27 Kesme olayının gerçekleşmesi

Bir kesme olayının gerçekleşme sırasını şu şekilde açıklanabilir;

1- Kesme sinyali geldiğinde (daha önce kurulan bir kesme varsa), ana program çalışmasını keser ve o anki PC (program sayıcı) içeriği STACK registerinin en üst hafıza bölümüne (TOS, Top Of Stack) yerleştirilir.
2- INTCON registerinin biti (GIE) sıfırlanarak kesme anında başka bir kesmenin meydana gelmesi önlenir.
3- 04h adresindeki komut çalıştırılır. Bu komut kesme alt programının ilk komutudur.
4- Kesme alt programı çalışmasını bitirir.
5- RETFIE komutu TOS içindeki PC içeriğini alarak yeniden PC' ye yükler.
6- Yeni kesmelerin gerçekleşebilmesi için GIE yeniden 1 yapılır.
7- Ana program kaldığı yerden devam eder.

6.1 PIC16F84 Mikrodenetleyicisinde Kesme Oluşturulması

Mikrodenetleyicide istenilen şekilde bir kesme oluşturabilmek için bazı özel file registerlerin (SFR) özelliklerinin bilinmesi gerekmektedir. Bütün kesme kaynaklarının geçerli ya da geçersiz olabilmesi INTCON registerinin 8 bitinin durumuna bağlıdır. TMR0 taşma kesmesini ve PORTB0 kesmesini kullanmak için OPTION, EEPROM'a yazma kesmesi için de EECON1 registerlerinin özellikleri bilinmelidir. Bu çalışmada TMR0 taşma kesmesi detaylı olarak anlatılacağı için INTCON ve OPTION registerlerin özellikleri açıklanacaktır. Kesme kaynakları ve oluşturdukları kesmeler Şekil 28'de özet olarak verilmiştir.

6.2 Intcon Register

INTCON register PIC16F84 mikrodenetleyicisinin RAM belleğinin 0Bh adresinde bulunmaktadır. Bütün kesme işlemlerinin kontrolü bu registerin bitlerine bağlı olarak yapılmaktadır. Program içerisinde en az bir kesme kullanılacak ise bu registerin 7. biti olan GIE = 1 yapılmalıdır. Bu işlem bütün kesmeleri aktif hale getirecektir. 6, 5, 4 ve 3. bitler program içerisinde kullanılması istenilen kesme çeşidinin aktif yapılmasını sağlar. 2, 1, ve 0. bitler ise herhangi bir kesme oluşmuş ise bunların durumunu bildirir. Örnek olarak, RB0/INT ucundan gelen bir sinyal ile kesme oluşması isteniyor ise OPTION registerinin 4. biti (INTE) 1 yapılmalıdır. Bu durumda bir kesme oluşmuş ise OPTION registerinin 1. biti (INTF) 1 olarak bu durumu bildirmektedir. (Microchip, 2000 ve Katzen, 2001)

INTCON registerinin bütün bitlerinin özellikleri ve kullanışına ilişkin açıklamalar Çizelge 13'te detaylı olarak verilmiştir.

Çizelge 13 INTCON register

INTCON REGISTER

7	6	5	4	3	2	1	0
GIE	**EEIE**	**TOIE**	**INTE**	**RBIE**	**TOIF**	**INTF**	**RBIF**
(R/W -0)	(R/W -0)	(R/W -0)	(R/W -0)	(R/W -0)	(R/W -0)	(R/W -0)	(R/W -x)

R = Okunabilir bit W = Yazılabilir bit -n = Bitin POR resetindeki değeri

Bit 7 **GIE :** Kesmelerin tümünü aktif yapma biti
0 : Hiçbir kesmeye müsaade edilmez
1 : Bütün kesmelere (interrupts) müsaade edilir.

Bit 6 **EEIE :** EEPROM belleğe yazma işlemini tamamlama kesmesini aktif yapma biti.
1 : EEPROM' a veri yazma işlemi tamamlama kesmesi aktif yapılır
0 : EEPROM' a veri yazma işlemi tamamlama kesmesi geçersiz yapılır

Bit 5 **TOIE :** TMR0 taşma kesmesini aktif yapma biti
1 : TMR0 taşma kesmesi aktif yapılır
0 : TMR0 taşma kesmesi geçersiz yapılır

Bit 4 **INTE :** RB0/INT kesmesini aktif yapma biti
1 : RB0/INT ucundan gelen kesmeler aktif yapılır.
0 : RB0/INT ucundan gelen kesmeler geçersiz yapılır.

Bit 3 **RBIE :** PORTB (RB7...RB4) uçlarındaki değişiklik kesmesini aktif yapma biti
1 : PORTB değişme kesmesi aktif yapılır
0 : PORTB değişme kesmesi geçersiz yapılır

Bit 2 **TOIF :** TMR0 taşma kesmesi bayrağı biti
0 : TMR0' da taşma olduğunda
1 : TMR0' da taşma olmadığında

Bit 1 **INTF :** RB0/INT kesmesi bayrağı biti
0 : RB0/INT kesmesi oluştuğu zaman
1 : RB0/INT kesmesi oluşmadığı zaman

Bit 0 **RBIF :** PORTB (RB7...RB4) uçlarında değişiklik kesmesi bayrağı biti
1 : PORTB (RB7...RB4) uçlarından en az birisinde değişiklik olursa
0 : PORTB (RB7...RB4) uçlarında hiçbir değişiklik yoksa

6.3 Option Register

OPTION register RAM belleğin 81h adresinde bulunur. Bu yüzden bu register ile işlem yapabilmek için bank1'e geçmek gerekir. Bu registerin 7. biti PIC devresinde B portuna yapılan pull-up'ların geçerli olması, 6. biti PORTB0 kesmesinin bu porttaki sinyalin hangi kenarında (yükselen veya düşen kenar) oluşacağı gibi durumları kontrol eder. 5. bit ise PIC içerisinde bulunan TMR0 sayıcısının kaynağını belirler. 4. bit programlarda kullanılacak olan sayıcı çeşidini (TMR0 veya WDT), son üç bit ise bu sayıcılar için prescaler (frekans bölme sayısını) atamak için kullanılır. (Microchip, 2000 ve Katzen, 2001)

OPTION registerinin bütün bitlerinin özellikleri ve kullanışına ilişkin açıklamalar Çizelge 14'te detaylı olarak verilmiştir.

Çizelge 14 OPTION register

OPTION REGISTER

7	6	5	4	3	2	1	0
RBPU	**INTEDG**	**TOCS**	**TOSE**	**PSA**	**PS2**	**PS1**	**PS0**
(R/W -1)	(R/W -1)	(R/W -1)	(R/W -1)	(R/W -1)	(R/W -1)	(R/W -1)	(R/W -1)

R = Okunabilir bit W = Yazılabilir bit -n = Bitin POR resetindeki değeri

Bit 7 **RBPU :** PORTB Pull-up geçerli yapma biti
0 : PORTB Pull-up' lar geçerli olur
1 : PORTB Pull-up' lar iptal edilir.

Bit 6 **INTEDG :** Interrupt kenar seçme biti
1 : RB0/INT ucundan gelen interupt'lar yükselen kenarda algılanır
0 : RB0/INT ucundan gelen interupt'lar düşen kenarda algılanır

Bit 5 **TOCS :** TMR0 saat kaynağı seçme biti
1 : RA4/TOCKI pinine (dışarıdan) verilen sinyal kaynak olarak seçilir
0 : Dahili komut saykılı kaynak olarak seçilir

Bit 4 **TOSE :** TMR0 kaynağı kenar seçme biti (TOCS = 1 ise)
1 : RA4/TOCKI ucundan gelen clock darbeleri düşen kenarda saati arttırır
0 : RA4/TOCKI ucundan gelen clock darbeleri yükselen kenarda saati arttırır

Bit 3 **PSA :** Prescaler atama biti
1 : Prescaler WDT' ye atanır
0 : Prescaler TMR0' a atanır

Bit 2-0 **PS2 : PS0 :** Prescaler oranı seçme bitleri. Bitlerin durumuna göre değerler aşağıda verilmiştir :

PS2	PS1	PS0	TMRO ORANI	WDT ORANI
0	0	0	1 : 2	1 : 1
0	0	1	1 : 4	1 : 2
0	1	0	1 : 8	1 : 4
0	1	1	1 : 16	1 : 8
1	0	0	1 : 32	1 : 16
1	0	1	1 : 64	1 : 32
1	1	0	1 : 128	1 : 64
1	1	1	1 : 256	1 : 128

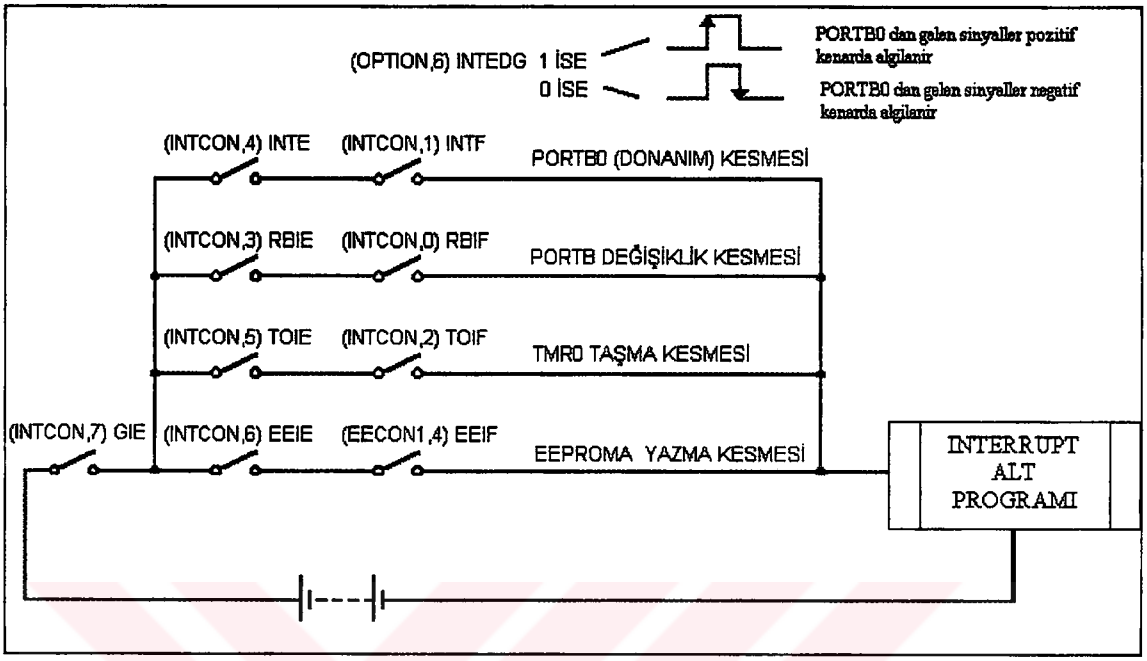
6.4 PIC16F84 Mikrodenetleyicisinde Kesme Çeşitleri

PIC16F84 mikrodenetleyicilerinde bulunan kesme çeşitlerini ve kesme durumlarını kontrol eden Özel registerlerin ilgili bitlerinin durumu Şekil 28'de verilmiştir.Bütün kesmelerin özel durumları aşağıda açıklanmıştır.

6.4.1 PORTB0 kesmesi

B portunun 0. bitinden (RB0) gelen sinyaller ile oluşturulan dış kaynaklı kesmedir. Bu kesmenin aktif olabilmesi için TRISB'nin 0. biti 0 yapılmalı yani RB0 ucu giriş olarak yönlendirilmeli, INTCON registerinin 4. biti (INTE) 1 yapılarak bu kesme aktif hale getirilmelidir. Ayrıca OPTION registerinin 6. biti INTEDG sayesinde bu kesmenin RB0 ucundaki sinyalin pozitif veya negatif kenarında meydana gelmesi sağlanabilir. RB0 kesmesi oluştuğunda INTCON registerinin 1. biti (INTF) 1 olur. INTE ve INTF bitleri oluşabilecek yeni kesmeleri engellemek için kesme alt programı başlangıcında 0

yapılmalı, kesme bitişinde ise INTE biti tekrar 1 yapılarak yeni kesmelerin oluşabilmesine olanak sağlanmalıdır.



Şekil 28 Kesme çeşitleri

6.4.2 PORTB değişiklik kesmesi

B portunun son 4 bitinden (7, 6, 5, 4) herhangi birisinde bir lojik seviye değişikliği olduğu anda oluşan kesmedir. Bu kesmenin oluşması isteniyor ise INTCON registerinin 3. biti (RBIE) 1 yapılmalıdır. Bununla birlikte B portunun belirtilen bitlerinde herhangi bir değişiklik oluşursa bu INTCON registerinin 0. biti (RBIF) set olur ve kesme meydana gelir. B portundaki değişiklikler bu portun durumunun daha önce bir register içerisine kaydedilmesi ve bu değerin çeşitli komutlar aracılığı ile yeni port değeri ile karşılaştırılması sonucunda algılanır. Kesme alt programı başlangıcında RBIF ve RBIE bitleri silinerek kesme anında başka kesmelerin oluşması engellenebilir. Eğer B portunun belirtilen bitlerinden sadece, bazılarındaki değişikliklerin kesme oluşturması isteniyorsa, aritmetik veya mantıksal komutlar aracılığıyla istenilen bitler bazı sayılarla karşılaştırılarak istenilen durum sonucunda RBIF biti silinebilir.

6.4.3 EEPROM'a yazma kesmesi

Bu kesme ise PIC içerisinde bulunan ve PIC'in enerjisinin kesilmesi durumunda içerisindeki veriyi koruyabilen EEPROM veri belleği içerisine veri yazmak için yapılan kesmedir. EEPROM' a veri yazabilmek için INTCON registerinin 6. biti (EEIE) ve EECON1 registerinin 4. bitinin (EEIF) durumları ve ayrıca yazma işleminin

tamamlanmasına kadar geçen süre içinde dikkate alınması gereken başka faktörler de bulunmaktadır. Daha detaylı bilgi için microchip firmasının hazırlamış olduğu data sheet'lere bakılabilir.

Yukarıda sayılan 3 kesme çeşidinden genel olarak bahsedilmiştir. Tez çalışmasında ağırlıklı olarak TMR0 kesmeleri ve bu kesmeler sayesinde oluşturulan zaman geciktirme programları kullanıldığından TMR0 kesmesine ait açıklamalara ağırlık verilmiştir.

6.4.4 TMR0 taşma kesmesi

TMR0 kesmeleri PIC içerisinde RAM belleğin 01h adresinde bulunan ve özel bir register olan TMR0 sayıcısında meydana gelen sayı taşması (FFh'dan 00h'a geçiş) anında meydana gelen kesmelerdir. Bu kesme çeşidinin daha iyi anlaşılabilmesi için RAM bellek içerisinde bulunan TMR0 sayıcısı ve PIC'in iç yapısında bulunana bir RC ositatör devresi sayesinde çalışan WDT (Watch Dog Timer) sayıcısına ait özellikler aşağıda verilmiştir. Bu sayıcılar zaman kontrolü ya da sayımı yaptıkları için aynı zamanda zamanlayıcı olarak da adlandırılmaktadırlar.

6.4.4.1 TMR0 sayıcısı

- 8 bitlik bir sayıcıdır.
- Okunabilir ve yazılabilir.
- 8 bitlik programlanabilir prescaler'a (frekans bölme değerine) sahiptir.
- Dahili veya dış (RA4/TOCI ucundaki) clock darbelerini sayabilme özelliğine sahiptir.
- Sayma değeri FFh`tan 00h`a geçtiğinde (taşma anı) kesme oluşturabilir.
- Sayma sürekli artan yöndedir ve istenilen sayıdan başlatılabilir. En son FFh'a kadar sayabilmekte ve bu sayıdan sonra 00h'dan devam etmektedir. (Microchip, 2000)

TMR0 PIC osilatör devresinden gelen dahili sinyalleri veya RB0 ucundan gelen sinyalleri sayabilmektedir. Hangi kaynağın TMR0'ı saydıracağı ise OPTION registerin 5. biti (TOCS) tarafından belirlenir. Bu bitin 1 yapılması ile TMR0 kaynağı, RA4/TOCKI ucundan gelen sinyaller, 0 yapılması ile de osilatör uçlarından gelen sinyaller olmaktadır. TMR0'ı saydıran RA4 ise, TMR0 bu uçtan gelen sinyallerin yükselen veya düşen kenarı baz alınarak saydırılabilir. Bu özellik için OPTION registerinin 4. biti (TOSE) kullanılmalıdır.

TMR0 sinyal kaynağı olarak osilatör girişi seçilirse TMR0'a bu frekansın ¼'ü uygulanır. Yani PIC osilatör frekansı dörde bölünerek TMR0'a uygulanmaktadır. Örneğin, 4 Mhz frekansa sahip bir XT osilatör devresi kullanıldığında bu frekansın ¼'ü (1 Mhz) TMR0'ın

artmasına sebep olur. Bu frekansa dahili komut frekansı (DKF) olarak tanımlanır. Ayrıca DKF ile birlikte OPTION registerin ilk üç biti ile ayarlanan prescaler değeri de TMR0'ın artma zaman aralığını belirleyen ikinci bir faktördür. Zaman hesaplamaları zaman geciktirme bölümlerinde ayrıntılı olarak verilmiştir.

6.4.4.2 WDT (Watch Dog Timer) zamanlayıcısı

WDT herhangi bir dahili veya harici bir kaynak gerektirmeden çalışan bir RC osilatördür. Bu zamanlayıcının asıl görevi PIC'i bir döngüde kilitlenmekten kurtarmaktır. Böyle bir durum yazılımda bir hata veya harici elektriksel kıvılcımlar nedeniyle ortaya çıkabilir. Sürekli çalışması gerekmeyen uygulamalarda, (ör: hırsız alarmı) WDT kullanımı kolaylıklar sağlamaktadır. Böyle uygulamalarda PIC uyuma moduna (sleep) sokulur, ve çeşitli sensör vb. elemanların kontrolü belli zaman aralıklarıyla yapılır. WDT belli zaman aralıkları ile PIC'i reset ederek bu uyma modundan kurtarır. Sleep modunda iken PIC'in çalışması durur ve çok düşük bir akım çeker. WDT, PIC'e osilatör uçlarından bağımsız olarak sürekli kalp atışı sağlar. Bu kalp atışlarının, OPTION registerin ilk 3 biti ile ayarlanan frekans bölme değerine (prescaler) göre zaman aralıkları ayarlanabilir. WDT prescaler değeri 1:1 olduğunda 18 ms aralıklarla çalışır. Bu durumda her 18 ms süreyle PIC'i reset eder. Bu özellik her programda kullanılmaz, çünkü sürekli resete zorlanan bir PIC'in reset ile birlikte, kullanılan programa göre bazı registerlerinin durumu değişebilir. Bu yüzden WDT kullanılan programlarda, programın çalışmasına uygun düzenlemeler yapılmalıdır. Bunlardan birisi, WDT'nin prescaler değerinin değiştirilerek süresinin uzatılması olabilir. Eğer WDT' nin reset yapması engellenmek isteniyor ise WDT, CLRWDT komutu ile temizlenmelidir. WDT düzenli aralıklarla temizlenmezse PIC'i reset'e zorlar. Yukarıda verilen 18 ms'lik zaman dilimi prescaler oranı 1:128 olarak ayarlandığında 18x128 = 2304 ms' ye çıkarılabilir. Bu değerler PIC için nominal şartların bulunduğu (28 °C, 5V ± %10 besleme gerilimi) zamanlar için geçerlidir. Bu şartlar değişirse zaman aralığı da değişecektir. (Katzen, 2001., Predko, 1998., Microchip 2000., ve Altınbaşak, 2000)

WDT zamanlayıcısının kullanılmadığı uygulamalarda konfigürasyon bitleri ayarlanırken _WDT _OFF yapılmalıdır. Bu düzenleme Programın başında yapılabileceği gibi, kullanılan PIC programlayıcısında "fuses" veya "config" pencerelerinden de işaretleme ile etkili ya da etkisiz yapılabilir.

6.4.4.3 TMR0 ve WDT'ye prescaler değerinin atanması

TMR0 ve WDT'ye prescaler değerinin atanması için ayarlanacak olan OPTION registerin ilgili bitleri Çizelge 14'de verilmiştir. OPTION registerinin ilk üç biti prescaler'i (frekans

bölme değeri) ve 3. biti olan PSA da belirlenen prescaler değerinin hangi sayıcı için atanacağını belirlemektedir.

Bir prescaler (frekans bölme değeri), ilgili sayıcının sayma aralıklarını düzenlemeye yarar. Örnek olarak, 1:4 prescaler değerinin TMR0 için seçildiği bir programda OPTION registerinin ilk dört biti '1010' olarak ayarlanmalıdır. 4. bit prescaler değerinin TMR0 için ayarlanmasını, ilk üç bit de prescaler oranının 1:4 olmasını sağlar. Prescaler oranının 1:4 olması ile TMR0 sayıcısına uygulanan dahili komut saykılı (DKS) 4'e bölünerek TMR0' a uygulanır. 4 Mhz osilatör frekansı ile beslenen bir pic için dahili komut frekansının (DKF) 1 Mhz olduğunu söylemiştik. F = 1/T (Frekans = 1/Peryot) olduğundan, uygulanan sinyallerin zaman aralıkları (DKS, dahili komut saykılı) 1/1 = 1 µs olacaktır. O halde TMR0' ın sayı artışı 1 x 4 = 4 µs zaman aralıkları ile gerçekleşir. Prescaler değeri 1:8 olduğunda aynı PIC için TMR0'ın sayma zaman aralığı 1 x 8 = 8 µs olur.

6.4.4.4 TMR0'dan WDT'ye ve WDT'den TMR0'a prescaler değeri atamak

TMR0 sayıcını direkt olarak besleme kaynağına bağlamak için prescaler değeri WDT'ye atanmaktadır. Bu durumda TMR0'a yazmak için kullanılan komutlar (ör: CLRF, MOVWF, BSF...) prescaler değerini silerek yeni prescaler değeri girmeye hazırlar. Prescaleri TMR0'dan WDT'ye ve WDT'den TMR0'a atama işlemlerinde prescalerin silinmesi nedeniyle PIC'in çalışması esnasında oluşabilecek resetleri önlemek için aşağıdaki program parçaları mutlaka kullanılmalıdır. (Microchip, 2000 ve Altınbaşak 2000)

TMR0'dan → WDT'ye prescaler değeri atanırken;

```
BCF     STATUS, RP0     ;Bank0'a geç
CLRF    TMR0            ;TMR0'ı ve prescaler'i sil
BSF     STATUS, RP0     ;Bank1'e geç
CLRWDT                  ;WDT'yi sil
MOVLW   b'xxxx1xxx'     ;Yeni prescaler değerini seç
MOVWF   OPTION_REG      ;Bu değeri OPTION registere yükle
BCF     STATUS, RP0     ;Bank0'a geç
```

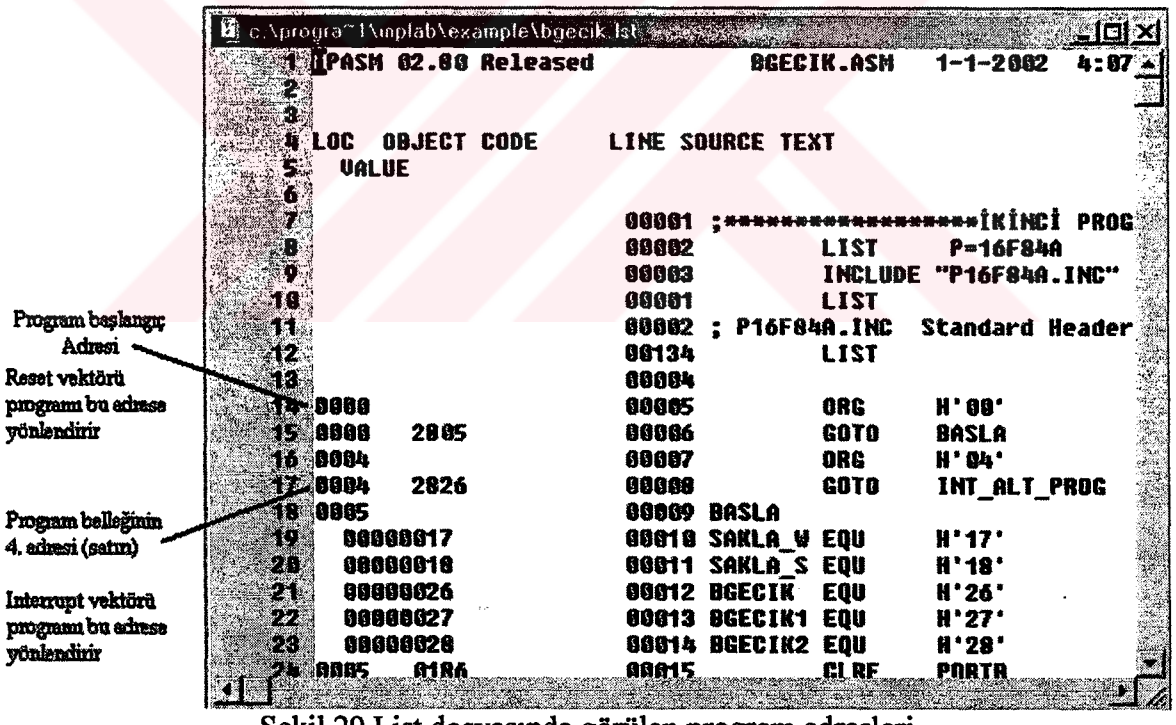
WDT'den → TMR0'a prescaler değeri atanırken;

```
CLRWDT                  ;WDT'yi ve prescaler değerini sil
BSF     STATUS, RP0     ;Bank1'e geç
MOVLW   b'xxxx0xxx'     ;TMR0'ı, yeni prescaleri ve sinyal kaynağını seç
MOVWF   OPTION_REG      ;Bu değeri OPTION registerine yaz
BCF     STATUS, RP0     ;Bank0'a geç
```

CLRWDT komutu STATUS registerinin 3. ve 4. biti olan TO ve PD bitlerinin içeriğini etkileyerek 1 yapar. Böylece PIC'in reset'e gitmesi engellenmiş olur.

6.4.4.5 TMR0 taşma kesmesinin gerçekleştirilmesi

TMR0 kesmesinin oluşabilmesi için öncelikle bütün kesmeleri aktif yapan INTCON registerinin 7. biti olan GIE (Global Interrupt Enable)'nin değeri 1 yapılmalıdır. Ayrıca bu registerin 5. biti (TOIE) 1 yapılırsa, PIC TMR0 kesmelerini algılayabilecek duruma geçer. Bu durumda sayıcıda bir taşma meydana gelirse INTCON registerinin 2. biti set (1) olur ve o anda ana programın çalışması kesilerek, 04h adresinde bulunan program satırına geçilir. Herhangi bir kesme meydana geldiğinde çalışan program, PIC program belleğinin 4. satırına geçer. Bunu sağlayan adres yönlendiricisi "interrupt vektör" olarak anılmaktadır. "Reset vektörü" olarak adlandırılan ikinci bir vektör de program çalışırken bir reset oluştuğunda, program akışını, program belleğinin 00h adresinde bulunan ilk satırına yönlendirir. Program adresleri, yazılıp derlenen bir programın .LST uzantılı dosyasında rahatlıkla izlenebilir. Kesme kullanılan bir program örneği için .LST dosyası Şekil 29'da görüldüğü gibidir.



Şekil 29 List dosyasında görülen program adresleri

Reset vektörü ve interrupt vektörünün birbiriyle çakışmadan uyum içinde çalışmalarının sağlanması, programın sıhhati açısından önemlidir. Bu yüzden 00h ve 04h adresindeki satırlar ORG deyimi ile düzenlenmelidir. ORG deyimi program komutlarının istenilen program adres satırına yazılabilmesini sağlar. Kesme kullanılmayan programlarda kullanılmasına gerek yoktur. Bir program normalde çalışmasına 00h adresinden başlar ve CALL, GOTO gibi dallanma komutları kullanılmadığı sürece sırayla yazılan komut

satırlarını çalıştırır. Bu komutların, PC (program sayıcı) içeriğini değiştirerek programın başka bir etikete dallanmasına sebep olduğuna komutların anlatıldığı bölümlerde değinilmişti. Bu komutlar ile interupt ve reset vektörünün arasında benzerlik bulunsa da çok önemli bir fark vardır. Komutlar ile istenilen adres satırına gidilir fakat, vektörler programı hep aynı adres satırına yönlendirir.

Kesme kullanılan bir programda ana programın ilk komutu mutlaka 00h, kesme alt programının ilk komutu da 04h adres satırına yazılmalıdır. Bu ilk satıra yazılabilecek bir GOTO komutu ile program akışı başka adreslere yönlendirilir. Yönlendirilen programın adresi önemli değildir. Bu şekilde, kesme kullanılan bir program aşağıdaki formatta yazılabilir.

```
                ORG         00h                 ;Pic'e enerji verildiğinde ya da bir
                GOTO        ANA_PROGRAM         ;reset oluştuğunda ana programdan
                                                ;başla
                ORG         04h                 ;Kesme oluştuğunda kesme alt
                GOTO        KESME_ALT_PROGRAMI  ;programını çalıştır

ANA_PROGRAM                                     ;Ana program komutları
                ....................            ;   "
                ....................            ;   "
KESME_ALT_PROGRAMI                              ;Kesme alt programı komutları
                ....................            ;   "
                ....................            ;   "
                RETFIE                          ;Ana Programda kalınan yere dön
```

Kesme alt programı istenilirse GOTO komutu kullanılmadan ORG 04h satırının hemen altına yazılabilir. Bu yazılanlar bütün kesmeler için geçerlidir.

Bir kesme oluşturabilmek için yukarıda bahsettiğimiz registerlerin (INTCON, OPTION) içeriklerinin ayarlanması ana program başlangıcında yapılır. TMR0 kesmesi için INTCON registerinin içeriği B'1x1xx0xx' şeklinde yüklenmelidir. Ayrıca kesme aralıklarının ayarlanmasında etkili olan OPTION registerin içeriği de atanacak olan prescaler değeri ve bu değerin hangi sayıcıya atanacağına bağlı olarak düzenlenmelidir. TMR0 kesmeleri için örnek programlar zaman geciktirme rutinleri bölümünde verilmiştir.

6.5 Kesme Alt Programlarında W ve STATUS Register İçeriklerini Korumak

Bir kesme alt programı çalışırken W registeri ve STATUS registerinin içeriklerini değiştirebilecek komutlar kullanılmış ise, kesme alt programından ana programa dönüşte bu registerler yeni içeriklere sahip olabileceklerdir. Bu durum ana programda bazı karışıklıklara sebep olabilir. Örnek verecek olursak, ana programda bir aritmetik işlem yapılıp işlem sonucunun 0 olup olmadığı, yani STATUS registerinin 2. bitinin (zero flag) 1

olup olmadığının tespit edilmeye çalışılacağı (BTFSS STATUS,2) bir anda kesme meydana geldiğinde, program kesme alt programını çalıştıracaktır. Kesme alt programında STATUS registerinin 2. biti değişmiş ise, ana programa dönüşte yanlış bir sonuca götürecektir. Bütün bu gibi olumsuz etkileri önlemek için bir kesme alt programının başlangıcında STATUS ve W registerleri önceden tanımlanmış olan iki registere yüklemek ve alt program çıkışında bu registerleri tekrar W ve STATUS registere geri yüklemek gerekir. Bu yüzden W ve STATUS registerinin saklandığı bir kesme alt programı aşağıdaki gibi olmalıdır.(Matic ve Andric, 2000)

```
;Kesme alt programı başlangıcı
MOVWF     W_TEMP            ; W → W_TEMP
SWAPF     STATUS, W         ; STATUS Reg. Swap yap ve W registere yükle
MOVWF     STATUS_TEMP       ; W → STATUS_TEMP
.....                       ; Kesme alt programı komutları
.....                       ;       "
.....                       ;       "
SWAPF     STATUS_TEMP, W    ; STATUS_TEMP Reg. tekrar swap yap ve W reg. yükle
MOVWF     STATUS            ; W → STATUS
SWAPF     W_TEMP, F         ; W_TEMP reg. Swap yap ve yine W_TEMP reg. yükle
SWAPF     W_TEMP, W         ; W_TEMP reg. tekrar swap yap ve W reg. yükle
RETFIE
;Kesme alt programı sonu
```

Burada W_TEMP, W registerinin, STATUS_TEMP de SATATUS registerinin kesme başlangıcındaki değerlerinin saklandığı registerlerdir. Registerlerin isimleri önemli olmayıp kullanıcının tanımlayacağı herhangi iki file register bu işi görecektir. Registerlere yükleme işleminin SWAPF komutuyla yapılmasının nedeni, bu komutun STATUS registeri etkilememesinden dolayıdır. Bu komut bir file registerin ilk dört biti ile son dört bitini yer değiştirdiğinden, kesme sonundaki geri yüklemelerde W register içeriği (W_TEMP) iki kere SWAPF komutu kullanılarak yüklenmelidir.

BÖLÜM VII

ZAMAN GECİKTİRME RUTİNLERİ

7.1 Zaman Geciktirme Çeşitleri

PIC mikrodenetleyicilerinde program yazarken belli bir zaman gecikmesinin istendiği durumlarda iki türlü gecikme sağlanabilir. Birincisi, CALL komutu ile çağrılan bir alt programın çalışmasının bitmesine kadar geçen sürenin kullanılması şeklindedir. İkinci olarak, çalışması sürekli olan, TMR0 sayıcısının zaman aşımı anlarını kullanarak oluşturulan kesme alt programları sayesinde elde edilen gecikmelerdir.

Gecikmelere ait açıklamalara geçmeden önce bazı tanımlamaların yapılması faydalı olacaktır. Daha önceki bölümlerde belirtildiği gibi PIC mikrodenetleyicilerinde 4 farklı osilatör çeşidi kullanılmaktadır.. Bu nedenle PIC için yazılan bir programın çalışması kullanılan osilatörün özelliğine göre farklılık arz etmektedir. Bu farklılığın temel sebebi osilatörler arasındaki frekans farkıdır. Yüksek frekanslı bir osilatör devresi ile beslenen bir PIC için komutların icra süreleri kısa olacak, yani komutlar daha hızlı çalışacaktır. Bu nedenle C ile ifade ettiğimiz komut saykıl sayısı aynı olmakla birlikte, komut icra (işleme) süresi (KİS), kullanılan osilatöre göre değişkenlik göstermektedir. Bir program yazılmadan önce kullanılacak olan osilatör tipi belirlenmeli ve ona göre hesaplamalar yapılmalıdır. Bütün osilatörler için ortak olan bazı hesaplamalar aşağıda verilmiştir.

PIC16F84 osilatör girişine uygulanan frekansın ¼'ü dahili komut frekansı (DKF) olmaktadır. Periyot = 1 / Frekans olduğundan, dahili komut saykılı (DKS) = 1/ DKF ve bir komut için işleme süresi, KİS = C x DKS olmaktadır.

Bir komut için örneğin, DECFSZ SAY,1 için komut işleme süresini 4 Mhz ve 10 Mhz kristal osilatörler için hesaplayacak olursak,

4 Mhz kristal osilatör için; DKF = 4/4 = 1 Mhz, DKS = 1/1 = 1 µs,
KİS = 1(2) x 1 = 1 (2) µs olacaktır.

Komutların anlatıldığı bölümlerdeki çizelgelerde verilen C sayılarından bazıları 1(2) şeklinde gösterilmiştir.Bu komutlar BTFSC, BTFSS, INCFSZ ve DECFSZ komutlarıdır.

Bu komutlardan BTFSC ve BTFSS, bir file registerin herhangi bir bitini test ederek bir karar verme işlemini yerine getirirler. Diğer iki komut ise hem bir aritmetik işlem yapan hem de işlem sonucunda bir karar veren komutlardır. Bütün bu komutlarda komut işlemesi sonunda sorulan soruya evet cevabı veriliyor ise yalnızca o işleme için C = 2, diğer işlemeler için C = 1 alınmalıdır. Yukarıdaki DECFSZ komutunun sorusu şöyledir; SAY registerini 1 azaltıldığında, sonuç 0 mı?.

10 Mhz kristal osilatör için; DKF = 10/4 = 2,5 Mhz, DKS = 1/2,5 = 0,4 μs,
KİS = 1(2) x 0,4 = 0,4 (0,8) μs olacaktır.

Görüldüğü gibi osilatör frekansı ile komutların işleme süresi ters orantılı olarak değişmektedir. Aşağıda vereceğimiz gecikme sürelerinin hesaplamalarında da bu özellikler dikkate alınmalıdır. Örneğin, 4 Mhz frekansa göre, 1 sn'lik bir gecikme sağlamak için hesaplanan sayılar, 10 Mhz'lik osilatör kullanıldığında 1 sn'den daha az bir gecikme sağlar.

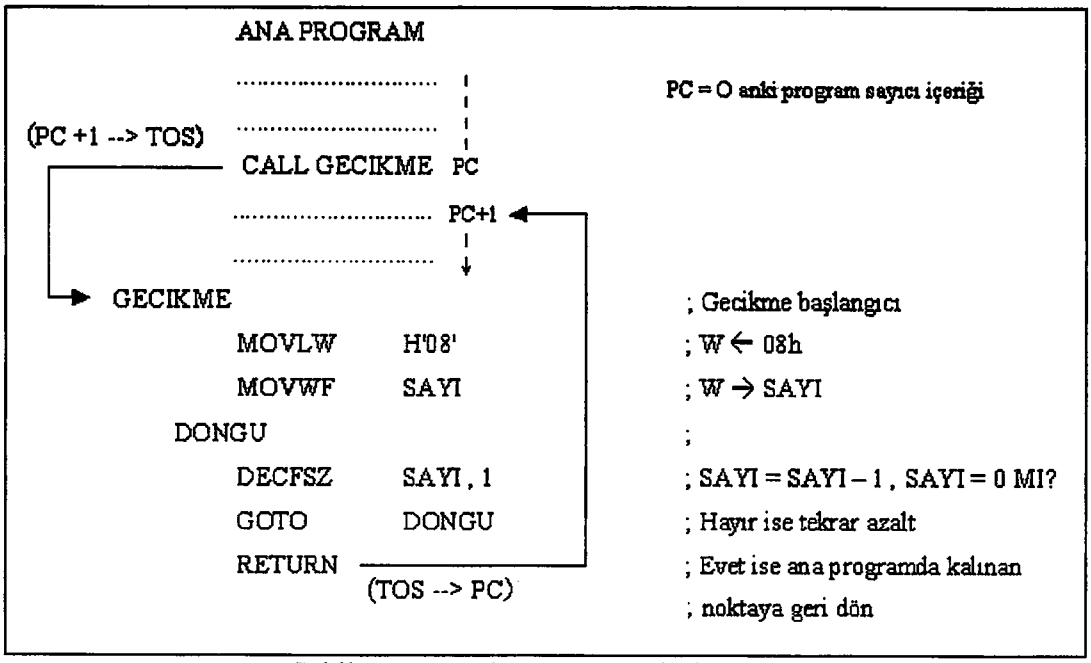
7.2 Döngülü Zaman Gecikmeleri

Bu tür gecikmeler, bir ana program çalışırken bir alt programın çağırılması ve çalışmasının bitmesine kadar geçen süre ile sağlanmaktadır. Zaman süresinin uzunluğuna göre de bu alt program programda tek döngülü, çift döngülü gibi isimlerle anılmaktadır. Bu şekilde yazılan bir programın çalışması Şekil 30'da verilmiştir..

7.2.1 Tek döngülü zaman gecikmesi

Yazılan alt programın asıl amacı bekletmek olduğundan belirli sayılar önceden tayin edilir, W registere ardından başka bir registere yüklenir ve bu sayı 0 oluncaya kadar 1 eksiltilir. Bu eksiltmelerin her defasında bir komut (ör:INCF, INCFSZ) komut kullanılacağından komutun işleme süresi olan, KİS ile komutun işleme sayısının çarpımı kadar bir gecikme elde edilmektedir.

Şekil 30'da görüldüğü gibi program komutları icra edilirken CALL GECIKME satırına geldiğinde program GECIKME etiketinin (alt programın) bulunduğu satıra gider. Gitmeden önce o anda bulunulan satırın adresinin bir fazlası (PC+1) STACK registerinin üst hafiza bölümüne (TOS) yazılır. Alt program çalışmasını bitirince yani RETURN komutunun bulunduğu satıra gelindiğinde STACK register'e yazılan adres program counter'a yüklenir ve program bu adrese dönerek (CALL komutunun bulunduğu satırın bir alt satırı) çalışmasına devam eder.



Şekil 30 Döngülü zaman geciktirmeleri

Ana programı ile birlikte bir örnek verecek olursak, aşağıdaki gibi basit bir program yazabiliriz. Bu program, B portunun ikinci biti olan RB2'nin bir süre enerjilenip bir süre enerjisiz kalması ve işlemin bu şekilde devam etmesini sağlar. RB2 ucuna bir LED bağlanılırsa led'in belli bir süre ile yanıp sönmesi izlenebilir. Fakat bu değişim zaman süresi çok kısa olduğundan göz ile görülemez. Göz ile değişimi görebilmek için daha uzun zaman gecikmesi sağlayan çift veya daha fazla döngülü zaman gecikme döngüleri kullanılabileceği gibi aynı alt program arka arkaya da çağrılabilir.

```
        LIST P=16F84A              ;PIC16F84' ü tanıt
        INCLUDE "P16F84A.INC"      ;P16F84.INC dosyasını ekle
        SAYI    EQU  H'0D'         ;SAYI registerini 0Dh adresine ata
        CLRF    PORTB              ;Portb' yi sil
        BSF     STATUS,5           ;BANK1 E GEÇ
        MOVLW   B'00000000'        ;W ← B'11111111'
        MOVWF   TRISB              ;W (H'FF') → TRİSB (portb'yi çıkış yap)
        MOVLW   B'11111111'        ;W ← B'11111111'
        MOVWF   TRISA              ;W → TRISA (porta yı giriş yap)
        BCF     STATUS,5           ;Bank0' a geç
BASLA
        BSF     PORTB,2            ;RB2 set yap
        CALL    GECIKME            ;GECIKME alt programını çağır
        BCF     PORTB,2            ;RB2 reset
        CALL    GECIKME            ;GECIKME alt programını çağır
        GOTO    BASLA              ;BASLA etiketine dön
GECIKME                            ;Tek döngülü gecikme alt programı
        MOVLW   H'FF'              ;      "
        MOVWF   SAYI               ;      "
```

```
DONGU                           ;       "
        DECFSZ      SAYI,1      ;       "
        GOTO        DONGU       ;       "
        RETURN                  ;       "
        END                     ;Program sonu
```

Tek döngülü zaman geciktirme alt programlarındaki, zaman gecikmesine (T) ait birkaç µs hatalı olabilen bir formül aşağıda verilmiştir.

T = 3 x SAYI x KİS µs

Yukarıdaki örnek için (4 MHz osilatör kullanıldığında),

T = 3x255x1 = 765 µs olarak hesaplanır.

Bu formül kullanılırken KİS, C sayısı 1 alınarak hesaplanmalıdır. Yukarıdaki örnek için bunun sebebi açıklanacak olursa, SAYI registeri içerisine yüklenen FFh (255d) sayısı DECFSZ komutunun her işlemesinde 1 eksiltildiğinden ancak 255. komutta C = 2 olmaktadır. Hassas hesaplamalarda her komutun işleme sayısı ve C dikkate alınmalıdır. Bu gecikme için gerçek zamanı hesaplamak istersek CALL komutu için 2x1 = 2, MOVLW için 1x1 = 1, MOVWF için 1x1 = 1, DECFSZ için (254x1)+(1x2) = 256, GOTO için 2x254 = 508, RETURN için 2x1 = 2 µs ve toplam olarak, 2+1+1+256+508 = 768 µs olmaktadır. Bu aradaki 3 µs lik süre önemli ise tek tek hesaplama yapılmalıdır.

7.2.2 Çift döngülü zaman gecikmesi

Çift döngülü zaman gecikmesi alt programı için örnek ve zaman gecikmesi formülü aşağıdaki gibidir.

```
GECIKME
        MOVLW       H'04'       ;ÇİFT DÖNGÜ İLE ZAMAN GECIKMESİ
        MOVWF       SAYI        ;ALT PROGRAMI
DONGU                           ;       "
        MOVLW       H'FF'       ;       "
        MOVWF       SAYI1       ;       "
DONGU1                          ;       "
        DECFSZ      SAY1,F      ;       "
        GOTO        DONGU1      ;       "
        DECFSZ      SAY,F       ;       "
        GOTO        DONGU       ;       "
        RETURN                  ;       "
```

T = 3 x SAYI x SAYI1 x KİS µs

7.2.3 Üç döngülü zaman gecikmesi

Çift döngülü zaman gecikmesi alt programı için örnek ve zaman gecikmesi formülü aşağıdaki gibidir.

```
GECIKME
        MOVLW     H'04'       ;ÜÇLÜ DÖNGÜ İLE ZAMAN GECIKMESİ
        MOVWF     SAYI        ;ALT PROGRAMI
DONGU                         ;       "
        MOVLW     H'FF'       ;       "
        MOVWF     SAYI1       ;       "
DONGU1                        ;       "
        MOVLW     H'FF'       ;       "
        MOVWF     SAYI2       ;       "
DONGU2                        ;       "
        DECFSZ    SAYI2,F     ;       "
        GOTO      DONGU2      ;       "
        DECFSZ    SAYI1,F     ;       "
        GOTO      DONGU1      ;       "
        DECFSZ    SAYI        ;       "
        GOTO      DONGU       ;       "
        RETURN                ;       "
```

T = 3 x SAYI x SAYI1 x SAYI2 x KİS μs

7.3 TMR0 Zaman Aşımı Kesmesiyle Sağlanan Zaman Gecikmeleri

Bu tez çalışmasında ağırlıklı olarak TMR0 zaman aşımı kesmesi kullanarak elde edilen zaman gecikmeleri kullanılmıştır. Bu tür bir gecikme için bir kesme alt programı, ve ana program içerisinde kesme alt programı ile değiştirilen register içeriklerinin karşılaştırıldığı program parçaları kullanılmıştır. Ana program içerisinde W registere hesap ile belirlenen sayılar yüklenerek, bu sayı ile program başlangıcında tanımlanmış bir veya daha fazla "file register" içeriği SUBWF, ADDWF, IORWF gibi aritmetik veya lojik işlem yapan komutlar ile işleme tabi tutulup, ardından STATUS registerinin 0. biti ve 2. biti olan CARRY (C) ve ZERO (Z) bitlerinin durumu sorgulanarak istenilen özellikte gecikmeler elde edilmiştir. Bir aritmetik ve mantıksal işlem sonucu 0 ise Z bitinin içeriği, o işlem sonunda 1 olmaktadır. İşlem sonucunda bir taşma varsa C bitinin içeriği 1 olmaktadır. Bu bitlerin durumu sorgulanarak karşılaştırılan iki sayının birbirinden büyük, küçük, birbirine eşit, olma durumları belirlenebilmektedir. Bu çalışmada genellikle SUBWF komutu kullanılmıştır. Bu komuta ait karşılaştırma durumları Çizelge 7'de verilmiştir.

Bu tez çalışmasında endüstriyel sistem kontrolü uygulama programlarında iki çeşit kesme gecikmesi kullanılmıştır. Bu gecikmelere ait blok diyagram ve program parçalarına ait açıklamalar bu bölümde verilmiş ve bu kesme gecikmelerinin kullanıldığı programların anlatımında, blok diyagramlarda fazla yer tutmaması için, sadece gecikmenin çeşidi ve ne için kullanıldığı belirtilmiştir. Sırasıyla anlatılacak olan birinci kesme gecikmesi programı "Kesme1" gecikmesi ve ikinci kesme gecikmesi programı da "Kesme2" gecikmesi olarak isimlendirilmiştir.

7.3.1 Kesmel zaman gecikmesi programı

Bu TMR0 kesme gecikmesi örneğinde, çalışması PLC'lerde "off delay timer" olarak bilinen, ilk enerjilendiğinde konum değiştiren ve belli bir süre sonra eski konumuna geri dönen TIMER'ların çalışma prensibine benzer şekilde çalışan, bir gecikme programı hazırlanmıştır. Bu tür bir gecikmeye örnek olarak aşağıdaki programı verebiliriz..

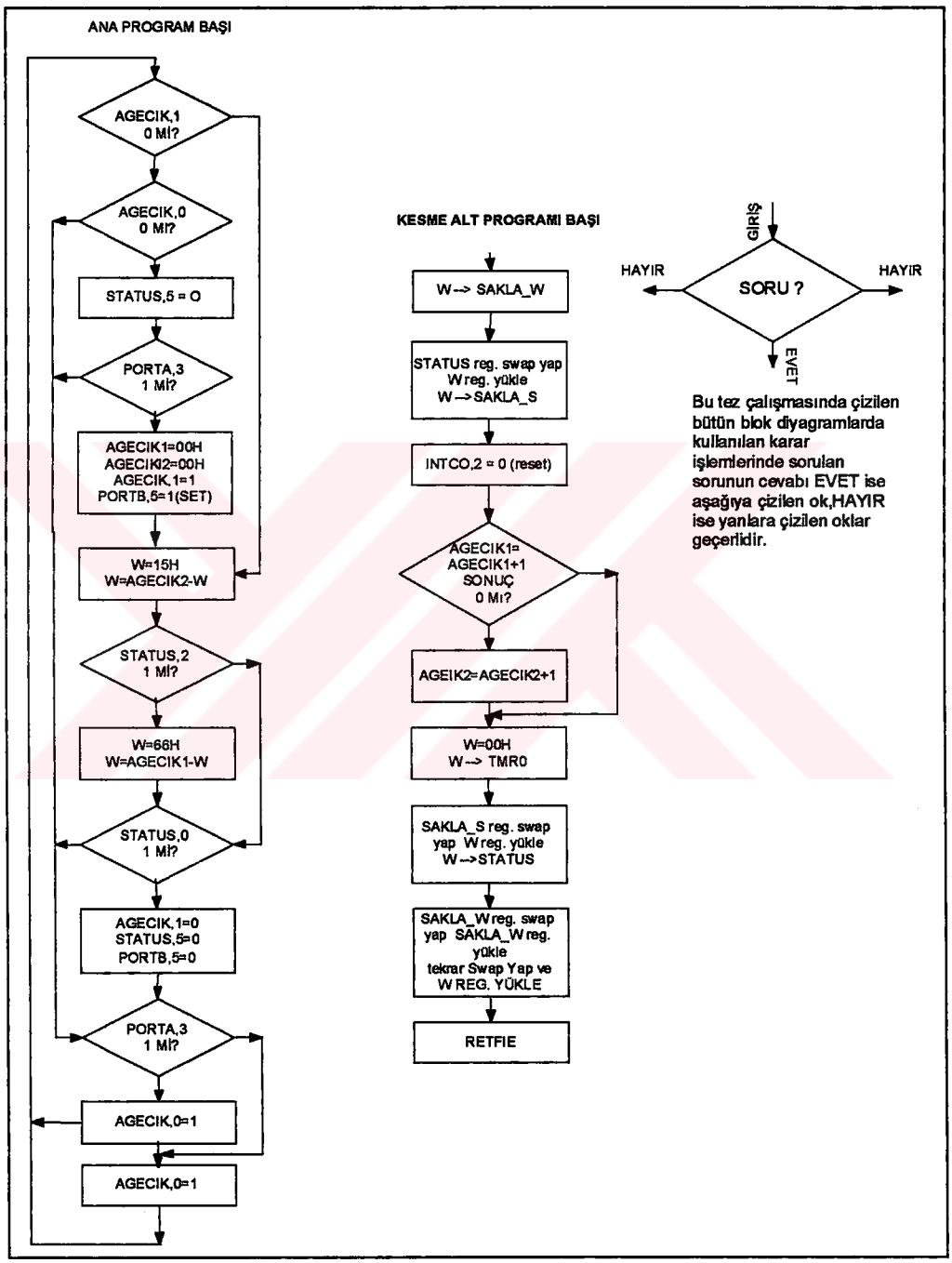
Örneğimizde PIC16F84'ün RA3 ucuna uygulanan bir giriş sinyalinin pozitif (yükselen) kenarı ile RB5 ucu hemen başlamak üzere set olacak ve belirlenen süre kadar bu set durumunu koruyacak, süre sonunda reset olacaktır. Eğer RB5 Lojik 1 olmuş ise reset oluncaya kadar RA3 ucundan gelen sinyaller dikkate alınmayacak, RB5'in reset olduğu anda RA3'ün değeri 1 ise bu değerin önce 0 daha sonra tekrar 1 olduğu (yükselen kenar) durumda RB5 ucunda tekrar gecikmeli bir çıkış oluşacaktır. Böyle bir örnek için ana program ile kesme alt programı aşağıda verilmiştir. Bu örnek için akış şeması ise Şekil 31'de verilmiştir.

```
        LIST    P=16F84A            ;16F84A'yı tanıt
        INCLUDE "P16F84A.INC"       ;16F84A.INC dosyasını ekle
        ORG     H'00'               ;Program başlangıç adresi H'00'
        GOTO    BASLA               ;BASLA etiketine git
        ORG     H'04'               ;interrupt(kesme sinyali gelirse)
        GOTO    INT_ALT_PROG        ;INT_ALT_PROG etiketine git
BASLA
SAKLA_W     EQU     H'17'           ;SAKLA_W registerini ve adresini tanımla
SAKLA_S     EQU     H'18'           ;SAKLA_S        "
AGECIK      EQU     H'26'           ;AGECIK         "
AGECIK1     EQU     H'27'           ;AGECIK1        "
AGECIK2     EQU     H'28'           ;AGECIK2        "
        CLRF        PORTB           ;PORTB'yi sıfırla
        CLRF        PORTA           ;PORTA'yı sıfırla
        BSF         STATUS,5        ;BANK1'e geç
        MOVLW       B'00000000'     ;W ← H'00'
        MOVWF       TRISB           ;W (H'00') → TRİSB (PORTB'yi çıkış yap)
        MOVLW       B'11111111'     ;W ← H'FF'
        MOVWF       TRISA           ;W → TRISA (PORTA'yı giriş yap)
        CLRWDT                      ;WDT'yi sil
        MOVLW       B'10001111'     ;W ← B'10001111'
        MOVWF       OPTION_REG      ;W → OPTION_REG
        BCF         STATUS,5        ;BANK0'a geç
        MOVLW       H'00'           ;W <-- H'00'
        MOVWF       TMR0            ;W --> TMR0   (TMR0'ı sıfırla)
        MOVLW       B'10100000'     ;W <-- B'10100000' (TMR0 taşma kesmesi aktif)
        MOVWF       INTCON          ;W --> INTCON
START                               ;Programın çalışmasına ait açıklamalar aşağıda verilmiştir
        BTFSC       AGECIK,1        ;
        GOTO        AGECIK_SAY      ;
        BTFSC       AGECIK,0        ;
        GOTO        AGECIK_SET      ;
        BCF         STATUS,5        ;
```

```
        BTFSS       PORTA,3             ;
        GOTO        AGECIK_SET          ;
        CLRF        AGECIK1             ;
        CLRF        AGECIK2             ;
        BSF         AGECIK,1            ;
        BSF         PORTB,5             ;
AGECIK_SAY                              ;
        MOVLW       H'15'               ;
        SUBWF       AGECIK2,W           ;
        BTFSS       STATUS,2            ;
        GOTO        AGECIK_ KONTROL;
        MOVLW       H'66'               ;
        SUBWF       AGECIK1,W           ;
AGECIK_ KONTROL                         ;
        BTFSS       STATUS,0            ;
        GOTO        AGECIK_SET          ;
        BCF         AGECIK,1            ;
        BCF         STATUS,5            ;
        BCF         PORTB,5             ;
AGECIK_SET                              ;
        BTFSS       PORTA,3             ;
        GOTO        AGECIK_SIL          ;
        BSF         AGECIK,0            ;
        GOTO        AGECIK_CIKIS        ;
AGECIK_SIL                              ;
        BCF         AGECIK,0            ;
AGECIK_CIKIS                            ;
        GOTO        START
INT_ALT_PROG                            ;Kesme alt programı başlangıcı
        MOVWF       SAKLA_W             ;W → SAKLA_W
        SWAPF       STATUS,W            ;Status reg. swap yap ve W'ye yükle
        MOVWF       SAKLA_S             ;W → SAKLA_S
        BCF         INTCON,2            ;TOIF <-- '0' (zaman aşımı yok)
        INCFSZ      AGECIK1,F           ;AGECIK1 = AGECIK1+1, sonuç 0 mı?
        GOTO        CIKIS               ;Sonuç sıfır değilse
        INCF        AGECIK2,F           ;Sonuç 0 ise, AGECIK2 = AGECIK2+1
CIKIS                                   ;
        MOVLW       H'00'               ;TMR0'ı sıfırla
        MOVWF       TMR0                ;       "
        SWAPF       SAKLA_S,W           ;W ve Status reg. yeniden yükle
        MOVWF       STATUS              ;       "
        SWAPF       SAKLA_W,F           ;       "
        SWAPF       SAKLA_W,W           ;       "
        RETFIE                          ;Ana programa geri dön
        END                             ;Program sonu
```

Programdan ve Şekil 31'den de görüldüğü gibi A portunun 3. bitinin 1 olduğu anda B portunun 5. biti hemen lojik 1 olur, ve belli bir süre bu durum devam eder. Süre dolunca B portunun 5. biti 0 olur. RB5 lojik 1 iken gecikme süresi tekrar başlatılamaz. Bu süre zarfında RA3'ten gelen sinyaller değerlendirilmez. Blok diyagramdan da görüleceği gibi ilk anda AGECIK registerinin 1. biti ve 0. biti 0 dır. Bu anda RA3'ün değeri 1 ise AGECIK1 ve AGECIK2 registerlerinin içerikleri resetlenir, AGECIK registerinin 1. biti ve

RB5 set edilerek çıkış vermesi sağlanır. Gecikme süresi bu anda başlar. Bu sürenin tayini W registere yüklenecek sayılara bağlı olarak değişecektir. Bundan sonra sırayla takip edilecek olursa AGECIK2 registeri içeriğinden W registerine yüklenen sayı çıkarılacak ve arkasından bu iki sayının eşit olup olmadığı sorulacaktır (zero flag 1 ise eşittir).



Şekil 31 Kesme1 gecikmesine ait akış şeması

İlk anda AGECIK2 registeri değeri 0 olduğundan Status registerinin 0. biti olan carry flag 'in 1 olup olmadığı sorgulanacaktır. AGECIK2 registerinin içeriği W registerden (15h) büyük veya eşit ise Carry flag 1 olur. İlk anda AGECIK2 registerinin içeriği 15h'dan

küçük olduğundan tekrar RA3'ün durumuna bakılarak çıkılır. Eğer RA3 hala 1 ise AGECIK registerinin 0. biti 1, değilse 0 yapılarak çıkılır. Bu bit, bir gecikme başlamış ve bitmiş olduğu anda RA3 ucu 1 ise tekrar bir gecikmenin başlamasını engeller. Böylelikle RA3 ucundan gelen sinyallerin (eğer o anda gecikme yoksa) daima yükselen kenarında bir gecikme başlatılır. İkinci ve takip eden diğer döngülerde program, AGECIK registerinin 1. biti 1 olduğundan sayı karşılaştırmasının yapıldığı bölüme dallanır. Tekrar AGECIK2 registerinin içeriğinden 15h sayısı çıkartılarak karşılaştırma yapılır. Bu arada kesme meydana gelmiş ise AGECIK1 ve AGECIK2 registerlerinin içeriği değişmiştir. Kesme alt programı incelenecek olursa AGECIK1 registerinin içeriğinin 0 olduğu zamanlarda AGECIK2 registerinin içeriği 1 artırılmaktadır. AGECIK1 registerinin içeriğinin 0 olduğu durumlar her FFh (256d) sayısından 00h sayısına geçtiği anlardır. Bir kesme meydana geldiğinde AGECIK1 registeri 1 artırılır, eğer sonuç 0 olmuş ise AGECIK2 registeri de 1 arttırılır. Dolayısıyla AGECIK2 registerinin içeriğinin 1 artması için 256 defa kesme meydana gelmesi gerekir. Örneğimizdeki 15h (21d) sayısına ulaşmak için, prescaler değeri TMR0'dan WDT'ye 1/256 olarak atandığından ve her 256 komut saykılı sonunda kesme meydana geldiğinden dolayı (4 Mhz kristal osilatörlü PIC için) 256x256x21 = 1376256 µs geçmelidir. AGECIK2 registerinin içeriğinin 15h olduğu döngüde ve daha sonraki döngülerde AGECIK1 registerinin içeriği ile W registerine yüklenen 66h (102d) sayısı karşılaştırılacaktır (SUBWF komutu ile karşılaştırmalar Çizelge 7'de verilmiştir). 102 kesme daha sonunda AGECIK1 registeri 66h sayısına ulaşacak ve STATUS,0 = 1 olacağından AGECIK registerinin 1. biti ve A portunun 5. biti RA5 resetlenecektir. Bu anda RA3'ün değeri 1 ise AGECIK registerinin 0. biti 1 olacak ve tekrar bir gecikme olmasını engelleyecektir. Bundan sonra tekrar bir gecikme ile RB5'in çıkış vermesi için RA3 resetlenmeli ve tekrar set olmalıdır. AGECIK1 için yapılan karşılaştırmalar için geçen zaman ise 102x256 = 26112 µs olacaktır. Görülüğü gibi asıl zaman uzunluğunu belirleyen AGECIK2 registerinin karşılaştırıldığı (W registerine yüklenen) sayıdır. AGECIK1 için yapılan karşılaştırmalarla daha hassas bir süre elde edilebilmektedir. O halde toplam süre 1376256+ 26112 = 1402368 µs olacaktır.Bu süre yaklaşık olarak 1402368/1000000 = 1,4 sn olmakla birlikte kesme gecikmelerinin kullanıldığı, hassas zaman ölçümü gerektiren uygulamalarda kesme alt programının uzunluğu veya program içerisinde kullanılan döngülü zaman gecikmelerinin etkileri dikkate alınmalıdır.

W registerine yüklenen sayılar değiştirilerek farklı zaman gecikmeleri elde edilebilir. Bu gecikme zamanını formüle edecek olursak, W registerine AGECIK2 için sayı1 ve AGECIK1 için sayı2 değerini girdiğimizi ve 4 Mhz osilatör kullandığımızı varsayarsak;

T = (prescaler oranı x 256 x sayı1) + (prescaler oranı x sayı2) μs olur.

Bu tür gecikmeler endüstriyel sistem uygulama örneklerinin çözümünde verilen blok diyagramlarının bazılarında detaylı olarak çizilmeyip sadece giriş ve çıkış isimleri ile gecikme süreleri belirtilmiştir.

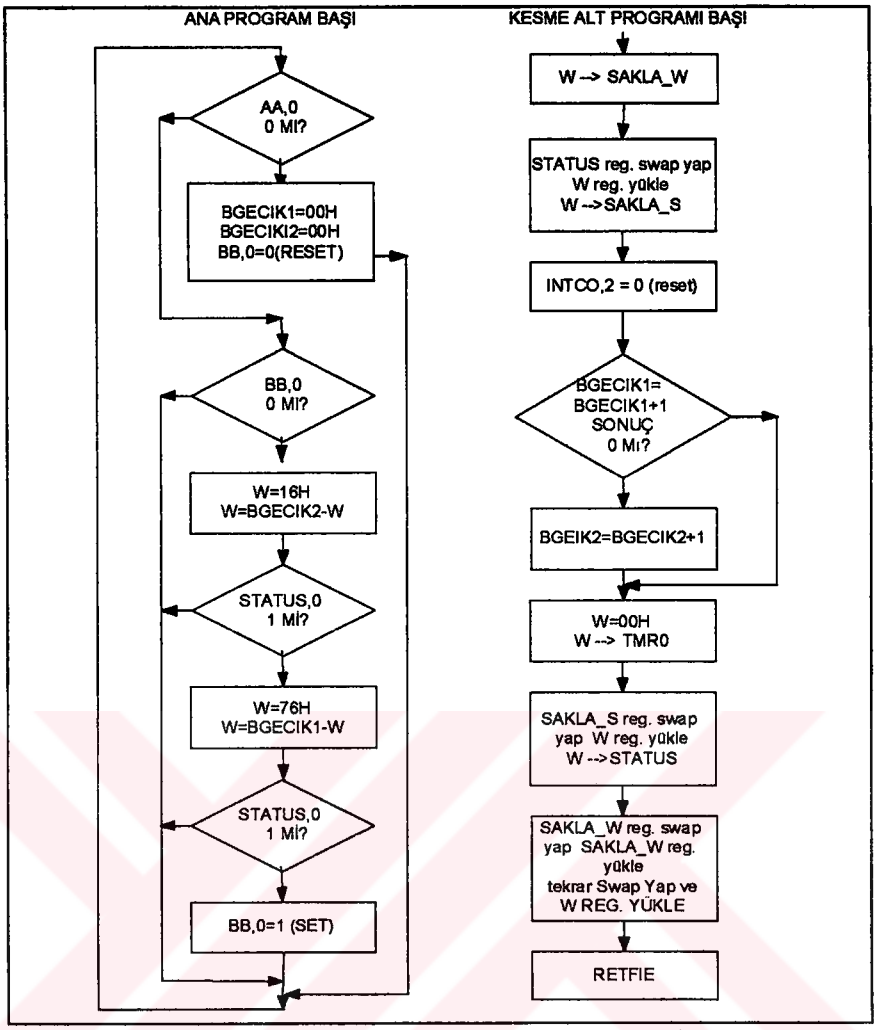
Bu çalışmada çizilen bütün blok diyagramlarda Karşılaştırma (karar verme) işlemlerinin yapıldığı akış diyagramı sembollerinde sorulan sorunun cevabı evet ise aşağıya, hayır ise yanlara çizilen oklar geçerlidir.

7.3.2 Kesme2 zaman gecikmesi programı

Bu örnekte ise PIC16F84'ün RA3 ucundan uygulanan bir sinyalden sonra gecikme başlamakta ve gecikme sonunda RB5 ucundan çıkış alınmaktadır. Eğer RA3 ucuna verilen sinyal en az gecikme süresi kadar uygulanmaz ise, RB5 ucundan çıkış alınamaz. Bu program PLC'lerde bulunan "On Delay Timer" olarak bilinen, gecikmeli olarak konum değiştiren, TIMER'lerin çalışmasına benzer. Beklenen süre sonunda RB5 ucunun set (1) olma durumunun devam etmesi ise RA3 ucuna uygulanan lojik 1 girişinin devamlılığına bağlıdır. RA3 girişindeki lojik seviye 0 olursa RB5 çıkışı reset (0) olur.

Bu şekilde çalışan ana program aşağıda verilmiştir. Bu programa ait kesme alt programı ve gecikme sürelerinin hesaplanması bir önceki örnekte olduğu gibidir. Değişiklik sadece kullanılan file register isimlerindedir. Programa ait akış şeması Şekil 32'de verilmiştir.

```
START
        BTFSC       PORTA,3             ;
        GOTO        BGECIK_SAY          ;
        CLRF        BGECIK1             ;
        CLRF        BGECIK2             ;
        BCF         PORTB,5             ;
        GOTO        BGECIK_CIKIS        ;
BGECIK_SAY                              ;
        BTFSC       PORTB,5             ;
        GOTO        BGECIK_CIKIS        ;
        MOVLW       H'16'               ;
        SUBWF       BGECIK2,W           ;
        BTFSS       STATUS,0            ;
        GOTO        BGECIK_CIKIS        ;
        MOVLW       H'66'               ;
        SUBWF       BGECIK1,W           ;
        BTFSS       STATUS,0            ;
        GOTO        BGECIK_CIKIS        ;
        BSF         PORTB,5             ;
BGECIK_CIKIS                            ;
        GOTO START
```



Şekil 32 Kesme2 zaman gecikmesi programına ait akış şeması

7.4 TMR0 Kesmesi ve Döngülü ile Sağlanan Gecikmeler Arasındaki Farklar

Bu iki gecikme arasında çok önemli bir farkı belirtmekte yarar vardır. Döngülü zaman geciktirmelerinde gecikmenin başladığı andan bittiği ana kadar, yani alt programın bitişine kadar, PIC için yazılan ana program içerisinde hiçbir işlem yapılmaz. Ana program o anda durur. Mesela RA1 ucundan gelen sinyal ile RB1 ucunda 100 ms'lik ve RA2 ucundan gelen sinyal ile RB2 ucunda 100 ms'lik bir çıkış sağlanmasının istenildiği bir programda, döngülü zaman gecikmeleri kullanılırsa, önce RA1 ucuna bir sinyal geldiğini düşünürsek o anda RB1 set yapılır ve alt program çağrılarak bu set durumunun 100 ms bekletilmesi sağlanır. Fakat alt programın çalıştığı anda RB1 ucundan gelen bir sinyal varsa bu sinyal alt program bitinceye kadar dikkate alınamaz .

Döngülü zaman gecikmesi alt programları ile kesme gecikmesi alt programlarının bir arada kullanıldığı programlarda, döngülü zaman gecikmesinin devam ettiği anda kesme meydana gelebilir. Bu durumda en son yazılan program adresi TOS içerisine yazılır.

BÖLÜM VIII.

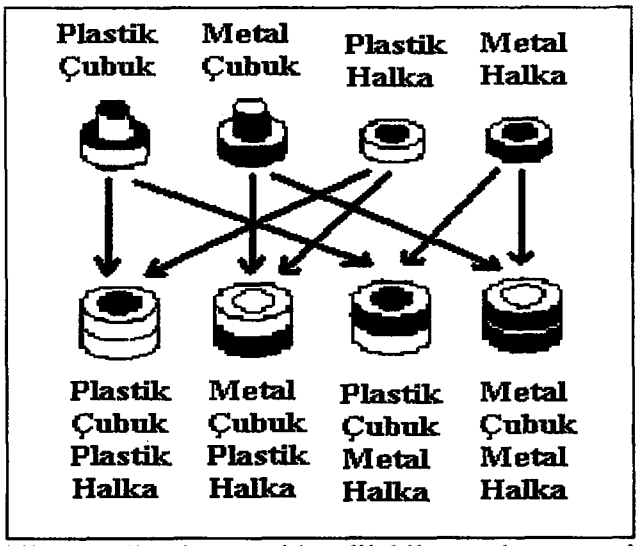
DENEYSEL ENDÜSTRİYEL SİSTEM İÇİN KONTROL UYGULAMALARI

Bu bölümde, önce deneysel endüstriyel sistem üzerinde yapılan değişiklikler anlatılmış ve PIC16F84 ile bu sistem arasında bağlantıyı sağlayan uygulama devresi tanıtılmıştır. Devamında ise, değiştirilmiş hali Şekil 34'te görülen sisteme uygun olarak tanımlanan 4 farklı probleme ait açıklamalar ve problemlerin çözümüne ilişkin çizilen akış diyagramları verilmiştir.

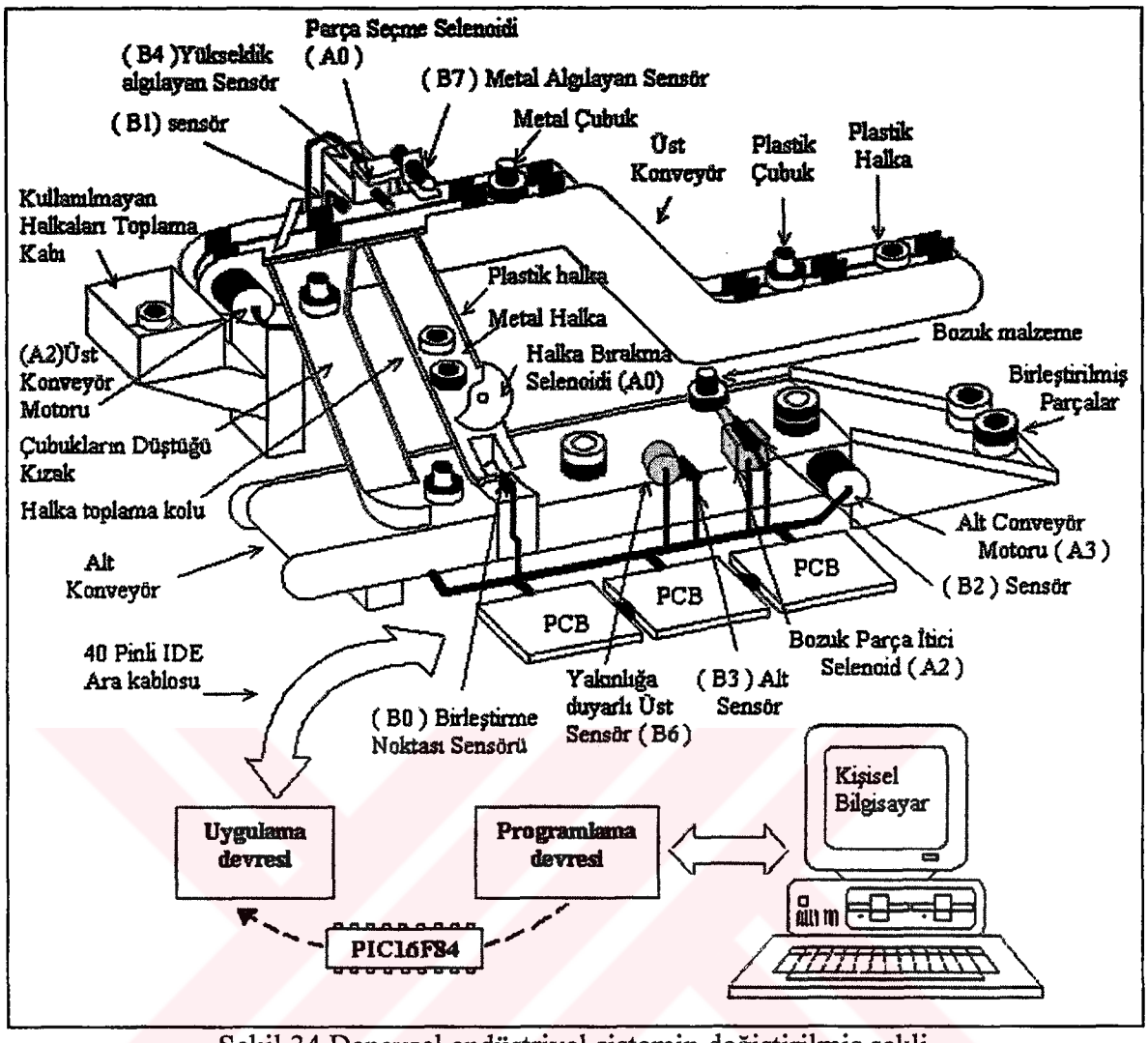
8.1 Endüstriyel Sistemin Çalışması ve Yapılan Değişiklikler

Şekil 34'te değiştirilmiş hali görülen deneysel endüstriyel sistem ile BÖLÜM II.'de anlatıldığı gibi parça tanıma, ayırma ve birleştirme işlemleri gerçekleştirilebilmektedir. Şekil 3'te verilen orijinal sistemde parça seçme selenoidinin biraz önüne metal cisimlere duyarlı bir sensör (B7) yerleştirilmiştir. Bununla birlikte elimizde bulunan metal çubuk ve plastik halkaya ilave olarak plastik çubuk ve metal halka kullanılarak birbirinden farklı parça çeşidi 4'e çıkarılmıştır.

Parça çeşidinin artması ile birlikte birleştirilebilecek malzeme çeşidi (ürün) de artmıştır. Şekil 33'te görüldüğü gibi 4 farklı parça ile dört farklı ürün elde edilebilmektedir.



Şekil 33 Kullanılan ve elde edilebilen malzeme çeşitleri

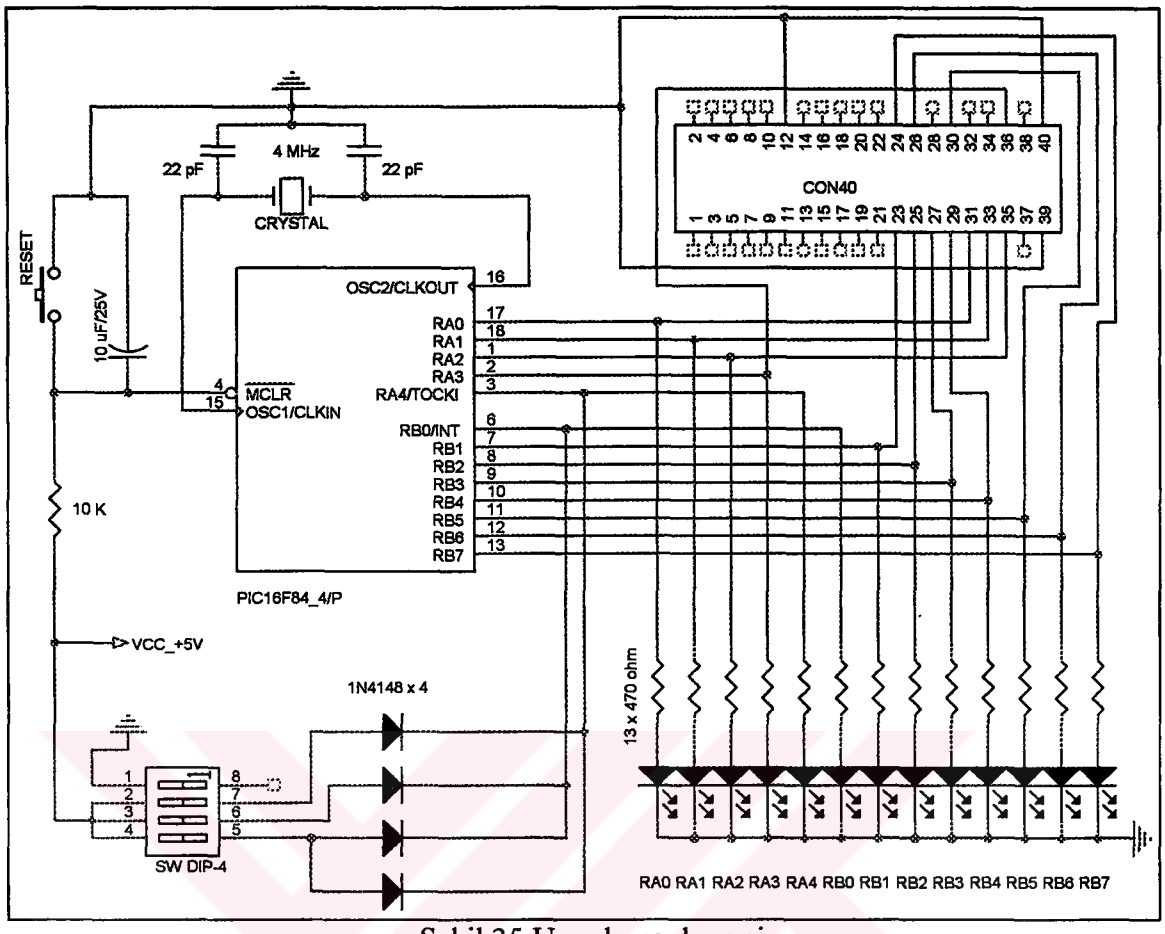


Şekil 34 Deneysel endüstriyel sistemin değiştirilmiş şekli

8.2 Uygulama Devresi

Deneysel endüstriyel kontrol sisteminin PIC mikrodenetleyicisi ile kontrol edilebilmesi için, PIC ile sistem arasındaki bağlantıyı sağlayacak ve PIC'in çalışması için gerekli enerji ile osilatör devresi vb. ihtiyaçları karşılayacak olan Şekil 35'teki devre gerçekleştirilmiştir. Bu devrenin endüstriyel sistem ile bağlantı uçları, sensör ve selenoid isimleri Çizelge 15'te görülmektedir. PROPIC programı ve programlama devresi ile içerisine program yüklenmiş olan PIC16F84 bu uygulama devresine takılarak endüstriyel sistem kontrol edilmiştir.

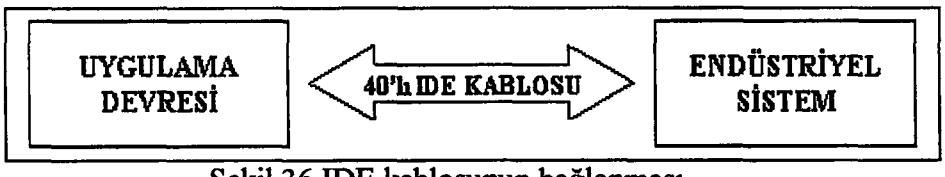
Uygulama devesinde görülen reset butonu PIC'i reset etmek için ve dörtlü dip-switch ise bu tez çalışmasında geliştirilen 4 farklı programdan birisini seçmek için kullanılmıştır. PIC16F84' ün iki girişi (RA4 ve RB0) program seçmek için ayrılarak bu dip-switch'e Şekil 35'te görüldüğü gibi bağlanmıştır. Bu şekilde PIC içerisine yazılan 2'den 4'e multiplexer (çoğullayıcı) mantığıyla çalışan bir program parçası ile 4 farklı programdan birini seçmek mümkün olmaktadır.



Şekil 35 Uygulama devresi

Devrede ayrıca PIC bacaklarına gelen ve giden sinyalleri gözlemlemek için 13 adet 470 ohm'luk ön dirençli LED bağlanmıştır. Sensörlere bağlı olan PIC bacaklarına gelen bilgiler ile parça seçme ve bırakma işini yapan selenoidlere PIC bacaklarından gönderilen bilgiler, 470 ohm'luk dirençlerin üzerine düşen gerilimler sayesinde uygulanmaktadır. Ayrıca 470 ohm'luk dirençler, PIC bacaklarında hiçbir ilgi yok iken, bütün bacaklara sürekli 0 V uygulanmasını sağlamaktadırlar.

40 pinli bir IDE kablosu Şekil 36'da görüldüğü gibi endüstriyel sistem ile uygulama devresi arasındaki bağlantıyı gerçekleştirmek için kullanılmıştır. Deneysel endüstriyel sistem, üzerinde bulunan elektronik kartlardaki, motorlar için sürücü devreleri, optokuplör ve buffer entegreleri gibi elemanlar sayesinde +5V ile kontrol edilebilmektedir. Sistem üzerinde bulunan elektronik kartların devre şemaları EK-B'de verilmiştir.



Şekil 36 IDE kablosunun bağlanması

Çizelge 15 Uygulama devresi bağlantı uçları

PIC16F84 PİN NUMARASI VE AÇIKLAMA		IDE ARA KABLO PİN NUMARASI	DENEYSEL ENDÜSTRİYEL SİSTEM ELEMANLARI	
17 (RA.0)	I/O Pin'i (PORTA)	31	A0	Sınıflandırma Selenoidi
18 (RA.1)	I/O Pin'i (PORTA)	33	A1	Dönel Selenoid
1 (RA.2)	I/O Pin'i (PORTA)	35	A2	Bozuk Parça Ayırıcı Selenoid
2 (RA.3)	I/O Pin'i (PORTA)	37	A3	Zincirli Bant Motoru
2 (RA.3)	I/O Pin'i (PORTA)	38	A4	Kayışlı Bant Motoru
7 (RB.1)	I/O Pin'i (PORTB)	23	B0	Kızılötesi Yansıtıcılı Toplama Bölgesi Sensörü
8 (RB.2)	I/O Pin'i (PORTB)	25	B1	Kızılötesi Yansıtıcılı Sınıflandırma Bölgesi Sensörü
9 (RB.3)	I/O Pin'i (PORTB)	27	B2	Kızılötesi Yansıtıcılı Ayırma Bölgesi Sensörü
10 (RB.4)	I/O Pin'i (PORTB)	29	B3	Kızılötesi Yansıtıcılı Kapasitif Sensör
11 (RB.5)	I/O Pin'i (PORTB)	30	B4	Kızılötesi Yansıtıcılı Metal Algılayıcı Sensör
12 (RB.6)	I/O Pin'i (PORTB)	26	B6	Yakınlığa Duyarlı (Kapasitif) Sensör
13 (RB.7)	I/O Pin'i (PORTB)	24	B7	Metal Algılayıcı (İndüktif) Sensör
5 (Vss)	Besleme ucu (-)	12, 39, 40		GND

8.3 Endüstriyel Sistem Uygulama Problemleri

Problem 1: Deneysel Endüstriyel sistemde metal çubuk ve plastik halkaların birleştirilmesi işlemi gerçekleştirilecek. Ayrıca alt konveyör sonunda birleşmeyen (tek) parçalar varsa bu parçalar hatalı olarak değerlendirilip bozuk malzeme kutusuna atılacaktır. Halka toplama (besleme) kolu en fazla 5 adet halka alabildiği için eğer, bu kolda 5 adet halka bulunuyor ise üst konveyörde gelen halkalar, halka toplama koluna itilmeyecek ve üst konveyör sonunda bulunan, kullanılmayan halkaları toplama kabına taşınacaktır.

Problem 2 : Endüstriyel sistemde üst konveyörde sırası belli olmadan karışık bir şekilde gelen plastik ve metal halkalardan ilk önce metal halka, halka toplama koluna itilecek ve kesintisiz olarak gelen metal halka sayısı kadar plastik halka itilecek. İtilen en az bir metal halkadan sonra plastik halka gelmiş ise, bu gelen plastik halkalardan itilen metal halka sayısı kadar itilmedikçe arada gelen metal halkalar itilmeyecektir. Bu problemde çubukların metal yada plastik olması önemli olmayıp gelen her çubuk halka toplama kolunda halka varsa, bu halkalarla birleştirilecektir. Birinci problemde olduğu gibi halka toplama kolunda en fazla 5 adet halka bulunabilecek ve alt konveyör sonunda hatalı parça varsa ayrılacaktır.

Problem 3 : Üst konveyörden karışık bir biçimde gelen 4 farklı parçadan Şekil 33'te görülen 4 farklı şekilde birleşmeler sağlanacaktır. Plastik halka (Plastic Ring) PR, metal halka (Metallic Ring) MR, plastik çubuk (Plastic Peg) PP, metal çubuk (Metallic Peg) MP ile gösterilirse, 3 adet PRPP, 3 adet MRMP, 2 adet PRMP ve 2 adet MRPP olmak üzere toplam 10 adet birleşme işlemi gerçekleştirilecek ve bu sayılara ulaşıldığında birleşme ve ayıklama işlemleri duracaktır. Bir birleşme için sayı tamamlanmış ise, örneğin MRPP birleşmesi 2 adet üretilmiş ve o anda halka toplama kolunda MR varsa ve PP gelmiş ise bu parçaların birleştirilmesi engellenecektir. Bu problemde halka toplama kolunda bir anda sadece 1 adet halka bulunacak ve gelen çubuklar ile karşılaştırılıp gerekiyorsa birleştirilecektir. Birinci ve ikinci problemlerdeki gibi (hatalı) bozuk parça ayırma işlemi bu problemde de yapılacaktır.

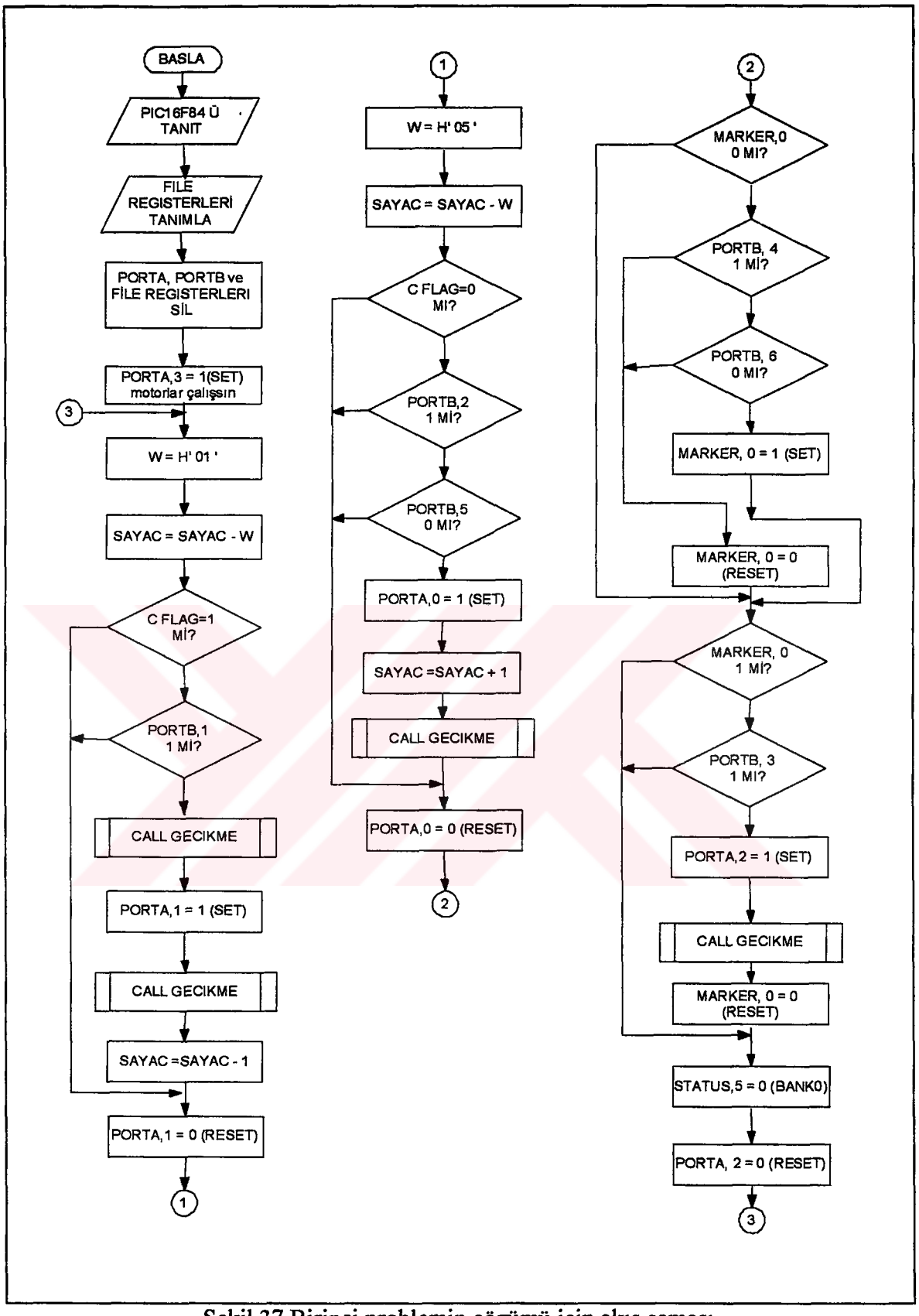
Problem 4 : 3. Problemde verilen sayıda birleşmeler gerçekleştirilecek olmakla birlikte, üretilmek istenen sayıya ulaşıncaya kadar gelen bütün halkalar, (halka toplama kolundaki halka sayısı 5'ten az ise) sırası önemli olmadan halka toplama koluna yerleştirilecek ve gelen her çubuk, bu kolun en başında bulunan halka ile karşılaştırılarak birleştirilecektir. Bu sayede birleşme işlemi, seri ve hızlı bir şekilde gerçekleşecektir. Bozuk parça ayırma işlemi bu problem için de geçerlidir.

8.4 Uygulama Problemlerinin çözümleri

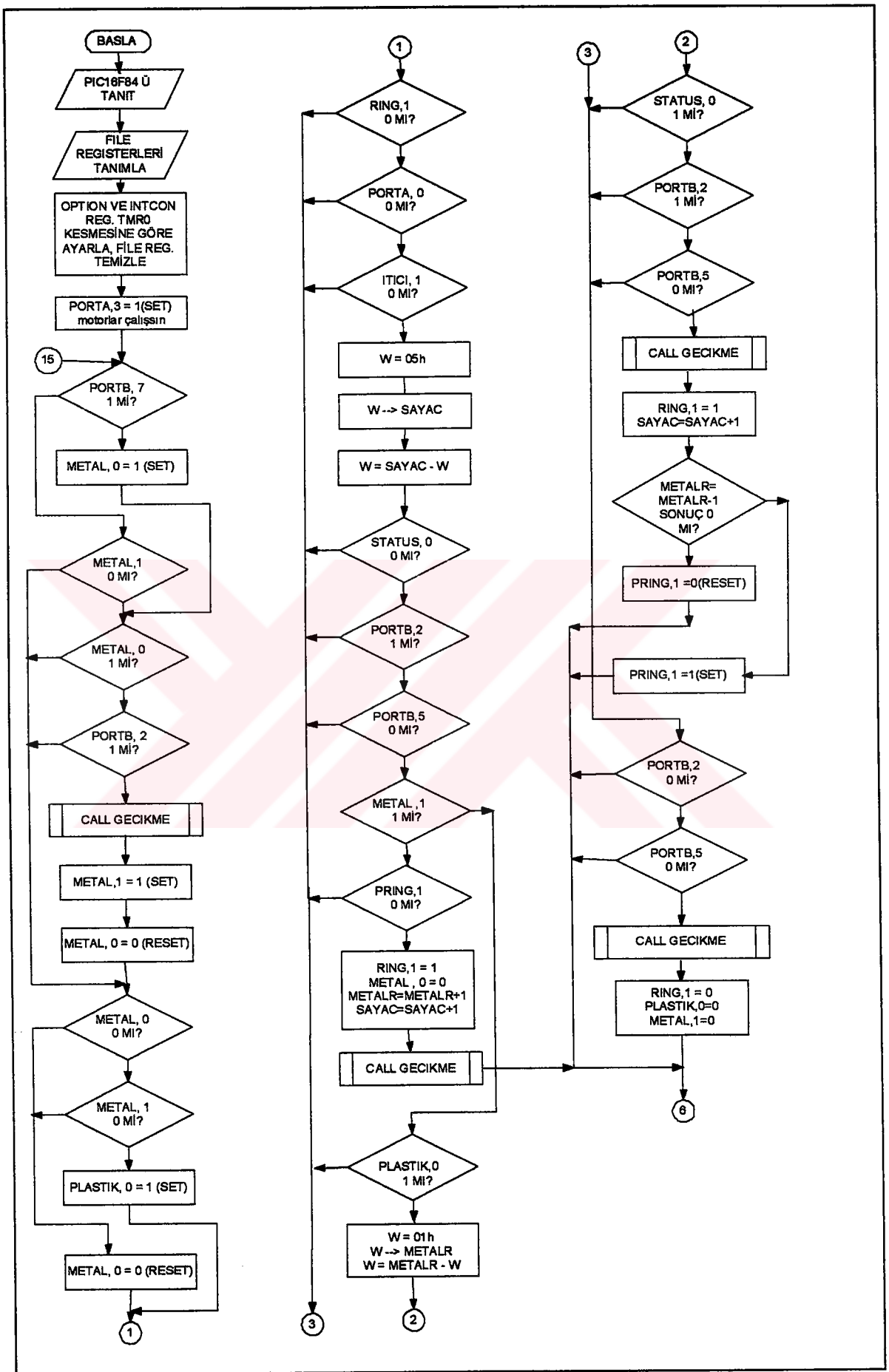
Aşağıda her bir problemin PIC16F84 için PIC assemly dili ile çözümlerine ilişkin yazılan programların akış şemaları verilmiştir. Bu programların ASM uzantılı dosyaları ise EK-A'da verilmiştir.

Buna göre, problem 1 için Şekil 37'de, problem 2 için Şekil 38'de, problem 3 için Şekil 39'da, problem 4 için Şekil 40'ta verilen akış diyagramları çizilmiştir.

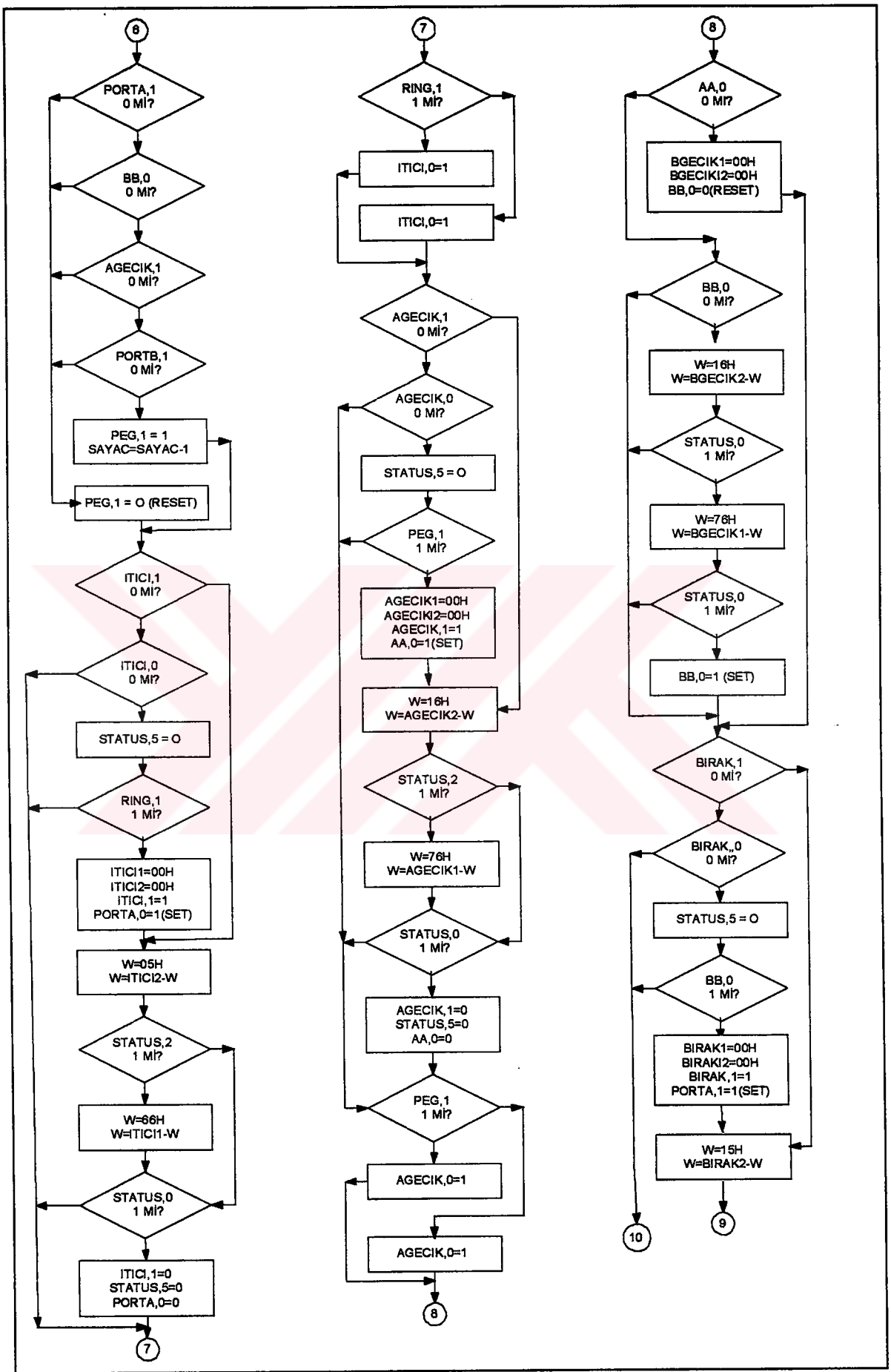
Bütün akış diyagramlarında Şekil 31'de belirtildiği gibi, karşılaştırma (karar) işlemlerinin yapıldığı akış diyagramı sembollerinin yanlarına evet yada hayır cevabı yazılmamış olup, eğer sorulan soruya evet cevabı veriliyor ise aşağıya çizilen ok, hayır cevabı veriliyor ise yanlara çizilen oklar geçerlidir.



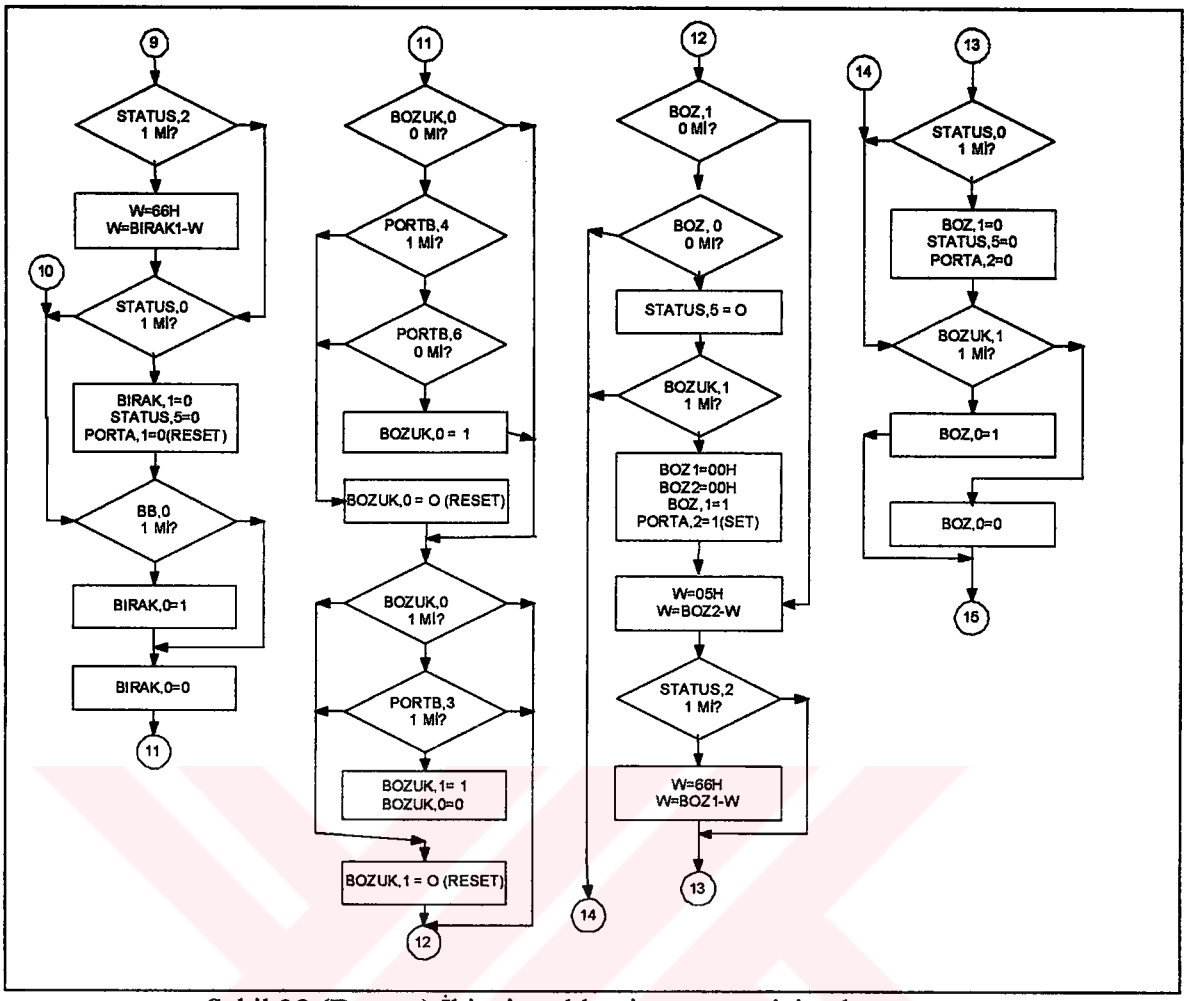
Şekil 37 Birinci problemin çözümü için akış şeması



Şekil 38 İkinci problemin çözümü için akış şeması



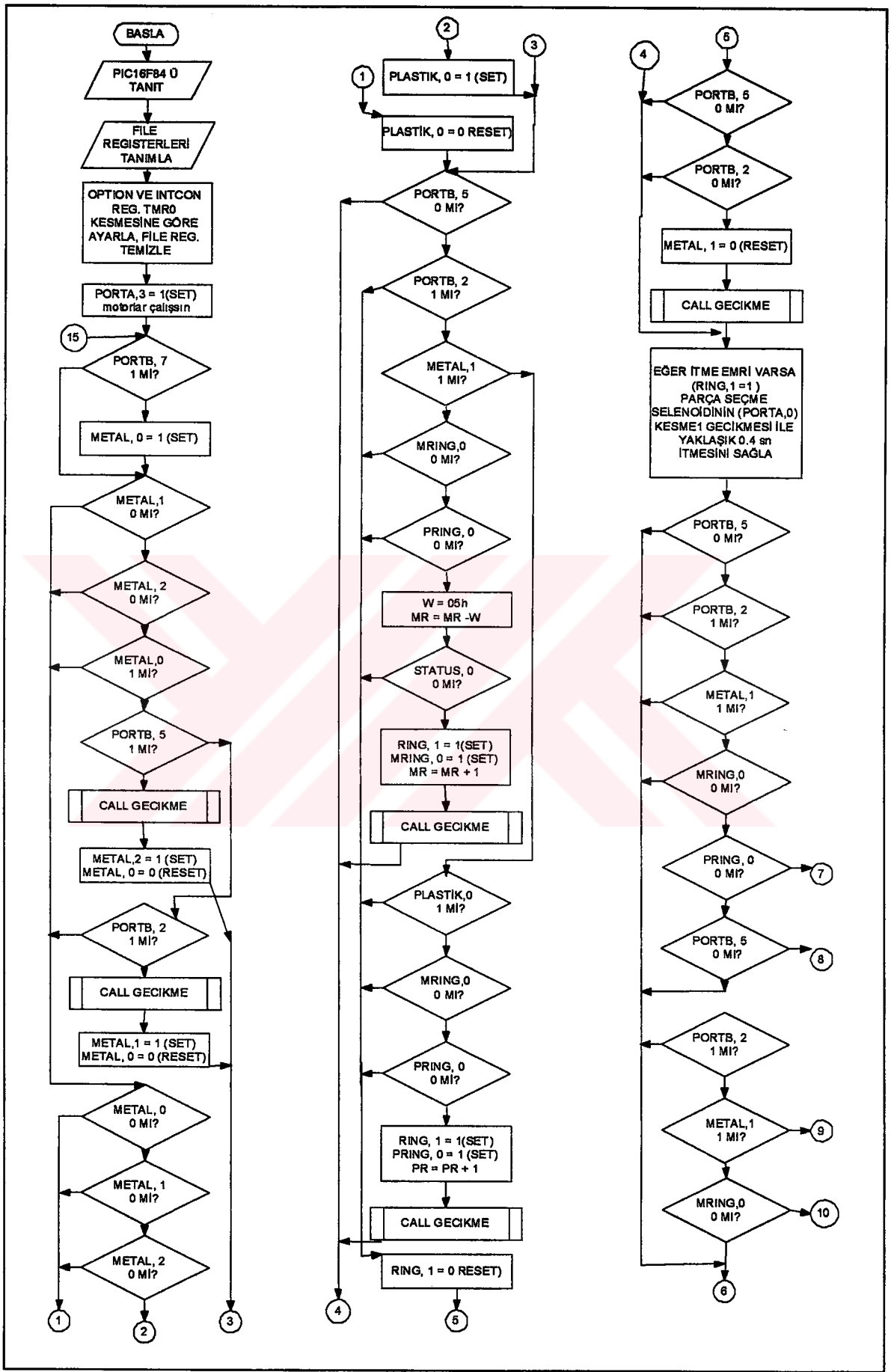
Şekil 38 (Devam) İkinci problemin çözümü için akış şeması



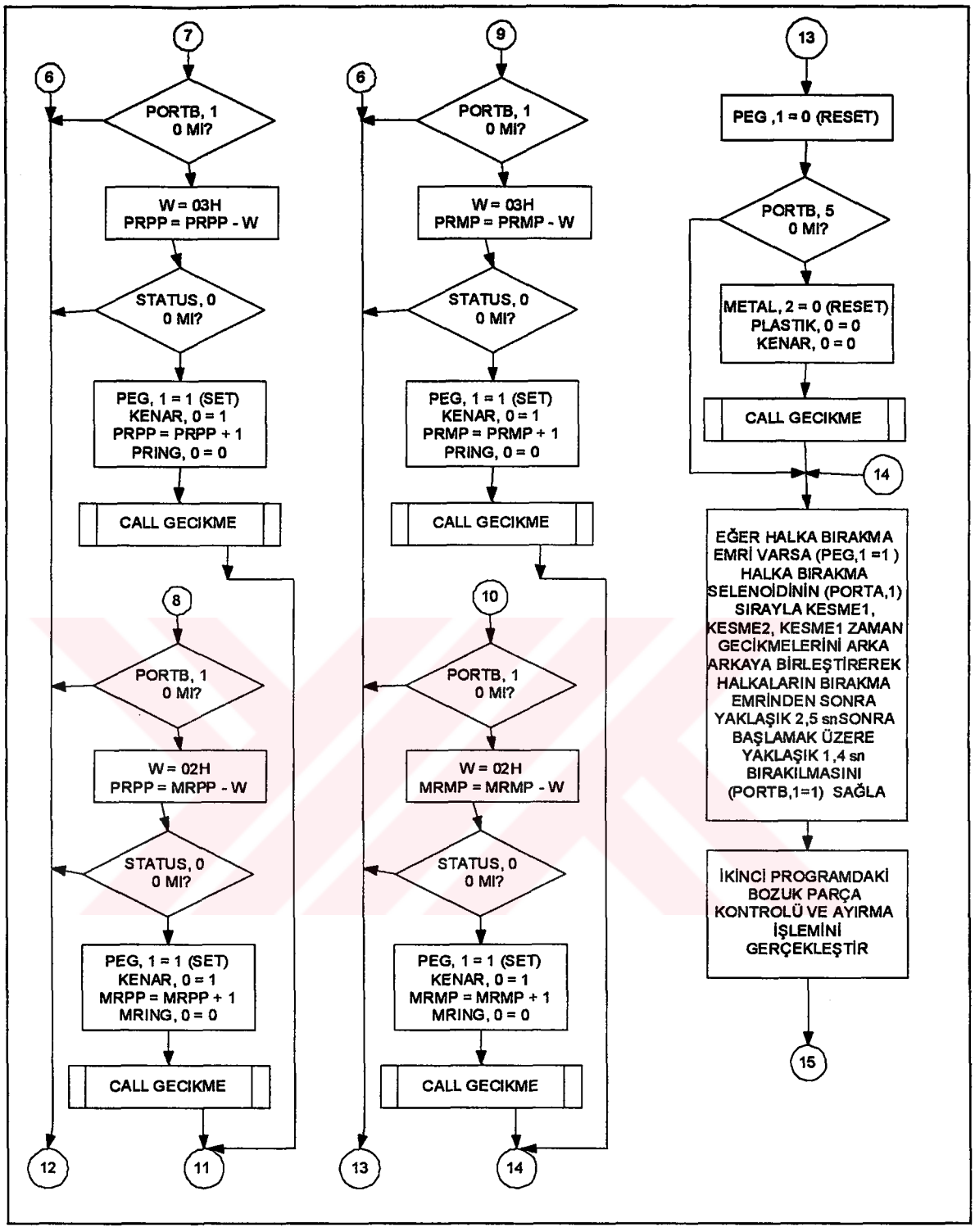
Şekil 38 (Devam) İkinci problemin çözümü için akış şaması

Birinci problemdeki bütün zaman geciktirme işlemleri döngülü zaman geciktirme alt programı sayesinde gerçekleştirilmiştir. Diğer üç problemde ise bütün sensörlerin ve selenoidlerin aynı anda başka bir işlevi gerçekleştirebilmeleri için, TMR0 kesmesi ile sağlanan zaman geciktirmeleri kullanılmıştır. Döngülü zaman geciktirmeleri gecikmesi kullanılsaydı, zaman geciktirme döngüsü bitinceye kadar sistemde başka hiçbir kontrol işlemi gerçekleştirilmemiş olacaktı.

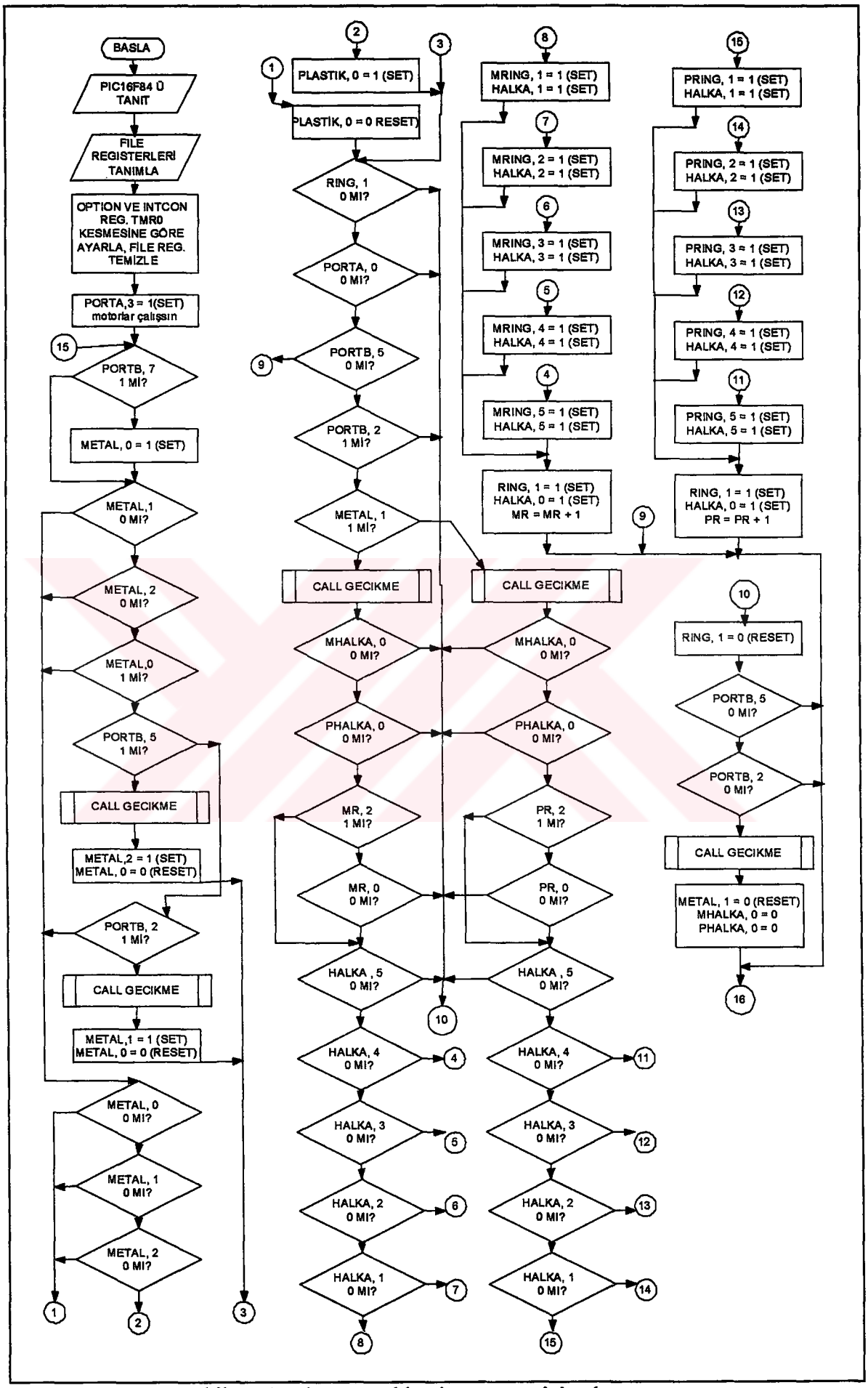
İkinci problemin çözümüne ilişkin akış şemasında kesme gecikmelerine ait program parçaları verilmiş olup, üçüncü ve dördüncü problemlerin akış şemalarında daha kolay anlaşılabilmesi için, sadece hangi kesme gecikmesi kullanıldığı (Bölüm 7.3.'te verilen) ve ne kadar sürelik bir gecikme yaptırdığı yazılacaktır. Bütün programlardaki kesme gecikme alt programları bölüm 7.3.'te verilen mantık ile yazılmıştır.



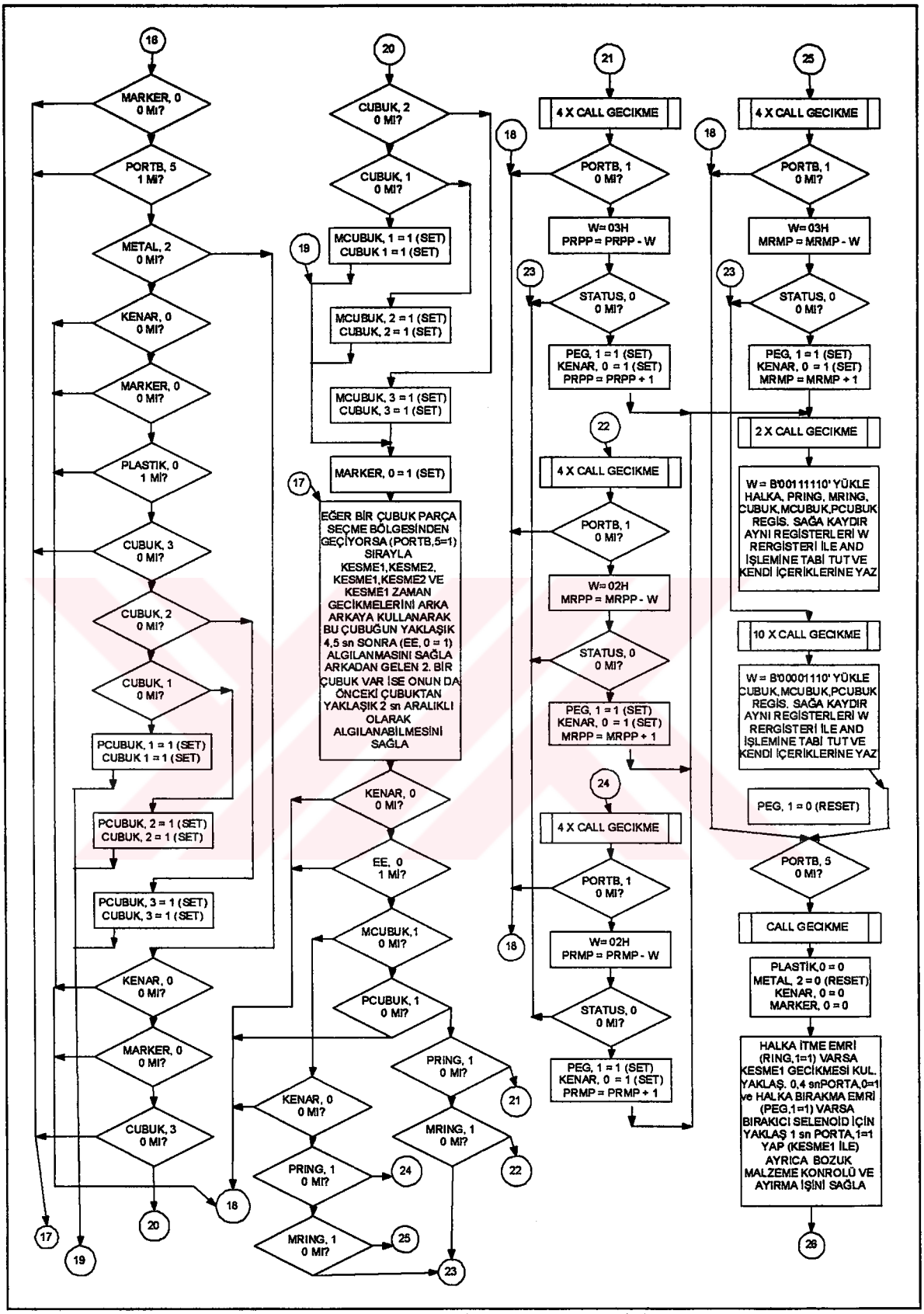
Şekil 39 Üçüncü problemin çözümü için akış şeması



Şekil 39 (Devam) Üçüncü problemin çözümü için akış şeması



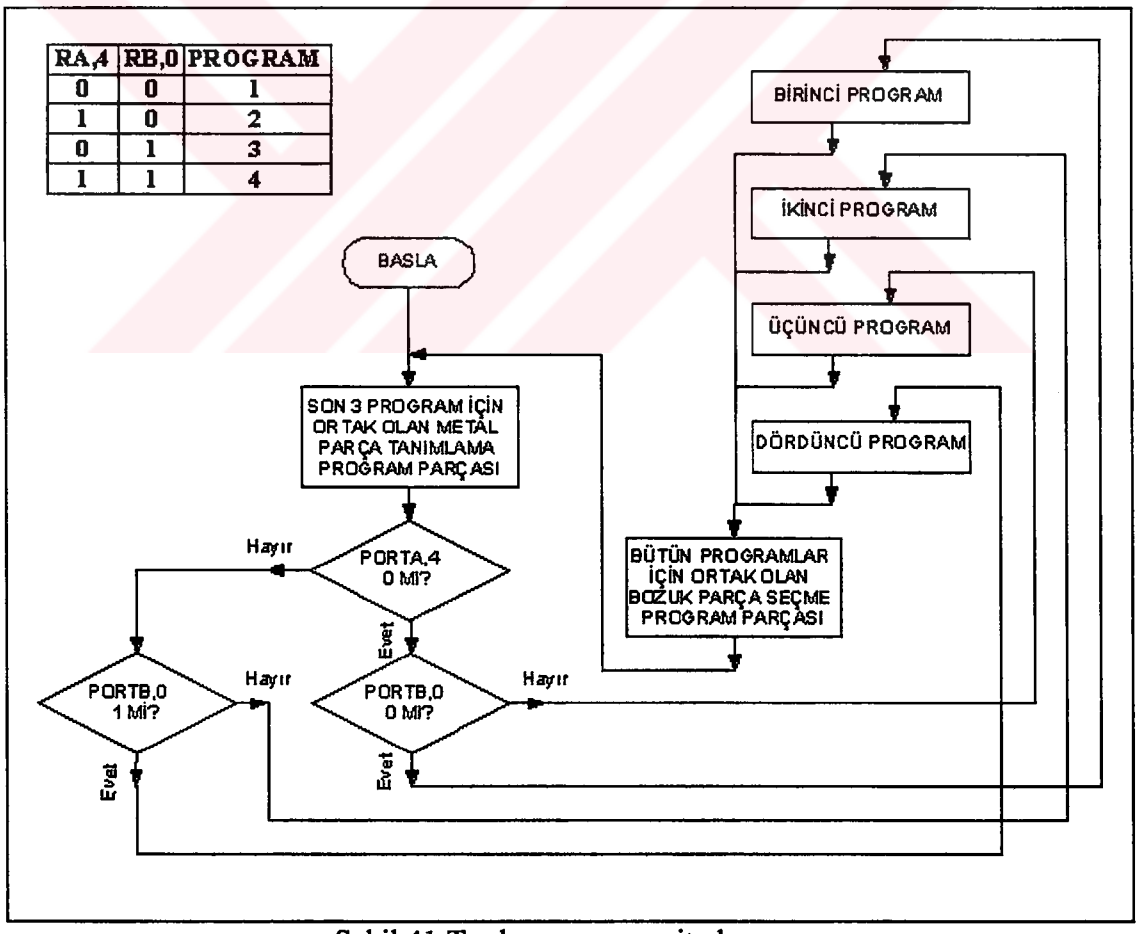
Şekil 40 Dördüncü problemin çözümü için akış şeması



Şekil 40 (Devam) Dördüncü problemin çözümü için akış şeması

8.5 Uygulama Problemlerinin Çözümüne İlişkin Programların Birleştirilmesi

Bölüm 8.3'de verilen problemlerin tek tek çözümlerine ilişkin programlar birleştirilerek tek bir PIC16F84 mikrodenetleyicisine yüklenmiştir. Programların ayrı ayrı elde edilen .HEX dosyalarının toplam büyüklüğü 1 Kbx14 bit olan PIC16F84 program belleğinden çok daha fazla yer tutmaktadır. Bu yüzden, toplu programda her program için aynı veya farklı isimlerle kullanılan file registerlerin çoğu ortak kullanılmıştır. Ayrıca her programda yer alan hatalı (bozuk) malzeme ayıklama program parçası ve bazı kesme gecikmeleri de ortak kullanılarak, toplam program hafızası azaltılmıştır. Bu sayede 1 Kbx14 bit program hafızası olan PIC16F84'e yüklenebilecek büyüklükte bir program elde edilmiştir. Uygulama devresinde verilen dörtlü dip-switch ile her programın ayrı ayrı seçilerek çalıştırılabilmesi için de bir program parçası eklenmiştir. Bu program parçası ve toplu programa ait akış şeması aşağıda verilmiştir. Toplu programın .ASM uzantılı dosyası ise EK-A'da verilmiştir.



Şekil 41 Toplu programa ait akış şeması

BÖLÜM IX

SONUÇ ve ÖNERİLER

Bu çalışmada PIC16F84 mikrodenetleyicisi ile deneysel bir endüstriyel otomasyon sisteminin kontrolü gerçekleştirilmiştir. Kontrol işleminin yapılabilmesi için ilgili bölümlerde ayrıntılı olarak açıklandığı gibi, program geliştirme yazılımı (Ör: MPLAB, MPASM), programlama yazılımı (Ör: PROPIC), programlama devresi ve uygulama devresi gibi bazı donanım ve yazılımların kullanılması gerekmektedir.

PIC mikrodenetleyicileri çok geniş ürün ailesine sahip olması, ucuzluğu, EEPROM belleğe sahip olup kullanım kolaylığını artırması vb. gibi birçok özellikleri sayesinde gün geçtikçe popülerliğini arttırmaktadır.

Bu tez çalışmasında kullanılan PIC16F84 yerine daha fazla özelliklere sahip mikrodenetleyiciler ve PIC ASSEMBLY programlama dili yerine PIC BASIC, PIC C, PIC PASCAL gibi yüksek seviyeli programlama dilleri yada PARSIC, PICBIT gibi paket programları kullanıldığında ise PIC programlarının geliştirilmesi, çok daha rahat ve hızlı bir şekilde gerçekleştirilebilmektedir. Bu programlar çok karmaşık sistemlerin çözümünde yetersiz kalabilmektedirler. Yüksek seviyeli bu programlama dilleri assembly dili komutları ile birlikte kullanılabilmektedir. Bu durumda yüksek seviyeli bir PIC programlama dili ve PIC assembly dili birlikte kullanıldığında bu sorun ortadan kalkmaktadır.

Bu çalışmada yapılmış olan 4 farklı endüstriyel sistem uygulama problemlerinin çözümlerine ilişkin programların PIC içerisine yüklenen .HEX uzantılı program kodlarının büyüklüğü 1. problem için 75x14 bit, 2. problem için 299x14 bit, 3. problem için 365x14 bit, 4. problem için 621x14 bit ve toplam olarak 1360x14 bit tutmaktadır. Programlar toplu olarak tek bir PIC16F84'e yüklenebilmesi için birleştirildiğinde 1024x14 bit olan program hafızasını aşmakta olduğu görülmüştür. Bu yüzden ancak bazı registerlerin ve program parçalarının ortak kullanılması gibi düzenlemelerle bu sorun çözülmüştür. Böylece toplu program için 1012x14 bit büyüklüğüne sahip .HEX uzantılı program kodu elde edilmiş ve PIC16F84 içerisine yüklenebilmiştir. PIC16F84' ten daha fazla program hafızasına sahip

bir PIC mikrodenetleyicisi kullanıldığında hafıza yetersizliğinden kaynaklanan böyle bir sorun ortaya çıkmayacaktır.

PIC16F84 ile daha fazla özelliğe sahip olan örneğin, PIC16F877 mikrodenetleyicisinin özelliklerinin karşılaştırılması Çizelge 16'da verilmiştir. Bu çizelge incelendiğinde PIC16F877 mikrodenetleyicisinin analog/dijital çevirici özelliği, giriş/çıkış port sayısının fazla olması (33 adet) vb. gibi birçok özelliği ile PIC16F84'ten daha üstün yeteneklere sahip olduğu görülür.

Çizelge 16 PIC16F84 ve PIC16F877 mikrodenetleyicilerinin karşılaştırılması

ÖZELLİKLER	**PIC16F877**	**PIC16F84**
Çalışma hızı (Maksimum)	DC-20Mhz	DC-10 Mhz
Program Belleği	8Kx14 bit Flash ROM	1Kx14 bit Flash ROM
EEPROM Belleği	256 byte	64 byte
Kullanıcı RAM	368 x 8 byte	68 x 8 byte
Giriş/Çıkış port sayısı	33	13
Timer	Timer0, Timer1, Timer2	Timer0
A/D çevirici	8 kanal 10 bit	YOK
Capture/Comp/PWM	16 bit Capture 16 bit Compare 10 bit PWM çözünürlük	YOK
Seri çevresel arayüz	SPI (Master) ve 12C (Master / Slave) modunda SPI portu (senkron seri port)	YOK
Paralel slave port	8 bit, harici RD,WR ve CS kontrollu	YOK
USART/SCI	9 bit adresli	YOK

PIC16F84'e göre üstün özellikleri verilen PIC16F877 mikrodenetleyicisi dışında, PIC ailesi içerisinde daha farklı özelliklere sahip mikrodenetleyiciler de bulunmaktadır. PIC mikrodenetleyicilerinin çok geniş bir ürün ailesine sahip olması ile birlikte, çok daha fazla program hafızası ve giriş/çıkış portu, analog/dijital çevirici, vb. gibi özellikler gerektiren uygulamalar için amaca uygun olarak seçilebilecek, bir PIC mikrodenetleyicisi bulunabilmektedir.

Sonuç olarak, bu çalışmada da görüldüğü gibi bir PIC mikrodenetleyicisinin endüstriyel sistem kontrolü amacıyla kullanılması sayesinde, sistem tasarımının kolaylığının yanı sıra, özellikle maliyeti azaltıcı bir etkisinin olması dikkat çekmektedir. Bu yönleriyle PIC mikrodenetleyicili endüstriyel sistem uygulamaları, klasik kumanda sistemleri, PLC kontrollü sistemler ve mikro işlemci tabanlı kontrol sistemlerine alternatif olabilecek bir nitelik taşımaktadır.

KAYNAKLAR

Altınbaşak, O., 2000, *Mikrodenetleyiciler ve Pic Programlama*, Altaş Yayınevi, İstanbul.

Çelik, H.H., 1997, *Mikroişlemcilerin Yapısını ve Devrelerini Kavrama*, M.Ü.T.E.F. Ders notları, İstanbul.

Dinçer. G., 2000, *PIC Microcontroller Uygulama Devreleri*, İnfogate Yayınları,

Katzen. S., 2001, *The Quintessential Pic Microcontroller*, Springer-Verlag Publications, ISBN: 185233309X

Matic. N. and Andric. D., 2000, *The Pic Microcontroller Book 1*, Microelectronica Publications.

Microchip., 2000, *Microchip Tecnical Library CD-ROM (Second* Edition), Microchip Tecnology Inc., Arizona, Disc 1 of 2, 30430c.pdf ve 35007a.pdf dosyaları, Arizona.

Özdemir, A. 2000, *Endüstriyel Elektronik*, Kozan Ofset Matbaa, Ankara.

Özkan. T., 1995, *Mikroişlemciler Mikrobilgisayarlar Assembly dili ve Uygulamaları*, 2E Yayınları, İstanbul,.

Predko. M., 1998, *Handbook of Microcontrollers*, McGraw-Hill Companies

Predko. M., 2000, *Programing and Customizing PIC Microcontrollers*, McGraw-Hill Companies, ISBN: 0071361723

Turgutlu. H. F.,2001, *PICBIT Programı ile Lojik Tasarım*, Yüksek Lisans Semineri, Niğde Üniversitesi, Niğde,44s.

Yağımlı, M. ve Akar, F. 1998, *Dijital Elektronik*, Beta Yayınevi, İstanbul

Yalçın C., 1998, *PIC Programlama El Kitabı(Çeviri)*, Dinçer. G, Era-İnfogate Yayınları,

Yazır, T., 1998, *PLC İleri Kumanda Teknikleri*, Bursa.

İNTERNET SİTELERİ

1. http://www.microchip.com/
2. http://members1.chello.nl/~f.vdburgh/picbit/index.htm
3. http://www.antrak.org.tr/gazete/
4. http://www.jdm.homepage.dk/newpic.htm
5. http://www.pp.clinet.fi/~newmedia/pic/index.html
6. http://www.dontronics.com/piclinks.html

7. http://members.optushome.com.au/donmck/dtait/piclinks.html
8. http://www.lpilsley.co.uk/software.htm
9. http://www.techfreakz.org/oldstuff/picb.html
10. http://www.ibfriedrich.com/
11. http://www.technick.net/
12. http://come.to/thepicarchive
13. http://www.bubblesoftonline.com
14. http://www.labcenter.co.uk
15. http://www.vmdesign.com
16. http://www.basicmicro.com
17. http://www4.tpg.com.au
18. http://www.melabs.com
19. http://www.letbasic.com
20. http://www.parsic.de
21. http://www.picsystems.org
22. http://www.rentron.com
23. http://sol.spaceports.com/~engels/programming/pic/index.html
24. http://www.to-pic.com
25. http://www.parallaxinc.com
26. http://downloads.cambs.net/itmidx1.htm
27. http://www.vitrum.cz/snail/icd.htm
28. http://www.sprut.de/electronic/pic/projekte/brenner5/index.htm
29. http://www.devrs.com/pic
30. http://www.myke.com
31. http://www.brouhaha.com/~eric/pic
32. http://www.home.netcom.com/~fastfwd/answers.html

EKLER

EK-A Deneysel Endüstriyel Sistem Uygulama Problemlerinin Çözümüne İlişkin PIC16F84 İçin Hazırlanmış .ASM Dosyaları

EK-A.1 Birinci Program

```
        LIST P=16F84A                   ;16F84A YI TANIT
        INCLUDE "P16F84A.INC"           ;16F84A.INC DOSYASINI EKLE
SAYAC   EQU     H'0C'                   ;SAYAC REGİSTERİNİ VE ADRESİNİ TANIMLA
SAY     EQU     H'0D'                   ;SAY      "       "     "       "
SAY1    EQU     H'0E'                   ;SAY1     "       "     "       "
SAY2    EQU     H'0F'                   ;SAY2     "       "     "       "
BOZUK   EQU     H'13'                   ;BOZUK    "       "     "       "
        CLRF    PORTA                   ;PORTA YI SİL
        BSF     STATUS,5                ;BANK1 E GEÇ
        MOVLW H'FF'                     ;W REGİSTERİNE H'FF' SAYISINI YÜKLE
        MOVWF TRISB                     ;W REGİSTERİ--> (H'FF') TRİSB YE YÜKLE PORTB'Yİ GİRİŞ YAP
        MOVLW B'00010000'               ;W <-- B'00010000'
        MOVWF TRISA                     ;W --> TRISA (PORTA YI 5. BİT HARİÇ ÇIKIŞ YAP)
        BCF     STATUS,5                ;BANK0'A GEÇ
        CLRF    SAYAC                   ;SAYAC REGİSTERİNİ TEMİZLE
        CLRF    BOZUK                   ;BOZUK REGİSTERİNİ TEMİZLE
        BSF     PORTA,3                 ;PORTA 3. BİTİNİ 1 YAP ( MOTORLAR ÇALIŞSIN )
BASLA
        MOVLW H'01'                     ;W = H'01'
        SUBWF   SAYAC,0                 ;SAYAC = SAYAC-W
        BTFSS   STATUS,0                ;CF=1 Mİ? (SAYAC>=1 Mİ)
        GOTO    BIRAKMA                 ;HAYIR(TOPLAMA BÖLGESİ BOŞ)İSE
PORTB_KONTROL1
        BTFSC   PORTB,1                 ;PORTB 1.BİT 0 MI?
        GOTO    BIRAKMA                 ;HAYIR İSE
BIRAK
        CALL    GECIKME                 ;BIRAKMADAN ÖNCE BEKLE
        BSF     PORTA,1                 ;EVET İSE, PORTA 1.BİTİNİ 1 YAP (BIRAK)
        CALL    GECIKME                 ;BIRAKTIKTAN SONRA BEKLE
        DECF    SAYAC,F                 ;SAYAC = SAYAC-1 (SAYACI BİR AZALT)
        GOTO    KONTROL1_CIKIS          ;KONTROL1_CIKIS ETİKETİNE GİT
BIRAKMA
        BCF     PORTA,1                 ;PORTA,1 = 0 YAP (BIRAKMA)
KONTROL1_CIKIS
;*****************************
        MOVLW H'05'                     ;W = H'05'
        SUBWF   SAYAC,0                 ;SAYAC = SAYAC-W
        BTFSC   STATUS,0                ;CF=0 MI? (SAYAC<5 Mİ?)
        GOTO    ITME                    ;HAYIR(TOPLAMA BÖLGESİNDE 5 HALKA VAR) İSE
PORTB_KONTROL2
        BTFSS   PORTB,2                 ;PORTB 2.BİT 1(AYIRMA BÖLGESİNDE PARÇA VAR) Mİ?
        GOTO    ITME                    ;HAYIR İSE
PORTB_KONTROL3
        BTFSC   PORTB,5                 ;PORTB 5.BİT 0 (BU PARÇA HALKA MI) MI?
        GOTO    ITME                    ;HAYIR İSE
IT
        BSF     PORTA,0                 ;EVET İSE, PORTB 0. BİTİNİ 1 YAP (İT)
        INCF    SAYAC,F                 ;SAYAC = SAYAC+1(SAYACI BİR ARTIR)
        CALL    GECIKME                 ;İTTİKTEN SONRA BEKLE
        GOTO    KONTROL2_CIKIS          ;KONTROL2_CIKIS ETİKETİNE GİT
ITME
        BCF     PORTA,0                 ;PORTA,0 = 0 YAP (İTME)
```

```
KONTROL2_CIKIS
;*******************************
;
BOZUK_KONTROL1
        BTFSC   BOZUK,0             ;BOZUK,0. BİT 0 (BOZUK PARÇA YOK) MI?
        GOTO    BOZUK_KONTROL4      ;HAYIR (BOZUK,0=1) İSE HAFIZADA TUT
                                    ;EVET (BOZUK,0=0 İSE) TEKRAR KONTROL ETMEYE BASLA
        BTFSS   PORTB,4             ;PORTB,4 = 1 Mİ? (BOZUK BÖLGE ALT SENSÖR)
        GOTO    BOZUK_ITME          ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL2
        BTFSC   PORTB,6             ;PORTB,6 = 0 MI? (BOZUK BÖLGE ÜST SENSÖR)
        GOTO    BOZUK_ITME          ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL3
        BSF     BOZUK,0             ;BOZUK,0 = 1 YAP (BOZUK PARÇAYI HAFIZAYA AL)
        GOTO    BOZUK_KONTROL4      ;BOZUK_KONTROL4 ETİKETİNE GİT
BOZUK_ITME
        BCF     BOZUK,0             ;HAFIZAYI SİL
;*******************************
BOZUK_KONTROL4
        BTFSS   BOZUK,0             ;BOZUK PARÇA ALGILANIŞ MI?
        GOTO    BOZUK_KONTROL7      ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL5
        BTFSS   PORTB,3             ;BOZUK PARÇA AYIRICI SELENOİD YANINDA MI?
        GOTO    BOZUK_KONTROL7      ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL6
        BSF     PORTA,2             ;BOZUK PARÇAYI İT
        CALL    GECIKME             ;İTTİKTEN SONRA BİRAZ BEKLE
        BCF     BOZUK,0             ;HAFIZAYI SİL
        GOTO    BOZUK_CIKIS         ;BOZUK_CIKIS ETİKETİNE GİT
BOZUK_KONTROL7
        BCF     STATUS,RP0          ;BANK0 A GEÇ
        BCF     PORTA,2             ;BOZUK PARÇA AYIRICI SELENOİDİ SIFIRLA
BOZUK_CIKIS
;*******************************
;
        GOTO    BASLA               ;BASLA ETİKETİNE GİT (PROGRAM BAŞINA)
;*******************************
;
GECIKME
        MOVLW   H'04'               ;ÜÇLÜ DÖNGÜ İLE ZAMAN GECIKMESİ
        MOVWF   SAY                 ;ALT PROGRAMI
DONGU                               ;       "
        MOVLW   H'FF'               ;       "
        MOVWF   SAY1                ;       "
DONGU1                              ;       "
        MOVLW   H'FF'               ;       "
        MOVWF   SAY2                ;       "
DONGU2                              ;       "
        DECFSZ  SAY2,F              ;       "
        GOTO    DONGU2              ;       "
        DECFSZ  SAY1,F              ;       "
        GOTO    DONGU1              ;       "
        DECFSZ  SAY                 ;       "
        GOTO    DONGU               ;       "
        RETURN                      ;       "
        END                         ;PROGRAM SONU
```

EK-A.2 İkinci Program

```
        LIST    P=16F84A                ;16F84A YI TANIT
        INCLUDE "P16F84A.INC"           ;16F84A.INC DOSYASINI EKLE
        ORG     H'00'                   ;PROGRAM BAŞLANGIC ADRESİ H'00'
        GOTO    BASLA                   ;BASLA ETİKETİNE GİT
        ORG     H'04'                   ;İNTERRUPT(KESME SİNYALİ GELİRSE)
        GOTO    INT_ALT_PROG            ;INT_ALT_PROG ETİKETİNE GİT
BASLA
SAYAC           EQU     H'0C'           ;SAYAC REGİSTERİNİ VE ADRESİNİ TANIMLA
SAY1            EQU     H'0E'           ;SAY1     "       "       "       "
SAY2            EQU     H'0F'           ;SAY2     "       "       "       "
MARKER          EQU     H'10'           ;MARKER   "       "       "       "
METAL           EQU     H'11'           ;METAL    "       "       "       "
PLASTIK         EQU     H'12'           ;PLASTIK  "       "       "       "
METALR          EQU     H'13'           ;METALR   "       "       "       "
ITICI           EQU     H'14'           ;ITICI    "       "       "       "
ITICI1          EQU     H'15'           ;ITICI1   "       "       "       "
ITICI2          EQU     H'16'           ;ITICI2   "       "       "       "
SAKLA_W         EQU     H'17'           ;SAKLA_W  "       "       "       "
SAKLA_S         EQU     H'18'           ;SAKLA_S  "       "       "       "
RING            EQU     H'19'           ;RING     "       "       "       "
BIRAK           EQU     H'1A'           ;BIRAK    "       "       "       "
BIRAK1          EQU     H'1B'           ;BIRAK1   "       "       "       "
BIRAK2          EQU     H'1C'           ;BIRAK2   "       "       "       "
PEG             EQU     H'1D'           ;PEG      "       "       "       "
BOZ             EQU     H'1E'           ;BOZ      "       "       "       "
BOZ1            EQU     H'1F'           ;BOZ1     "       "       "       "
BOZ2            EQU     H'20'           ;BOZ2     "       "       "       "
BOZUK           EQU     H'21'           ;BOZUK    "       "       "       "
BGECIK          EQU     H'22'           ;BGECIK   "       "       "       "
BGECIK1         EQU     H'23'           ;BGECIK1  "       "       "       "
BGECIK2         EQU     H'24'           ;BGECIK2  "       "       "       "
BB              EQU     H'25'           ;BB       "       "       "       "
AGECIK          EQU     H'26'           ;AGECIK   "       "       "       "
AGECIK1         EQU     H'27'           ;AGECIK1  "       "       "       "
AGECIK2         EQU     H'28'           ;AGECIK2  "       "       "       "
AA              EQU     H'29'           ;AA       "       "       "       "
PRING           EQU     H'30'           ;PRING    "       "       "       "
        CLRF    PORTB                   ;PORTB Yİ SIFIRLA
        CLRF    PORTA                   ;PORTAYI SIFIRLA
        BSF     STATUS,5                ;BANK1 E GEÇ
        MOVLW B'11111111'               ;W REGİSTERİNE H'FF' SAYISINI YÜKLE
        MOVWF TRISB                     ;W REGİSTERİ--> TRİSB YE YÜKLE(PORTB Yİ GİRİŞ YAP)
        MOVLW B'00010000'               ;W <-- B'00010000'
        MOVWF TRISA                     ;W --> TRISA (PORTA YI 5. BİT HARİÇ ÇIKIŞ YAP)
        CLRWDT                          ;WDT Yİ SİL
        MOVLW  B'10001111'              ;W <-- B'10001111'
        MOVWF  OPTION_REG               ;W --> OPTION_REG
        BCF     STATUS,5                ;BANK0 A GEÇ
        MOVLW  H'00'                    ;W <-- H'00'
        MOVWF  TMR0                     ;W --> TMR0
        MOVLW  B'10100000'              ;W <-- B'10100000'
        MOVWF  INTCON                   ;W --> INTCON
        CLRF    SAYAC                   ;SAYAC REGİSTERİNİ SIFIRLA
        CLRF    MARKER                  ;MARKER     "
        CLRF    METAL                   ;METAL      "
        CLRF    PLASTIK                 ;PLASTIK    "
        CLRF    METALR                  ;METALR     "
        CLRF    PEG                     ;PEG        "
        CLRF    RING                    ;RING       "
        CLRF    BB                      ;BB         "
        CLRF    AA                      ;AA         "
        CLRF    BIRAK                   ;BIRAK      "
        CLRF    PRING                   ;PRING      "
        BSF     PORTA,3                 ;PORTA 3. BİTİNİ 1 YAP ( MOTORLAR ÇALIŞSIN )
START
;*********************************
```

```
METAL_TANIMLA
        BTFSS   PORTB,7             ;METAL PARÇA VAR MI?
        GOTO    METALSIL            ;METAL PARÇA YADA HİÇBİR PARÇA YOKSA
        BSF     METAL,0             ;METAL REG. 0. BİTİNİ 1 YAP
        GOTO    METAL_SET           ;METAL_SET ETİKETİNE GİT
METALSIL                            ;
        BTFSC   METAL,1             ;METAL REGISTERI NIN 1.BİTİ 0 MI?
        GOTO    METAL_TAN_CIK       ;METAL REGISTERI NIN 1.BİTİ 1 İSE
METAL_SET                           ;
        BTFSS   METAL,0             ;METAL REGISTERI NIN 0.BİTİ 1 MI?
        GOTO    METAL_TAN_CIK       ;METAL REGISTERI NIN 0.BİTİ 0 İSE
        BTFSS   PORTB,2             ;METAL PARÇA AYIRICI BÖLGEYE GELDİ Mİ?
        GOTO    METAL_TAN_CIK       ;HAYIR
        CALL    GECIKME             ;BİRAZ BEKLE
        BSF     METAL,1             ;METAL PARÇAYI SET ET
        BCF     METAL,0             ;METAL,0 = 0 (ARKADAN GELENLER ALGILANABİLSİN)
        GOTO    METAL_TAN_CIK       ;
METAL_TAN_CIK                       ;
        BTFSC   METAL,0             ;METAL REGISTERI NIN 0.BİTİ 1 Mİ?
        GOTO    PLASTIK_TAN_CIK     ;HAYIR
        BTFSC   METAL,1             ;METAL REGISTERI NIN 1.BİTİ 0 MI?
        GOTO    PLASTIK_TAN_CIK     ;HAYIR
PLASTIK_TANIMLA                     ;
        BSF     PLASTIK,0           ;PLASTIK,0 = 1
        GOTO    ITICI_KONTROL       ;ITICI_KONTROL ETİKETİNE GİT
PLASTIK_TAN_CIK                     ;
        BCF     PLASTIK,0           ;PLASTIK,0 = 0
;*********************************
ITICI_KONTROL                       ;
        BTFSC   RING,1              ;RING,1 = 0 MI?
        GOTO    ITME                ;HAYIR
        BTFSC   PORTA,0             ;PORTA,0 = 0 MI?
        GOTO    ITME                ;HAYIR
        BTFSC   ITICI,1             ;ITICI,1 = 0 MI?
        GOTO    ITME                ;HAYIR
        MOVLW H'05'                 ;W <-- H'05'
        SUBWF SAYAC,W               ;SAYAC = SAYAC-W
        BTFSC   STATUS,0            ;CF=0 MI? (SAYAC<5 MI?)
        GOTO    ITME                ;TOPLAMA BÖLGESİ DOLU İSE
MR_IT                               ;
        BTFSS   PORTB,2             ;AYIRICI BÖLGEYE PARÇA GELDİ Mİ?
        GOTO    ITME                ;HAYIR
        BTFSC   PORTB,5             ;EVET İSE, BU PARÇA HALKA MI?
        GOTO    ITME                ;HAYIR
        BTFSS   METAL,1             ;HALKA METAL Mİ?
        GOTO    PR_IT               ;METAL DEĞİL İSE PR_IT ETİKETİNE GİT
        BTFSC   PRING,1             ;PRING,1 = 0 MI?
        GOTO    ITME                ;HAYIR
MRING_SET                           ;
        BSF     RING,1              ;RING,1 = 1
        BCF     METAL,0             ;METAL,0 = 0
        INCF    METALR,1            ;METALR = METALR + 1
        INCF    SAYAC,1             ;SAYAC = SAYAC + 1
        CALL    GECIKME             ;BİRAZ BEKLE
        GOTO    ITME_CIKIS          ;
;*********************************
PR_IT                               ;
        BTFSS   PLASTIK,0           ;PLASTIK,0 = 1 Mİ? (HALKA PLASTİK Mİ?)
        GOTO    ITME                ;HAYIR
        MOVLW H'01'                 ;W <-- H'00'
        SUBWF METALR,W              ;SAYAC = METALR - W
        BTFSS   STATUS,0            ;ZF=0 MI? (METALR = 0 MI?)
        GOTO    ITME                ;METALR = 0 İSE
        BTFSS   PORTB,2             ;METALR = 0 DEĞİLSE
        GOTO    ITME                ;ITME ETİKETİNE GİT
        BTFSC   PORTB,5             ;PARÇA HALKA MI?
        GOTO    ITME                ;ÇUBUK İSE
PRING_SET                           ;
        CALL    GECIKME             ;İTMEDEN ÖNCE BEKLE
```

```
        BSF     RING,1              ;HALKA İSE, PORTB 0. BİTİNİ 1 YAP (İT)
        INCF    SAYAC,1             ;SAYAC = SAYAC+1
        DECFSZ METALR,1             ;METALR=METALR-1 , METALR=0 MI?
        GOTO    SET_PRING           ;METALR=0 DEĞİL İSE SET_PRING ETİKETİNE GİT
RESET_PRING                         ;
        BCF     PRING,1             ;PRING,1 = 0 YAP
        GOTO    ITME_CIKIS          ;
SET_PRING                           ;
        BSF     PRING,1             ; PRING,1 = 1 YAP
        GOTO    ITME_CIKIS          ;
;*********************************
;
ITME                                ;
        BTFSC   PORTB,2             ;PORTB,2 = 0 MI?
        GOTO    ITME_CIKIS          ;HAYIR
        BTFSC   PORTB,5             ;PORTB,5 = 0 MI?
        GOTO    ITME_CIKIS          ;HAYIR
        CALL    GECIKME             ;BİRAZ BEKLE
        BCF     METAL,1             ;METAL,1 = 0
        BCF     PLASTIK,0           ;PLASTİK,0 = 0
        BCF     RING,1              ;RING,1 = 0
ITME_CIKIS                          ;
;*********************************
BIRAK_KONTROL                       ;
        BTFSC   PORTA,1             ;BIRAKICI SELENOİD ÇALIŞIYORSA
        GOTO    RING_BIRAKMA        ;
        BTFSC   BB,0                ;BIRAKMA EMRİ VERİLMİŞ İSE
        GOTO    RING_BIRAKMA        ;
        BTFSC   AGECIK,1            ;  "              "
        GOTO    RING_BIRAKMA        ;RING_BIRAKMA ETİKETİNE GİT
        MOVLW H'01'                 ;W <-- H'01'
        SUBWF   SAYAC,W             ;SAYAC = SAYAC-W
        BTFSS   STATUS,0            ;CF=1 Mİ? (SAYAC>=1 Mİ)
        GOTO    RING_BIRAKMA        ;HAYIR İSE
PORTB_KONTROL1                      ;
        BTFSC   PORTB,1             ;PORTB 1.BİT 0 MI?
        GOTO    RING_BIRAKMA        ;HAYIR İSE
RING_BIRAK                          ;
        BSF     PEG,1               ;EVET İSE, PEG,1 = 1 YAP (BRAKMA EMRİ VER)
        DECF    SAYAC,F             ;SAYAC = SAYAC-1
        GOTO    KONTROL1_CIKIS      ;
RING_BIRAKMA                        ;
        BCF     PEG,1               ;BIRAKMA EMRİ VERME
KONTROL1_CIKIS                      ;
;*********************************
ITICI_GECIKME                       ;HALKA İTİCİ SELENOİD İÇİN
        BTFSC   ITICI,1             ;KESME GECİKMESİ BÖLÜMÜ
        GOTO    ITICI_SAY           ;       "
        BTFSC   ITICI,0             ;İTİLME EMRİ (RING,1 =1) VAR İSE
        GOTO    ITICI_SET           ;O ANDA PARÇA İTİLİR VE BEKLENİR
        BCF     STATUS,5            ;       "
        BTFSS   RING,1              ;       "
        GOTO    ITICI_SET           ;       "
        CLRF    ITICI1              ;       "
        CLRF    ITICI2              ;       "
        BSF     ITICI,1             ;       "
        BSF     PORTA,0             ;       "
ITICI_SAY                           ;       "
        MOVLW H'05'                 ;       "
        SUBWF   ITICI2,W            ;       "
        BTFSS   STATUS,2            ;       "
        GOTO    ITICI_SAY_KONTROL   ;       "
        MOVLW H'66'                 ;       "
        SUBWF   ITICI1,W            ;       "
ITICI_SAY_KONTROL                   ;       "
        BTFSS   STATUS,0            ;       "
        GOTO    ITICI_SET           ;       "
        BCF     ITICI,1             ;       "
        BCF     STATUS,5            ;       "
        BCF     PORTA,0             ;       "
```

```
ITICI_SET                               ;       "
        BTFSS   RING,1                  ;       "
        GOTO    ITICI_SIL               ;       "
        BSF     ITICI,0                 ;       "
        GOTO    ITICI_GECIKME_CIKIS     ;       "
ITICI_SIL                               ;       "
        BCF     ITICI,0                 ;       "
ITICI_GECIKME_CIKIS                     ;       "
;*****************************************
;
BIRAK_GECIKME
        BTFSC   AGECIK,1                ; HALKA BIRAKICI SELENOİD İÇİN
        GOTO    AGECIK_SAY              ; KESME GECİKMESİ BÖLÜMLERİ BAŞLANGICI
        BTFSC   AGECIK,0                ;       "
        GOTO    AGECIK_SET              ; BU BÖLÜMDE BIRAKMA EMRİ (PEG,1=1) VARSA
        BCF     STATUS,5                ; AA REGİSTERİNİN 0. BİTİ HEMEN 1 OLUR
        BTFSS   PEG,1                   ; VE GECIKME BİTİNCE 0 OLUR
        GOTO    AGECIK_SET              ;       "
        CLRF    AGECIK1                 ;       "
        CLRF    AGECIK2                 ;       "
        BSF     AGECIK,1                ;       "
        BSF     AA,0                    ;       "
AGECIK_SAY                              ;       "
        MOVLW H'16'                     ;       "
        SUBWF AGECIK2,W                 ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    AGECIK_SAY_KONTROL      ;       "
        MOVLW H'76'                     ;       "
        SUBWF AGECIK1,W                 ;       "
AGECIK_SAY_KONTROL                      ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    AGECIK_SET              ;       "
        BCF     AGECIK,1                ;       "
        BCF     STATUS,5                ;       "
        BCF     AA,0                    ;       "
AGECIK_SET                              ;       "
        BTFSS   PEG,1                   ;       "
        GOTO    AGECIK_SIL              ;       "
        BSF     AGECIK,0                ;       "
        GOTO    AGECIK_GECIKME_CIKIS    ;       "
AGECIK_SIL                              ;       "
        BCF     AGECIK,0                ;       "
AGECIK_GECIKME_CIKIS                    ;       "
;*****************************************
;
;BIRAKMADAN ÖNCE BEKLE                  ; BU BÖLÜM AA REGİSTERİNİN 0. BİTİNİN 1 OLDUĞU
        BTFSC   AA,0                    ; ZAMAN ZAMA SÜRESİNCE SAYAR VE BELLİ BİR SÜRE
        GOTO    BGECIK_SAY              ; SONRA BB,0 = 1 OLUR. YANİ YUKARIDAKİ BÖLÜMLE
        CLRF    BGECIK1                 ; BİRLİKTE  BIRAKICI SELENOİDİN BIRAKMADAN ÖNCE
        CLRF    BGECIK2                 ; BEKLENMESİ BURADA SAĞLANIR. BURADA DİKKAT
        BCF     BB,0                    ; EDİLMESİ GEREKEN NOKTA AA,0 =1 OLDUĞU ZAMAN
        GOTO    BGECIK_CIKIS            ; SÜRESİNİN BU BÖLÜMDEKİ BEKLEME SÜRESİNDEN
BGECIK_SAY                              ; FAZLA OLMASI GEREKTİĞİDİR
        BTFSC   BB,0                    ;       "
        GOTO    BGECIK_CIKIS            ;       "
        MOVLW H'16'                     ;       "
        SUBWF BGECIK2,W                 ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BGECIK_CIKIS            ;       "
        MOVLW H'66'                     ;       "
        SUBWF BGECIK1,W                 ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BGECIK_CIKIS            ;       "
        BSF     BB,0                    ;       "
BGECIK_CIKIS                            ;       "
;*************************************************************
;
        BTFSC   BIRAK,1                 ; BU BÖLÜM İSE BIRAKICI SELENOİDİN (PORTB,1)
        GOTO    BIRAK_SAY               ; HEM ÇALIŞMASI HEMDE BELLİ BİR SÜRE
        BTFSC   BIRAK,0                 ; ÇALIŞIR HALDEBEKLEMESİ  İÇİN GEREKEN ZAMANI BİZE
        GOTO    BIRAK_SET               ; SAĞLAR BAŞLANGIÇ ŞARTI BB,0 =1, (1 PALS YETERLİDİR)
        BCF     STATUS,5                ; HEMEN PORTA,1=1 OLUR VE BELİLİ BİR SÜRE BEKLEYEREK
```

```
        BTFSS   BB,0                    ; 0 OLUR
        GOTO    BIRAK_SET               ;       "
        CLRF    BIRAK1                  ;       "
        CLRF    BIRAK2                  ;       "
        BSF     BIRAK,1                 ;       "
        BCF     STATUS,5                ;       "
        BSF     PORTA,1                 ;       "
BIRAK_SAY                               ;       "
        MOVLW H'15'                     ;       "
        SUBWF BIRAK2,W                  ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    BIRAK_SAY_KONTROL       ;       "
        MOVLWH'66'                      ;       "
        SUBWF BIRAK1,W                  ;       "
BIRAK_SAY_KONTROL                       ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BIRAK_SET               ;       "
        BCF     BIRAK,1                 ;       "
        BCF     STATUS,5                ;       "
        BCF     PORTA,1                 ;       "
BIRAK_SET                               ;       "
        BTFSS   BB,0                    ;       "
        GOTO    BIRAK_SIL               ;       "
        BSF     BIRAK,0                 ;       "
        GOTO    BIRAK_GECIKME_CIKIS     ;       "
BIRAK_SIL                               ;       "
        BCF     BIRAK,0                 ;       "
BIRAK_GECIKME_CIKIS                     ;       "
;*********************************************************
BOZUK_KONTROL1                  ;BOZUK,0. BİT 0 (BOZUK PARÇA YOK) MI?
        BTFSC   BOZUK,0         ;HAYIR (BOZUK,0=1) İSE HAFIZADA TUT
        GOTO    BOZUK_KONTROL4  ;EVET (BOZUK,0=0 İSE) TEKRAR KONTROL ETMEYE BASLA
        BTFSS   PORTB,4         ;PORTB,4 = 1 Mİ? (BOZUK BÖLGE ALT SENSÖR)
        GOTO    BOZUK_ITME      ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL2                  ;
        BTFSC   PORTB,6         ;PORTB,6 = 0 MI? (BOZUK BÖLGE ÜST SENSÖR)
        GOTO    BOZUK_ITME      ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL3                  ;
        BSF     BOZUK,0         ;BOZUK,0 = 1 YAP (BOZUK PARÇAYI HAFIZAYA AL)
        GOTO    BOZUK_KONTROL4  ;BOZUK_KONTROL4 ETİKETİNE GİT
BOZUK_ITME                      ;
        BCF     BOZUK,0         ;HAFIZAYI SİL
BOZUK_KONTROL4                  ;
        BTFSS   BOZUK,0         ;BOZUK PARÇA ALGILANMIŞ MI?
        GOTO    BOZUK_KONTROL7  ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL5                  ;
        BTFSS   PORTB,3         ;BOZUK PARÇA AYIRICI SELENOİD YANINDA MI?
        GOTO    BOZUK_KONTROL7  ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL6                  ;
        BSF     BOZUK,1         ;BOZUK PARÇAYI İTME EMRİ VER (BOZUK,1=1)
        BCF     BOZUK,0         ;HAFIZAYI SİL
        GOTO    BOZUK_CIKIS     ;BOZUK_CIKIS ETİKETİNE GİT
BOZUK_KONTROL7                  ;
        BCF     BOZUK,1         ;BOZUK PARÇA İTME EMRİ VERME (BOZUK,1=0)
BOZUK_CIKIS                     ;
;*********************************************************
BOZUK_GECIKME                   ;
        BTFSC   BOZ,1           ; BU BÖLÜM BOZUK PARÇA İTEN SELENOİD(PORTA,2) İÇİN
        GOTO    BOZUK_SAY       ; KESME GECİKMESİ SAĞLAR.
        BTFSC   BOZ,0           ; İTME EMRİ (BOZUK,1=1) ALGILANDIĞI ANDA PORTA,2=1
        GOTO    BOZUK_SET       ; OLUR PARÇA İTİLİR.
        BCF     STATUS,5        ; İTTİKTEN SONRA BİR SÜRE BEKLENMESİ SAĞLANIR
        BTFSS   BOZUK,1         ;       "
        GOTO    BOZUK_SET       ;       "
        CLRF    BOZ1            ;       "
        CLRF    BOZ2            ;       "
        BSF     BOZ,1           ;       "
        BSF     PORTA,2         ;       "
BOZUK_SAY                       ;       "
```

```
        MOVLW H'05'                 ;           "
        SUBWF BOZ2,W                ;           "
        BTFSS STATUS,2              ;           "
        GOTO  BOZUK_SAY_KONTROL     ;           "
        MOVLW H'66'                 ;           "
        SUBWF BOZ1,W                ;           "
BOZUK_SAY_KONTROL                   ;           "
        BTFSS STATUS,0              ;           "
        GOTO  BOZUK_SET             ;           "
        BCF   BOZ,1                 ;           "
        BCF   STATUS,5              ;           "
        BCF   PORTA,2               ;           "
BOZUK_SET                           ;           "
        BCF   STATUS,5              ;           "
        BTFSS BOZUK,1               ;           "
        GOTO  BOZUK_SIL             ;           "
        BSF   BOZ,0                 ;           "
        GOTO  BOZUK_GECIKME_CIKIS   ;           "
BOZUK_SIL                           ;           "
        BCF   BOZ,0                 ;           "
BOZUK_GECIKME_CIKIS                 ;           "
        GOTO  START                 ; START ETİKETİNE GİT
;*************************************************************
INT_ALT_PROG                        ;KESME ALT PROGRAMI BAŞLANGICI
        MOVWF SAKLA_W               ;W --> SAKLA_W
        SWAPF STATUS,W              ;STATUS REG. SWAP YAP VE W YE YÜKLE
        MOVWF SAKLA_S               ;W --> SAKLA_S
        BCF   INTCON,2              ;TOIF <-- '0' (zaman aşımı yok)
        INCFSZ ITICI1,F             ;ITICI1 = ITICI1+1, ITICI,1 = 0 MI?
        GOTO  CIKIS1                ;HAYIR İSE
        INCF  ITICI2,F              ;EVET İSE ITICI2 = ITICI2+1
CIKIS1                              ;
        INCFSZ BGECIK1,F            ;KESME ALT PROGRAMINDA BULUNAN BÜTÜN
        GOTO  CIKIS2                ; REGİSTERLER İÇİN AYNI İŞLEMİ
        INCF  BGECIK2,F             ; (İÇERİK ARTIRMA) TEKRARLA
CIKIS2                              ;
        INCFSZ BIRAK1,F             ;
        GOTO  CIKIS3                ;
        INCF  BIRAK2,F              ;
CIKIS3                              ;
        INCFSZ BOZ1,F               ;
        GOTO  CIKIS4                ;
        INCF  BOZ2,F                ;
CIKIS4                              ;
        INCFSZ AGECIK1,F            ;
        GOTO  CIKIS5                ;
        INCF  AGECIK2,F             ;
CIKIS5                              ;
        MOVLW H'00'                 ;TMRO I SIFIRLA
        MOVWF TMR0                  ;           "
        SWAPF SAKLA_S,W             ;W VE STATUS REG. TEKRAR YÜKLE
        MOVWF STATUS                ;           "
        SWAPF SAKLA_W,F             ;           "
        SWAPF SAKLA_W,W             ;           "
        RETFIE                      ;ANA PROGRAMA GERİDÖN
;*************************************************************
GECIKME                             ; ÇİFT DÖNGÜLÜ ZAMAN
        MOVLW H'08'                 ; GECİKMESİ ALT
        MOVWF SAY1                  ; PROGRAMI
DONGU1                              ;           "
        MOVLW H'FF'                 ;           "
        MOVWF SAY2                  ;           "
DONGU2                              ;           "
        DECFSZ SAY2,1               ;           "
        GOTO  DONGU2                ;           "
        DECFSZ SAY1,1               ;           "
        GOTO  DONGU1                ;           "
        RETURN                      ;           "
        END                         ;PROGRAM SONU
```

EK-A.3 Üçüncü Program

```
        LIST    P=16F84A                ;16F84A YI TANIT
        INCLUDE "P16F84A.INC"           ;16F84A.INC DOSYASINI EKLE
        ORG     H'00'                   ;PROGRAM BAŞLANGIC ADRESİ H'00'
        GOTO    BASLA                   ;BASLA ETİKETİNE GİT
        ORG     H'04'                   ;İNTERRUPT(KESME SİNYALİ GELİRSE)
        GOTO    INT_ALT_PROG            ;INT_ALT_PROG ETİKETİNE GİT
BASLA
ITICI           EQU     H'0C'           ;ITICI REGİSTERİNİ VE ADRESİNİ TANIMLA
ITICI1          EQU     H'0D'           ;ITICI1   "     "     "     "
ITICI2          EQU     H'0E'           ;ITICI2   "     "     "     "
SAKLA_W         EQU     H'0F'           ;SAKLA_W  "     "     "     "
SAKLA_S         EQU     H'10'           ;SAKLA_S  "     "     "     "
RING            EQU     H'11'           ;RING     "     "     "     "
MRING           EQU     H'12'           ;MRING    "     "     "     "
PRING           EQU     H'13'           ;PRING    "     "     "     "
MR              EQU     H'14'           ;MR       "     "     "     "
PR              EQU     H'15'           ;PR       "     "     "     "
METAL           EQU     H'16'           ;METAL    "     "     "     "
MRMP            EQU     H'17'           ;MRMP (METAL RING-METAL PEG)    "
MRPP            EQU     H'18'           ;MRPP (METAL RİNG-PLASTİK PEG)  "
PRPP            EQU     H'19'           ;PRPP (PLASTİC RİNG-PLASTİC PEG)  "
PRMP            EQU     H'1A'           ;PRMP (PLASTİC RİNG-METALLİC PEG) "
BIRAK           EQU     H'21'           ;BIRAK    "     "     "     "
BIRAK1          EQU     H'22'           ;BIRAK1   "     "     "     "
BIRAK2          EQU     H'23'           ;BIRAK2   "     "     "     "
PEG             EQU     H'24'           ;PEG      "     "     "     "
BOZ             EQU     H'25'           ;BOZ      "     "     "     "
BOZ1            EQU     H'26'           ;BOZ1     "     "     "     "
BOZ2            EQU     H'27'           ;BOZ2     "     "     "     "
BOZUK           EQU     H'28'           ;BOZUK    "     "     "     "
BGECIK          EQU     H'29'           ;BGECIK   "     "     "     "
BGECIK1         EQU     H'30'           ;BGECIK1  "     "     "     "
BGECIK2         EQU     H'31'           ;BGECIK2  "     "     "     "
BB              EQU     H'32'           ;BB       "     "     "     "
AGECIK          EQU     H'33'           ;AGECIK   "     "     "     "
AGECIK1         EQU     H'34'           ;AGECIK1  "     "     "     "
AGECIK2         EQU     H'35'           ;AGECIK2  "     "     "     "
AA              EQU     H'36'           ;AA       "     "     "     "
SAY1            EQU     H'37'           ;SAY1     "     "     "     "
SAY2            EQU     H'38'           ;SAY2     "     "     "     "
KENAR           EQU     H'39'           ;KENAR    "     "     "     "
PLASTIK         EQU     H'3A'           ;KENAR    "     "     "     "
        BSF     STATUS,5                ;BANK1'E GEÇ
        MOVLW   B'11111111'             ;W REGİSTERİNE H'FF' SAYISINI YÜKLE
        MOVWF   TRISB                   ;W REGİSTERİ--> TRİSB'YE YÜKLE (PORTB'Yİ GİRİŞ YAP)
        MOVLW   B'11110000'             ;W <-- B'00010000'
        MOVWF   TRISA                   ;W --> TRISA (PORTA YI 5. BİT HARİÇ ÇIKIŞ YAP)
        CLRWDT                          ;WDT Yİ SİL
        MOVLW   B'10001111'             ;W <-- B'10001111'
        MOVWF   OPTION_REG              ;W --> OPTION_REG
        BCF     STATUS,5                ;BANK0 A GEÇ
        MOVLW   H'00'                   ;W <-- H'00'
        MOVWF   TMR0                    ;W --> TMR0
        MOVLW   B'10100000'             ;W <-- B'10100000'
        MOVWF   INTCON                  ;W --> INTCON
        CLRF    PORTB                   ;PORTB Yİ TEMİZLE
        CLRF    PORTA                   ;PORTA YI  "
        BCF     ITICI,0                 ;DİĞER REGİSTERLERİ TEMİZLE
        BCF     RING,1                  ;       "
        CLRF    MRING                   ;       "
        CLRF    PRING                   ;       "
        BCF     PEG,1                   ;       "
        BCF     BOZUK,1                 ;       "
        BCF     BB,0                    ;       "
        CLRF    METAL                   ;       "
        CLRF    PLASTIK                 ;       "
```

```
        BCF     AA,0                    ;       "
        CLRF    MR                      ;       "
        CLRF    PR                      ;       "
        CLRF    PRPP                    ;       "
        CLRF    MRMP                    ;       "
        CLRF    PRMP                    ;       "
        CLRF    MRPP                    ;       "
        BSF     PORTA,3                 ;PORTA 3. BİTİNİ 1 YAP ( MOTORLAR ÇALIŞSIN )
START                                   ;
METAL_TANIMLA                           ;
        BTFSS   PORTB,7                 ;METAL PARÇA VAR MI?
        GOTO    METALSIL                ;METAL PARÇA YADA HİÇBİR PARÇA YOKSA
        BSF     METAL,0                 ;METAL REG. 0. BİTİNİ 1 YAP
METALSIL                                ;
        BTFSC   METAL,1                 ;METAL REGISTERI NIN 1.BİTİ 0 MI?
        GOTO    METAL_TAN_CIK           ;METAL REGISTERI NIN 1.BİTİ 1 İSE
        BTFSC   METAL,2                 ;METAL REGISTERI NIN 2.BİTİ 0 MI?
        GOTO    METAL_TAN_CIK           ;METAL REGISTERI NIN 2.BİTİ 1 İSE
METAL_PEG_SET                           ;
        BTFSS   METAL,0                 ;METAL REGISTERI NIN 0.BİTİ 1 Mİ?
        GOTO    METAL_TAN_CIK           ;METAL REGISTERI NIN 0.BİTİ 0 İSE
        BTFSS   PORTB,5                 ;PORTB NİN 5.BİTİ 1 Mİ?
        GOTO    METAL_RING_SET          ;PORTB NİN 5.BİTİ 0 İSE (ÇUBUK YOKSA)
        CALL    GECIKME                 ;BİRAZ BEKLE
        BSF     METAL,2                 ;METAL REGISTERININ 2. BİTİNİ 1 YAP
        BCF     METAL,0                 ;METAL REGISTERININ 0. BİTİNİ 0 YAP
        GOTO    ITICI_KONTROL           ;
METAL_RING_SET                          ;
        BTFSS   PORTB,2                 ;
        GOTO    METAL_TAN_CIK           ;
        CALL    GECIKME                 ;
        BSF     METAL,1                 ;METAL HALKA SET (METAL,1 = 1)
        BCF     METAL,0                 ;METAL,0 = 0
        GOTO    METAL_TAN_CIK           ;
METAL_TAN_CIK                           ;
        BTFSC   METAL,0                 ;METAL PARÇA ALGILANMIŞ İSE
        GOTO    PLAST_TAN_CIK           ;       "
        BTFSC   METAL,1                 ;       "
        GOTO    PLAST_TAN_CIK           ;       "
        BTFSC   METAL,2                 ;       "
        GOTO    PLAST_TAN_CIK           ;PLASTİK_TANIMLA_CIKIS ETİKETİNE GİT
PLASTIK_TANIMLA                         ;
        BSF     PLASTIK,0               ;PARÇAYI PLASTİK OLARAK TANI
        GOTO    ITICI_KONTROL           ;
PLAST_TAN_CIK                           ;
        BCF     PLASTIK,0               ;
;***************************************
ITICI_KONTROL
        BCF     STATUS,5                ; BANK0 A GEÇ
MR_IT                                   ;
        BTFSC   PORTB,5                 ;TOPLAMA BÖLGESİNDE HALKA VARMI
        GOTO    ITME_CIKIS              ;       "
        BTFSS   PORTB,2                 ;       "
        GOTO    RESET_RING              ;HİÇ BİR PARÇA YOKSA
        BTFSS   METAL,1                 ;HALKA VAR, AMA METAL Mİ?
        GOTO    PR_IT                   ;DEĞİL İSE PR_IT E GİT
        BTFSC   MRING,0                 ;MRING,0 YOKSA
        GOTO    RESET_RING              ;VARSA
        BTFSC   PRING,0                 ;PRING,0 YOKSA
        GOTO    RESET_RING              ;VARSA
        MOVLW H'05'                     ;METAL HALKA YETERİ KADAR İTİLMİŞ Mİ?
        SUBWF MR,0                      ;       "
        BTFSC   STATUS,0                ;       "
        GOTO    RESET_RING              ;EVET İSE BİR DAHA İTME
MRING_SET                               ;HAYIR İSE
        BSF     RING,1                  ;İTME EMRİ VER
        BSF     MRING,0                 ;MRİNG REG. 0.BİTİNİ SET ET
        INCF    MR,1                    ;MR=MR+1
        CALL    GECIKME                 ;BİRAZ BEKLE
```

```
        GOTO   ITME_CIKIS              ;
PR_IT
        BTFSS  PLASTIK,0               ;HALKA PLASTİK İSE
        GOTO   RESET_RING              ;DEĞİLSE
        BTFSC  MRING,0                 ;MRING,0 YOKSA
        GOTO   RESET_RING              ;VARSA
        BTFSC  PRING,0                 ;PRING,0 YOKSA
        GOTO   RESET_RING              ;VARSA
        MOVLW H'05'                    ;PLASTIK HALKA YETERİ KADAR İTİLMİŞ Mİ?
        SUBWF PR,0                     ;       "
        BTFSC  STATUS,0                ;       "
        GOTO   RESET_RING              ;EVET İSE BİR DAHA İTME
PRING_SET                              ;HAYIR İSE
        BSF    RING,1                  ;İTME EMRİ VER
        BSF    PRING,0                 ;PRİNG REG. 0.BİTİNİ SET ET
        INCF   PR,1                    ;PR=PR+1
        CALL   GECIKME                 ;BİRAZ BEKLE
        GOTO   ITME_CIKIS              ;
RESET_RING                             ;
        BCF    RING,1                  ;İTME EMRİNİ GERİ ÇEK
METAL_RING_SIL                         ;
        BTFSC  PORTB,5                 ;AYIRICI BÖLGEDE ÇUBUK YOKSA
        GOTO   ITME_CIKIS              ;VARSA
        BTFSC  PORTB,2                 ;HALKA AYIRICI BÖLDEDEN GİTMİŞ İSE
        GOTO   ITME_CIKIS              ;GİTMEMİŞSE
        BCF    METAL,1                 ;METAL,1=0 YAP
        CALL   GECIKME                 ;BİRAZ BEKLE
ITME_CIKIS
ITICI_GECIKME                          ;HALKA İTİCİ SELENOİD İÇİN
        BTFSC  ITICI,1                 ;KESME GECİKMESİ BÖLÜMÜ
        GOTO   ITICI_SAY               ;       "
        BTFSC  ITICI,0                 ;İTİLME EMRİ (RING,1 =1) VAR İSE
        GOTO   ITICI_SET               ;O ANDA PARÇA İTİLİR VE BİR SÜRE
        BCF    STATUS,5                ;İTİLİ(PORTA,0=1) OLARAK BEKLENİR
        BTFSS  RING,1                  ;       "
        GOTO   ITICI_SET               ;       "
        CLRF   ITICI1                  ;       "
        CLRF   ITICI2                  ;       "
        BSF    ITICI,1                 ;       "
        BSF    PORTA,0                 ;       "
ITICI_SAY                              ;       "
        MOVLWH'05'                     ;       "
        SUBWF  ITICI2,W                ;       "
        BTFSS  STATUS,2                ;       "
        GOTO   ITICI_SAY_KONTROL       ;       "
        MOVLWH'66'                     ;       "
        SUBWF  ITICI1,W                ;       "
ITICI_SAY_KONTROL                      ;       "
        BTFSS  STATUS,0                ;       "
        GOTO   ITICI_SET               ;       "
        BCF    ITICI,1                 ;       "
        BCF    STATUS,5                ;       "
        BCF    PORTA,0                 ;       "
ITICI_SET                              ;       "
        BCF    STATUS,5                ;       "
        BTFSS  RING,1                  ;       "
        GOTO   ITICI_SIL               ;       "
        BSF    ITICI,0                 ;       "
        GOTO   ITICI_GECIKME_CIKIS     ;       "
ITICI_SIL                              ;       "
        BCF    ITICI,0                 ;       "
ITICI_GECIKME_CIKIS                    ;       "
;*****************************************
;
BIRAKICI_KONTROL
        BTFSS  PORTB,5                 ;CUBUK VAR MI?
        GOTO   RESET_PEG               ;CUBUK YOKSA
        BTFSC  METAL,2                 ;CUBUK PLASTİK Mİ?
        GOTO   METAL_PEG               ;DEĞİLSE METAL ÇUBUĞA  BAK
PLASTIK_PEG
```

```
        BTFSS   PLASTIK,0       ;PLASTİK ÇUBUK VARSA
        GOTO    RESET_PEG       ;YOKSA
        BTFSC   KENAR,0         ;BİR HALKA İÇİN ÜRETİM
        GOTO    RESET_PEG       ;DEVAM EDİYORSA
        BTFSC   PRING,0         ;TOPLAMA BÖLGESİNDE PLASTİK HALKA VARMI?
        GOTO    PRPP_URET       ;PLASTİK HALKA VAR İSE PRPP ÜRET
        BTFSC   MRING,0         ;TOPLAMA BÖLGESİNDE METAL HALKA VARMI?
        GOTO    MRPP_URET       ;METAL HALKA VAR İSE MRPP ÜRET
        GOTO    RESET_PEG       ;HİÇ BİR HALKA YOKSA
METAL_PEG                       ;METAL CUBUK VARSA
        BTFSC   KENAR,0         ;BİR HALKA İÇİN ÜRETİM
        GOTO    RESET_PEG       ;DEVAM EDİYORSA
        BTFSC   PRING,0         ;TOPLAMA BÖLGESİNDE PLASTİK HALKA VARMI?
        GOTO    PRMP_URET       ;PLASTİK HALKA VAR İSE PRMP ÜRET
        BTFSC   MRING,0         ;TOPLAMA BÖLGESİNDE METAL HALKA VARMI?
        GOTO    MRMP_URET       ;METAL HALKA VAR İSE MRMP ÜRET
        GOTO    RESET_PEG       ;HİÇ BİR HALKA YOKSA
PRPP_URET                       ;PLASTİK HALKA+PLASTIK ÇUBUK ÜRETİMİ
        BTFSC   PORTB,1         ;ÜRETİLMİŞ BİR HALKA BEKLEMİYORSA
        GOTO    RESET_PEG       ;BEKLİYORSA
        MOVLW H'03'             ;YETERLİ SAYIDA PRPP ÜRETİLMEMİŞSE
        SUBWF   PRPP,0          ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    RESET_PEG       ;SAYI YETERLİ İSE
        BSF     PEG,1           ;ÜRETME EMRİ VER
        BSF     KENAR,0         ;
        INCF    PRPP,1          ;PRPP=PRPP+1
        BCF     PRING,0         ;TOPLAMA BÖLGESİNDEKİ PLASTİK HALKAYI SİL
        CALL    GECIKME         ;BİRAZ BEKLE
        GOTO    BIRAK_CIKIS     ;
MRPP_URET                       ;METAL HALKA+PLASTIK ÇUBUK ÜRETİMİ
        BTFSC   PORTB,1         ;ÜRETİLMİŞ BİR HALKA BEKLEMİYORSA
        GOTO    RESET_PEG       ;BEKLİYORSA
        MOVLW H'02'             ;YETERLİ SAYIDA MRPP ÜRETİLMEMİŞSE
        SUBWF   MRPP,0          ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    RESET_PEG       ;SAYI YETERLİ İSE
        BSF     PEG,1           ;ÜRETME EMRİ VER
        INCF    MRPP,1          ;MRPP=MRPP+1
        BSF     KENAR,0         ;
        BCF     MRING,0         ;TOPLAMA BÖLGESİNDEKİ PLASTİK HALKAYI SİL
        CALL    GECIKME         ;BİRAZ BEKLE
        GOTO    BIRAK_CIKIS     ;
PRMP_URET                       ;PLASTİK HALKA+METAL ÇUBUK ÜRETİMİ
        BTFSC   PORTB,1         ;ÜRETİLMİŞ BİR HALKA BEKLEMİYORSA
        GOTO    RESET_PEG       ;BEKLİYORSA
        MOVLW H'02'             ;YETERLİ SAYIDA PRMP ÜRETİLMEMİŞSE
        SUBWF   PRMP,0          ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    RESET_PEG       ;SAYI YETERLİ İSE
        BSF     PEG,1           ;ÜRETME EMRİ VER
        BSF     KENAR,0                 ;
        INCF    PRMP,1          ;PRMP=PRMP+1
        BCF     PRING,0         ;TOPLAMA BÖLGESİNDEKİ PLASTİK HALKAYI SİL
        CALL    GECIKME         ;BİRAZ BEKLE
        GOTO    BIRAK_CIKIS     ;
MRMP_URET                       ;METAL HALKA+METAL ÇUBUK ÜRETİMİ
        BTFSC   PORTB,1         ;ÜRETİLMİŞ BİR HALKA BEKLEMİYORSA
        GOTO    RESET_PEG       ;BEKLİYORSA
        MOVLW H'03'             ;YETERLİ SAYIDA MRMP ÜRETİLMEMİŞSE
        SUBWF   MRMP,0          ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    RESET_PEG       ;SAYI YETERLİ İSE
        BSF     PEG,1           ;ÜRETME EMRİ VER
        BSF     KENAR,0         ;
        INCF    MRMP,1          ;MRMP=MRMP+1
        BCF     MRING,0         ;TOPLAMA BÖLGESİNDEKİ PLASTİK HALKAYI SİL
        CALL    GECIKME         ;BİRAZ BEKLE
        GOTO    BIRAK_CIKIS     ;
```

```
RESET_PEG                               ;
        BCF     PEG,1                   ;ÜRETME (HALKA BIRAKMA) EMRİNİ SİL
METAL_PEG_SIL                           ;
        BTFSC   PORTB,5                 ;ÜRETİLEN BİR CUBUK AYIRICI
        GOTO    BIRAK_CIKIS             ;BÖLGEYİ TERKETMİŞ İSE
        BCF     METAL,2                 ;METAL ÖZELLİĞİNİ SİL
        BCF     PLASTIK,0               ;PLASTİK ÖZELLİĞİNİ SİL
        BCF     KENAR,0                 ;TEKRAR ÜRETİME İZİN VER
        CALL    GECIKME                 ;BİRAZ BEKLE
BIRAK_CIKIS
;************************************************
        BTFSC   AGECIK,1                ; HALKA BIRAKICI SELENOİD İÇİN
        GOTO    AGECIK_SAY              ; KESME GECİKMESİ BÖLÜMLERİ BAŞLANGICI
        BTFSC   AGECIK,0                ;       "
        GOTO    AGECIK_SET              ; BU BÖLÜMDE BIRAKMA EMRİ (PEG,1=1) VARSA
        BCF     STATUS,5                ; AA REGİSTERİNİN 0. BİTİ HEMEN 1 OLUR
        BTFSS   PEG,1                   ; VE GECIKME BİTİNCE 0 OLUR
        GOTO    AGECIK_SET              ;       "
        CLRF    AGECIK1                 ;       "
        CLRF    AGECIK2                 ;       "
        BSF     AGECIK,1                ;       "
        BSF     AA,0                    ;       "
AGECIK_SAY                              ;       "
        MOVLW H'26'                     ;       "
        SUBWF AGECIK2,W                 ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    AGECIK_SAY_KONTROL      ;       "
        MOVLW H'76'                     ;       "
        SUBWF AGECIK1,W                 ;       "
AGECIK_SAY_KONTROL                      ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    AGECIK_SET              ;       "
        BCF     AGECIK,1                ;       "
        BCF     STATUS,5                ;       "
        BCF     AA,0                    ;       "
AGECIK_SET                              ;       "
        BTFSS   PEG,1                   ;       "
        GOTO    AGECIK_SIL              ;       "
        BSF     AGECIK,0                ;       "
        GOTO    AGECIK_GECIKME_CIKIS    ;       "
AGECIK_SIL                              ;       "
        BCF     AGECIK,0                ;       "
AGECIK_GECIKME_CIKIS                    ;       "
;************************************************
        BTFSC   AA,0                    ; BU BÖLÜM AA REGİSTERİNİN 0. BİTİNİN 1 OLDUĞU SONRA
        GOTO    BGECIK_SAY              ; ZAMAN SÜRESİNCE SAYAR VE BELLİ BİR SÜRE
        CLRF    BGECIK1                 ; BB,0 = 1 OLUR. YANİ YUKARIDAKİ BÖLÜMLE BİRLİKTE
        CLRF    BGECIK2                 ; BIRAKICI SELENOİDİN BIRAKMADAN ÖNCE BEKLENMESİ
        BCF     BB,0                    ; BURADA SAĞLANIR. BURADA DİKKAT EDİLMESİGEREKEN
        GOTO    BGECIK_CIKIS            ;  NOKTA AA,0 =1 OLDUĞU ZAMAN SÜRESİNİN
BGECIK_SAY                              ; BU BÖLÜMDEKİ BEKLEME SÜRESİNDEN FAZLA OLMASI
        BTFSC   BB,0                    ; GEREKTİĞİDİR
        GOTO    BGECIK_CIKIS            ;       "
        MOVLW H'25'                     ;       "
        SUBWF BGECIK2,W                 ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BGECIK_CIKIS            ;       "
        MOVLW H'66'                     ;       "
        SUBWF BGECIK1,W                 ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BGECIK_CIKIS            ;       "
        BSF     BB,0                    ;       "
BGECIK_CIKIS                            ;       "
;************************************************
        BTFSC   BIRAK,1                 ; BU BÖLÜM İSE BIRAKICI SELENOİDİN (PORTB,1)
        GOTO    BIRAK_SAY               ; HEM ÇALIŞMASI HEMDE BELLİ BİR SÜRE ÇALIŞIR HALDE
        BTFSC   BIRAK,0                 ; BEKLEMESİ İÇİN GEREKEN ZAMANI SAĞLAR
        GOTO    BIRAK_SET               ; BAŞLANGIÇ ŞARTI (BB,0 =1, 1 PALS YETERLİDİR)
        BCF     STATUS,5                ; OLMASIDIR HEMEN PORTA,1=1 OLUR VE
```

```
        BTFSS   BB,0                    ; BELİLİ BİR SÜRE BEKLEYEREK 0 OLUR
        GOTO    BIRAK_SET               ;       "
        CLRF    BIRAK1                  ;       "
        CLRF    BIRAK2                  ;       "
        BSF     BIRAK,1                 ;       "
        BCF     STATUS,5                ;       "
        BSF     PORTA,1                 ;       "
BIRAK_SAY                               ;       "
        MOVLW H'15'                     ;       "
        SUBWF BIRAK2,W                  ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    BIRAK_SAY_KONTROL       ;       "
        MOVLW H'66'                     ;       "
        SUBWF BIRAK1,W                  ;       "
BIRAK_SAY_KONTROL                       ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BIRAK_SET               ;       "
        BCF     BIRAK,1                 ;       "
        BCF     STATUS,5                ;       "
        BCF     PORTA,1                 ;       "
BIRAK_SET                               ;       "
        BCF     STATUS,5                ;       "
        BTFSS   BB,0                    ;       "
        GOTO    BIRAK_SIL               ;       "
        BSF     BIRAK,0                 ;       "
        GOTO    BIRAK_GECIKME_CIKIS     ;       "
BIRAK_SIL                               ;       "
        BCF     BIRAK,0                 ;       "
BIRAK_GECIKME_CIKIS                     ;       "
;*********************************************************
BOZUK_KONTROL1                          ;BOZUK,0. BİT 0 (BOZUK PARÇA YOK) MI?
        BTFSC   BOZUK,0                 ;HAYIR (BOZUK,0=1) İSE HAFIZADA TUT
        GOTO    BOZUK_KONTROL4          ;EVET (BOZUK,0=0 İSE) TEKRAR KONTROL ETMEYE BASLA
        BTFSS   PORTB,4                 ;PORTB,4 = 1 Mİ? (BOZUK BÖLGE ALT SENSÖR)
        GOTO    BOZUK_ITME              ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL2                          ;
        BTFSC   PORTB,6                 ;PORTB,6 = 0 MI? (BOZUK BÖLGE ÜST SENSÖR)
        GOTO    BOZUK_ITME              ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL3                          ;
        BSF     BOZUK,0                 ;BOZUK,0 = 1 YAP (BOZUK PARÇAYI HAFIZAYA AL)
        GOTO    BOZUK_KONTROL4          ;BOZUK_KONTROL4 ETİKETİNE GİT
BOZUK_ITME                              ;
        BCF     BOZUK,0                 ;HAFIZAYI SİL
BOZUK_KONTROL4                          ;
        BTFSS   BOZUK,0                 ;BOZUK PARÇA ALGILANMIŞ MI?
        GOTO    BOZUK_KONTROL7          ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL5                          ;
        BTFSS   PORTB,3                 ;BOZUK PARÇA AYIRICI SELENOİD YANINDA MI?
        GOTO    BOZUK_KONTROL7          ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL6                          ;
        BSF     BOZUK,1                 ;BOZUK PARÇAYI İTME EMRİ VER (BOZUK,1=1)
        BCF     BOZUK,0                 ;HAFIZAYI SİL
        GOTO    BOZUK_CIKIS             ;BOZUK_CIKIS ETİKETİNE GİT
BOZUK_KONTROL7                          ;
        BCF     BOZUK,1                 ;BOZUK PARÇA İTME EMRİ VERME (BOZUK,1=0)
BOZUK_CIKIS                             ;
;*********************************************************
        BTFSC   BOZ,1                   ; BU BÖLÜM BOZUK PARÇA İTEN SELENOİD(PORTA,2)
        GOTO    BOZUK_SAY               ; İÇİN KESME GECİKMESİ SAĞLAR.
        BTFSC   BOZ,0                   ; İTME EMRİ (BOZUK,1=1) ALGILANDIĞI ANDA
        GOTO    BOZUK_SET               ; PORTA,2=1 OLUR PARÇA İTİLİR.
        BCF     STATUS,5                ; İTTİKTEN SONRA BİR SÜRE BEKLENMESİ SAĞLANIR
        BTFSS   BOZUK,1                 ;       "
        GOTO    BOZUK_SET               ;       "
        CLRF    BOZ1                    ;       "
        CLRF    BOZ2                    ;       "
        BSF     BOZ,1                   ;       "
        BSF     PORTA,2                 ;       "
BOZUK_SAY                               ;       "
```

```
        MOVLW H'05'                 ;           "
        SUBWF BOZ2,W                ;           "
        BTFSS STATUS,2              ;           "
        GOTO  BOZUK_SAY_KONTROL     ;           "
        MOVLW H'66'                 ;           "
        SUBWF BOZ1,W                ;           "
BOZUK_SAY_KONTROL                   ;           "
        BTFSS STATUS,0              ;           "
        GOTO  BOZUK_SET             ;           "
        BCF   BOZ,1                 ;           "
        BCF   STATUS,5              ;           "
        BCF   PORTA,2               ;           "
BOZUK_SET                           ;           "
        BCF   STATUS,5              ;           "
        BTFSS BOZUK,1               ;           "
        GOTO  BOZUK_SIL             ;           "
        BSF   BOZ,0                 ;           "
        GOTO  BOZUK_GECIKME_CIKIS   ;           "
BOZUK_SIL                           ;           "
        BCF   BOZ,0                 ;           "
BOZUK_GECIKME_CIKIS                 ;           "
        GOTO  START                 ; BAŞA DÖN
;***********************************************
INT_ALT_PROG                        ;KESME ALT PROGRAMI BAŞLANGICI
        MOVWF SAKLA_W               ;W --> SAKLA_W
        SWAPF STATUS,W              ;STATUS REG. SWAP YAP VE W YE YÜKLE
        MOVWF SAKLA_S               ;W --> SAKLA_S
        BCF   INTCON,2              ;TOIF <-- '0' (zaman aşımı yok)
        INCFSZ ITICI1,F             ;ITICI1 = ITICI1+1, ITICI,1 = 0 MI?
        GOTO  CIKIS1                ;HAYIR İSE
        INCF  ITICI2,F              ;EVET İSE ITICI2 = ITICI2+1
CIKIS1                              ;
        INCFSZ BGECIK1,F            ;KESME ALT PROGRAMINDA BULUNAN BÜTÜN
        GOTO  CIKIS2                ; REGİSTERLER İÇİN AYNI İŞLEMİ
        INCF  BGECIK2,F             ; (İÇERİK ARTIRMA) TEKRARLA
CIKIS2                              ;           "
        INCFSZ BIRAK1,F             ;           "
        GOTO  CIKIS3                ;           "
        INCF  BIRAK2,F              ;           "
CIKIS3                              ;           "
        INCFSZ BOZ1,F               ;           "
        GOTO  CIKIS4                ;           "
        INCF  BOZ2,F                ;           "
CIKIS4                              ;           "
        INCFSZ AGECIK1,F            ;           "
        GOTO  CIKIS5                ;           "
        INCF  AGECIK2,F             ;           "
CIKIS5                              ;           "
        MOVLW H'00'                 ;TMR0'I SIFIRLA
        MOVWF TMR0                  ;           "
        SWAPF SAKLA_S,W             ;W VE STATUS REG. TEKRAR YÜKLE
        MOVWF STATUS                ;           "
        SWAPF SAKLA_W,F             ;           "
        SWAPF SAKLA_W,W             ;           "
        RETFIE                      ;ANA PROGRAMA GERİDÖN
;***********************************
GECIKME                             ; ÇİFT DÖNGÜLÜ ZAMAN
        MOVLW H'08'                 ; GECİKMESİ ALT
        MOVWF SAY1                  ; PROGRAMI
DONGU1                              ;           "
        MOVLW H'FF'                 ;           "
        MOVWF SAY2                  ;           "
DONGU2                              ;           "
        DECFSZ SAY2,1               ;           "
        GOTO  DONGU2                ;           "
        DECFSZ SAY1,1               ;           "
        GOTO  DONGU1                ;           "
        RETURN                      ;           "
        END                         ;PPROGRAM SONU
```

EK-A.4 Dördüncü Program

```
        LIST    P=16F84A                ;16F84A YI TANIT
        INCLUDE "P16F84A.INC"           ;16F84A.INC DOSYASINI EKLE
        ORG     H'00'                   ;PROGRAM BAŞLANGIC ADRESİ H'00'
        GOTO    BASLA                   ;BASLA ETİKETİNE GİT
        ORG     H'04'                   ;İNTERRUPT(KESME SİNYALİ GELİRSE)
        GOTO    INT_ALT_PROG            ;INT_ALT_PROG ETİKETİNE GİT
BASLA
ITICI           EQU     H'0C'           ;FİLE REGİSTERLERİ TANIMLA
ITICI1          EQU     H'0D'           ;       "
ITICI2          EQU     H'0E'           ;       "
SAKLA_W         EQU     H'0F'           ;       "
SAKLA_S         EQU     H'10'           ;       "
RING            EQU     H'11'           ;       "
MRING           EQU     H'12'           ;       "
PRING           EQU     H'13'           ;       "
MR              EQU     H'14'           ;       "
PR              EQU     H'15'           ;       "
METAL           EQU     H'16'           ;       "
MRMP            EQU     H'17'           ;       "
MRPP            EQU     H'18'           ;       "
PRPP            EQU     H'19'           ;       "
PRMP            EQU     H'1A'           ;       "
BIRAK           EQU     H'1B'           ;       "
BIRAK1          EQU     H'1C'           ;       "
BIRAK2          EQU     H'1D'           ;       "
PEG             EQU     H'1E'           ;       "
BOZ             EQU     H'1F'           ;       "
BOZ1            EQU     H'20'           ;       "
BOZ2            EQU     H'21'           ;       "
BOZUK           EQU     H'22'           ;       "
KENAR           EQU     H'23'           ;       "
CUBUK           EQU     H'24'           ;       "
PCUBUK          EQU     H'25'           ;       "
MCUBUK          EQU     H'26'           ;       "
MARKER          EQU     H'29'           ;       "
TIMER           EQU     H'2A'           ;       "
HALKA           EQU     H'2B'           ;       "
MHALKA          EQU     H'2C'           ;       "
PHALKA          EQU     H'2D'           ;       "
URET            EQU     H'2E'           ;       "
SAY1            EQU     H'2F'           ;       "
SAY2            EQU     H'30'           ;       "
PLASTIK         EQU     H'31'           ;       "
AGECIK          EQU     H'32'           ;       "
AGECIK1         EQU     H'33'           ;       "
AGECIK2         EQU     H'34'           ;       "
BGECIK          EQU     H'35'           ;       "
BGECIK1         EQU     H'36'           ;       "
BGECIK2         EQU     H'37'           ;       "
CGECIK          EQU     H'38'           ;       "
CGECIK1         EQU     H'39'           ;       "
CGECIK2         EQU     H'3A'           ;       "
DGECIK          EQU     H'3B'           ;       "
DGECIK1         EQU     H'3C'           ;       "
DGECIK2         EQU     H'3D'           ;       "
EGECIK          EQU     H'3E'           ;       "
EGECIK1         EQU     H'3F'           ;       "
EGECIK2         EQU     H'40'           ;       "
AA              EQU     H'41'           ;       "
BB              EQU     H'42'           ;       "
CC              EQU     H'43'           ;       "
DD              EQU     H'44'           ;       "
EE              EQU     H'45'           ;       "
        BSF     STATUS,5                ;       "
        CLRF    PORTB                   ;PORTB Yİ SIFIRLA
        CLRF    PORTA                   ;PORTAYI SIFIRLA
```

```
        BSF     STATUS,5                ;BANK1 E GEÇ
        MOVLW B'11111111'               ;W REGİSTERİNE H'FF' SAYISINI YÜKLE
        MOVWF TRISB                     ;W REGİSTERİ--> TRİSB'YE YÜKLE (PORTB'Yİ GİRİŞ YAP)
        MOVLW B'11110000'               ;W <-- B'00010000'
        MOVWF TRISA                     ;W --> TRISA (PORTA YI 5. BİT HARİÇ ÇIKIŞ YAP)
        CLRWDT                          ;WDT Yİ SİL
        MOVLW B'10001111'               ;W <-- B'10001111'
        MOVWF OPTION_REG                ;W --> OPTION_REG
        BCF     STATUS,5                ;BANK0 A GEÇ
        MOVLW H'00'                     ;W <-- H'00'
        MOVWF TMR0                      ;W --> TMR0
        MOVLW B'10100000'               ;W <-- B'10100000'
        MOVWF INTCON                    ;W --> INTCON
        CLRF    PORTB                   ;PORTB Yİ SIFIRLA
        CLRF    PORTA                   ;PORTAYI SIFIRLA
        BCF     ITICI,0                 ;DİĞER FİLE REGİSTERLERİ SIFIRLA
        BCF     RING,1                  ;       "
        CLRF    MRING                   ;       "
        CLRF    PRING                   ;       "
        CLRF    HALKA                   ;       "
        BCF     PEG,1                   ;       "
        CLRF    BOZUK                   ;       "
        CLRF    KENAR                   ;       "
        CLRF    METAL                   ;       "
        CLRF    TIMER                   ;       "
        BCF     MHALKA,0                ;       "
        BCF     PHALKA,0                ;       "
        BCF     URET,0                  ;       "
        BCF     PLASTIK,0               ;       "
        CLRF    KENAR                   ;       "
        CLRF    CUBUK                   ;       "
        CLRF    MCUBUK                  ;       "
        CLRF    PCUBUK                  ;       "
        CLRF    MARKER                  ;       "
        CLRF    SAY1                    ;       "
        CLRF    SAY2                    ;       "
        CLRF    MR                      ;       "
        CLRF    PR                      ;       "
        CLRF    PRPP                    ;       "
        CLRF    MRMP                    ;       "
        CLRF    PRMP                    ;       "
        CLRF    MRPP                    ;       "
        CLRF    AA                      ;       "
        CLRF    BB                      ;       "
        CLRF    CC                      ;       "
        CLRF    DD                      ;       "
        CLRF    EE                      ;       "
        BSF     PORTA,3                 ;PORTA 3. BİTİNİ 1 YAP ( MOTORLAR ÇALIŞSIN )
START
;*****************************************
METAL_TANIMLA                           ;
        BTFSS   PORTB,7                 ;METAL PARÇA VAR MI?
        GOTO    METALSIL                ;METAL PARÇA YADA HİÇBİR PARÇA YOKSA
        BSF     METAL,0                 ;METAL REG. 0. BİTİNİ 1 YAP
METALSIL                                ;
        BTFSC   METAL,1                 ;METAL REGISTERI NIN 1.BİTİ 0 MI?
        GOTO    METAL_TAN_CIK           ;METAL REGISTERI NIN 1.BİTİ 1 İSE
        BTFSC   METAL,2                 ;METAL REGISTERI NIN 2.BİTİ 0 MI?
        GOTO    METAL_TAN_CIK           ;METAL REGISTERI NIN 2.BİTİ 1 İSE
METAL_PEG_SET                           ;
        BTFSS   METAL,0                 ;METAL REGISTERI NIN 0.BİTİ 1 Mİ?
        GOTO    METAL_TAN_CIK           ;METAL REGISTERI NIN 0.BİTİ 0 İSE
        BTFSS   PORTB,5                 ;PORTB NİN 5.BİTİ 1 Mİ?
        GOTO    METAL_RING_SET          ;PORTB NİN 5.BİTİ 0 İSE (ÇUBUK YOKSA)
        CALL    GECIKME                 ;BİRAZ BEKLE
        BSF     METAL,2                 ;METAL REGISTERININ 2. BİTİNİ 1 YAP
        BCF     METAL,0                 ;METAL REGISTERININ 0. BİTİNİ 0 YAP
        GOTO    ITME_CIKIS              ;
METAL_RING_SET                          ;
```

```
        BTFSS   PORTB,2             ;
        GOTO    METAL_TAN_CIK       ;
        CALL    GECIKME             ;
        BSF     METAL,1             ;METAL HALKA SET (METAL,1 = 1)
        BCF     METAL,0             ;METAL,0 = 0
        GOTO    METAL_TAN_CIK       ;
METAL_TAN_CIK                       ;
        BTFSC   METAL,0             ;METAL PARÇA ALGILANMIŞ İSE
        GOTO    PLAST_TAN_CIK       ;       "
        BTFSC   METAL,1             ;       "
        GOTO    PLAST_TAN_CIK       ;       "
        BTFSC   METAL,2             ;       "
        GOTO    PLAST_TAN_CIK       ;PLASTİK_TAN_CIK ETİKETİNE GİT
PLASTIK_TANIMLA                     ;
        BSF     PLASTIK,0           ;PARÇAYI PLASTİK OLARAK TANI
        GOTO    ITICI_KONTROL       ;
PLAST_TAN_CIK                       ;
        BCF     PLASTIK,0           ;
;*************************************
ITICI_KONTROL                       ;
        BCF     STATUS,5            ;BANK0 A GEC
        BTFSC   RING,1              ;ITME EMRİ VARSA, VE İTİCI ÇALIŞIYORSA
        GOTO    RESET_RING          ;RESET_RING ETİKETİNE GİT
        BTFSC   PORTA,0             ;       "
        GOTO    RESET_RING          ;       "
        BTFSC   PORTB,5             ;CUBUK VARSA
        GOTO    ITME_CIKIS          ;ITME_CIKIS ETİKETİNE GİT
        BTFSS   PORTB,2             ;
        GOTO    RESET_RING          ;HİÇ BİRİ YOKSA
        BTFSS   METAL,1             ;HALKA VAR AMA METAL Mİ?
        GOTO    PR_IT               ;METAL DEĞİLSE PR_IT ETİKETİNE GİT
MR_IT                               ;
        CALL    GECIKME             ;BİRAZ BEKLE
        BTFSC   MHALKA,0            ;MHALKA YOKSA
        GOTO    RESET_RING          ;VARSA
        BTFSC   PHALKA,0            ;PHALKA YOKSA
        GOTO    RESET_RING          ;VARSA
        BTFSS   MR,2                ;METAL HALKA YETERİ KADAR 5 ADET
        GOTO    CIK                 ;İTİLMİŞ İSE
        BTFSC   MR,0                ;       "
        GOTO    RESET_RING          ;RESET_RING ETİKETİNE GİT
CIK                                 ;
;*********************************  ;EĞER METAL HALKA YETERİ KADAR
        BTFSC   HALKA,5             ;İTİLMEMİŞ İSE HALKA REG. 5. BİTİNE BAK
        GOTO    RESET_RING          ;5. BİT DOLU(1)İSE RESET_RING ETİKETİNE GİT
        BTFSC   HALKA,4             ;HALKA REG. 5. BİTİ BOŞ İSE 4. BİTE BAK
        GOTO    MRING_5_SET         ;4. BİT DOLU(1) İSE MRING_5_SET ETİK. GİT
        BTFSC   HALKA,3             ;4. BİT BOŞ İSE 3. BİTE BAK
        GOTO    MRING_4_SET         ;3. BİT DOLU(1) İSE MRING_4_SET ETİK. GİT
        BTFSC   HALKA,2             ;3. BİT BOŞ(0) İSE 2. BİTE BAK
        GOTO    MRING_3_SET         ;2. BİT DOLU(1) İSE MRING_3_SET ETİK. GİT
        BTFSC   HALKA,1             ;2. BİT BOŞ(0) İSE 1. BİTE BAK
        GOTO    MRING_2_SET         ;1. BİT DOLU(1) İSE MRING_2_SET ETİK. GİT
MRING_1_SET                         ;1. BİT BOŞ(0) İSE 1. BİTİ DOLDUR
        BSF     MRING,1             ;MRING,1=1
        BSF     HALKA,1             ;HALKA,1=1
        GOTO    MRING_SET           ;MRING_SET ETİKETİNE GİT
MRING_2_SET                         ;
        BSF     MRING,2             ;MRING,2=1
        BSF     HALKA,2             ;HALKA,2=1
        GOTO    MRING_SET           ;
MRING_3_SET                         ;
        BSF     MRING,3             ;MRING,3=1
        BSF     HALKA,3             ;HALKA,3=1
        GOTO    MRING_SET           ;
MRING_4_SET                         ;
        BSF     MRING,4             ;MRING,4=1
        BSF     HALKA,4             ;HALKA,4=1
        GOTO    MRING_SET           ;
```

```
MRING_5_SET                             ;
        BSF     MRING,5                 ;MRING,5=1
        BSF     HALKA,5                 ;HALKA,5=1
        GOTO    MRING_SET               ;
MRING_SET                               ;
        BSF     RING,1                  ;RING,1=1 METAL HALKA İTME EMRİ VER
        BSF     MHALKA,0                ;MHALKA,0=1
        INCF    MR,1                    ;MR=MR+1
        GOTO    ITME_CIKIS              ;ITME_CIKIS ETİKETİNE GİT
;***************************************
PR_IT                                   ;
        CALL    GECIKME                 ;BİRAZ BEKLE
        BTFSC   MHALKA,0                ;MHALKA,0 YOK MU?
        GOTO    RESET_RING              ;VARSA
        BTFSC   PHALKA,0                ;PHALKA,0 YOK MU?
        GOTO    RESET_RING              ;VARSA (İTİLECEK HALKA HENÜZ AYIRICI
;*****************                      ;BÖLGEYİ TERKETMEMİŞSE)
        BTFSS   PR,2                    ;PLASTİK HALKA YETERİ KADAR 5 ADET
        GOTO    CIK1                    ;İTİLMİŞ İSE
        BTFSC   PR,0                    ;       "
        GOTO    RESET_RING              ;RESET_RING ETİKETİNE GİT
CIK1                                    ;EĞER PLASTİK HALKA YETERİ KADAR
        BTFSC   HALKA,5                 ;İTİLMEMİŞ İSE HALKA REG. 5. BİTİNE BAK
        GOTO    RESET_RING              ;5. BİT DOLU(1)İSE RESET_RING ETİKETİNE GİT
        BTFSC   HALKA,4                 ;HALKA REG. 5. BİTİ BOŞ İSE 4. BİTE BAK
        GOTO    PRING_5_SET             ;4. BİT DOLU(1) İSE PRING_5_SET ETİK. GİT
        BTFSC   HALKA,3                 ;4. BİT BOŞ İSE 3. BİTE BAK
        GOTO    PRING_4_SET             ;3. BİT DOLU(1) İSE PRING_4_SET ETİK. GİT
        BTFSC   HALKA,2                 ;3. BİT BOŞ(0) İSE 2. BİTE BAK
        GOTO    PRING_3_SET             ;2. BİT DOLU(1) İSE PRING_3_SET ETİK. GİT
        BTFSC   HALKA,1                 ;2. BİT BOŞ(0) İSE 1. BİTE BAK
        GOTO    PRING_2_SET             ;1. BİT DOLU(1) İSE PRING_2_SET ETİK. GİT
PRING_1_SET                             ;1. BİT BOŞ(0) İSE 1. BİTİ DOLDUR
        BSF     PRING,1                 ;PRING,1=1
        BSF     HALKA,1                 ;HALKA,1=1
        GOTO    PRING_SET               ;PRING_SET ETİKETİNE GİT
PRING_2_SET                             ;
        BSF     PRING,2                 ;PRING,2=1
        BSF     HALKA,2                 ;HALKA,2=1
        GOTO    PRING_SET               ;
PRING_3_SET                             ;
        BSF     PRING,3                 ;PRING,3=1
        BSF     HALKA,3                 ;HALKA,3=1
        GOTO    PRING_SET               ;
PRING_4_SET                             ;
        BSF     PRING,4                 ;PRING,4=1
        BSF     HALKA,4                 ;HALKA,4=1
        GOTO    PRING_SET               ;
PRING_5_SET                             ;
        BSF     PRING,5                 ;PRING,5=1
        BSF     HALKA,5                 ;HALKA,5=1
        GOTO    PRING_SET               ;
PRING_SET                               ;
        BSF     RING,1                  ;RING,1=1 PLASTİK HALKA İTME EMRİ VER
        BSF     PHALKA,0                ;PHALKA,0=1
        INCF    PR,1                    ;PR=PR+1
        GOTO    ITME_CIKIS              ;ITME_CIKIS ETİKETİNE GİT
;*********************************
RESET_RING                              ;
        BCF     RING,1                  ;İTME EMRİNİ GERİ ÇEK
METAL_RING_SIL                          ;
        BTFSC   PORTB,5                 ;RİNG AYIRICI BÖLGEYİ TERKETMİŞ İSE
        GOTO    ITME_CIKIS              ;       "
        BTFSC   PORTB,2                 ;       "
        GOTO    ITME_CIKIS              ;       "
        CALL    GECIKME                 ;BİRAZ BEKLE
        BCF     METAL,1                 ;METAL,1 =0
        BCF     MHALKA,0                ;MHALKA,0 =0
        BCF     PHALKA,0                ;PHALKA,0 =0 YAP
```

```
ITME_CIKIS                      ;
;*********************************
BIRAKICI_KONTROL                ;
        BTFSC   MARKER,0        ;
        GOTO    DOLDUR_CIKIS    ;
        BTFSS   PORTB,5         ;ÇUBUK VAR MI?
        GOTO    DOLDUR_CIKIS    ;ÇUBUK YOKSA DOLDUR_CIKIS ETİKETİNE GİT
        BTFSC   METAL,2         ;ÇUBUK PLASTİK Mİ?
        GOTO    M_PEG_DOLDUR    ;DEĞİLSE METAL PEG E BAK
P_PEG_DOLDUR                    ;
        BTFSC   KENAR,0         ;KENAR,0=0 MI?
        GOTO    RESET_PEG       ;HAYIR
        BTFSC   MARKER,0        ;MARKER,0=0 MI?
        GOTO    RESET_PEG       ;HAYIR
        BTFSS   PLASTIK,0       ;PLASTIK,0=1 Mİ?
        GOTO    RESET_PEG       ;HAYIR
        BTFSC   CUBUK,3         ;GELEN PLASTİK CUBUK İÇİN CUBUK REGİS.
        GOTO    DOLDUR_CIKIS    ;3. BİTİNE BAK DOLU İSE DOLDUR_CIKIS A GİT
        BTFSC   CUBUK,2         ;3. BİT BOŞ(0) İSE 2. BİTE BAK
        GOTO    PCUBUK_3_SET    ;2. BİT DOLU(1) İSE PCUBUK_3_SET ETİK. GİT
        BTFSC   CUBUK,1         ;2. BİT BOŞ(0) İSE 1. BİTE BAK
        GOTO    PCUBUK_2_SET    ;1. BİT DOLU(1) İSE PCUBUK_2_SET ETİK. GİT
        GOTO    PCUBUK_1_SET    ;1. BİT BOŞ(0) İSE PCUBUK VE CUBUK REGİS.
PCUBUK_1_SET                    ;1. BİTLERİNİ DOLDUR
        BSF     PCUBUK,1        ;PCUBUK,1 =1
        BSF     CUBUK,1         ;CUBUK,1 =1
        GOTO    CUBUK_SET       ;
PCUBUK_2_SET                    ;
        BSF     PCUBUK,2        ;PCUBUK,2 =1
        BSF     CUBUK,2         ;CUBUK,2 =1
        GOTO    CUBUK_SET       ;
PCUBUK_3_SET                    ;
        BSF     PCUBUK,3        ;PCUBUK,3 =1
        BSF     CUBUK,3         ;CUBUK,3 =1
        GOTO    CUBUK_SET       ;
;*********************************
M_PEG_DOLDUR                    ;CUBUK METAL İSE
        BTFSC   KENAR,0         ;KENAR,0 = 0 İSE
        GOTO    RESET_PEG       ;DEĞİLSE
        BTFSC   MARKER,0        ;MARKER,0 = 0 İSE
        GOTO    RESET_PEG       ;DEĞİLSE
        BTFSC   CUBUK,3         ;GELEN METAL CUBUK İÇİN CUBUK REGİS.
        GOTO    DOLDUR_CIKIS    ;3. BİTİNE BAK DOLU İSE DOLDUR_CIKIS A GİT
        BTFSC   CUBUK,2         ;3. BİT BOŞ(0) İSE 2. BİTE BAK
        GOTO    MCUBUK_3_SET    ;2. BİT DOLU(1) İSE MCUBUK_3_SET ETİK. GİT
        BTFSC   CUBUK,1         ;2. BİT BOŞ(0) İSE 1. BİTE BAK
        GOTO    MCUBUK_2_SET    ;1. BİT DOLU(1) İSE MCUBUK_2_SET ETİK. GİT
        GOTO    MCUBUK_1_SET    ;1. BİT BOŞ(0) İSE MCUBUK VE CUBUK REGİS.
MCUBUK_1_SET                    ;1. BİTLERİNİ DOLDUR
        BSF     MCUBUK,1        ;PCUBUK,1 =1
        BSF     CUBUK,1         ;CUBUK,1 =1
        GOTO    CUBUK_SET       ;
MCUBUK_2_SET                    ;
        BSF     MCUBUK,2        ;PCUBUK,2 =1
        BSF     CUBUK,2         ;CUBUK,2 =1
        GOTO    CUBUK_SET       ;
MCUBUK_3_SET                    ;
        BSF     MCUBUK,3        ;PCUBUK,3 =1
        BSF     CUBUK,3         ;CUBUK,3 =1
        GOTO    CUBUK_SET       ;
CUBUK_SET                       ;
        BSF     MARKER,0        ;
DOLDUR_CIKIS                    ;
;*************************************************
;PEG_GELIRSE_ZAMANLAYICIYI_BASLAT;
        BTFSC   AGECIK,1        ;BU BÖLÜM CUBUK GELİRSE BELLİ BİR SÜRE SONRA
        GOTO    AGECIK_SAY      ;HALKA İLE KARŞILAŞTIRMAYI SAĞLAYACAK OLAN
        BTFSC   AGECIK,0        ;KESME ZAMAN GECİKMESİNİN BAŞLANGICIDIR.
        GOTO    AGECIK_SET      ;CUBUK GELDİĞİNDE(PORTB,5=1) HEMEN AA REG.
```

```
        BCF     STATUS,5                ;0. BİTİ 1 OLUR VE BELLİ BİR SÜRE AA REG.
        BTFSS   PORTB,5                 ;0. BİTİNİ SET EDER, SÜRE BİTİNCE RESET EDER.
        GOTO    AGECIK_SET              ;       "
AGECIK_RESET                            ;       "
        CLRF    AGECIK1                 ;       "
        CLRF    AGECIK2                 ;       "
        BSF     AGECIK,1                ;       "
        BSF     AA,0                    ;       "
AGECIK_SAY                              ;       "
        MOVLW H'20'                     ;       "
        SUBWF AGECIK2,W                 ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    AGECIK_SAY_KONTROL      ;       "
        MOVLW H'38'                     ;       "
        SUBWF AGECIK1,W                 ;       "
AGECIK_SAY_KONTROL                      ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    AGECIK_SET              ;       "
AGECIK_ITME                             ;       "
        BCF     AGECIK,1                ;       "
        BCF     AA,0                    ;       "
AGECIK_SET                              ;       "
        BCF     STATUS,5                ;       "
        BTFSS   PORTB,5                 ;       "
        GOTO    AGECIK_SIL              ;       "
        BSF     AGECIK,0                ;       "
        GOTO    AGECIK_GECIKME_CIKIS;           "
AGECIK_SIL                              ;       "
        BCF     AGECIK,0                ;       "
AGECIK_GECIKME_CIKIS                    ;       "
;************************************************
        BTFSC   AA,0                    ;BU BÖLÜM AA,0=1 OLDUĞU ANDA ÇOK KISA
        GOTO    BGECIK_SAY              ;BİR ZAMAN DİLİMİ BB REG. 0. BİTİNİ
; RESET                                 ;SET EDER
        CLRF    BGECIK1                 ;       "
        CLRF    BGECIK2                 ;       "
        BSF     BB,0                    ;       "
        GOTO    BGECIK_CIKIS            ;       "
BGECIK_SAY                              ;       "
        BTFSS   BB,0                    ;       "
        GOTO    BGECIK_CIKIS            ;       "
        MOVLW H'00'                     ;       "
        SUBWF BGECIK2,W                 ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BGECIK_CIKIS            ;       "
        MOVLW H'78'                     ;       "
        SUBWF BGECIK1,W                 ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BGECIK_CIKIS            ;       "
        BCF     BB,0                    ;       "
BGECIK_CIKIS                            ;       "
;************************************************
        BTFSC   CGECIK,1                ;BU BÖLÜM İSE BB,0=1 OLDUĞU ANDA
        GOTO    CGECIK_SAY              ;CC REG. 0. BİTİNİN BELLİ BİR SÜRE
        BTFSC   CGECIK,0                ;SET OLMASINI VE ZAMAN DOLUNCA
        GOTO    CGECIK_SET              ;RESET OLMASINI SAĞLAR.
        BCF     STATUS,5                ;       "
        BTFSS   BB,0                    ;       "
        GOTO    CGECIK_SET              ;       "
CGECIK_RESET                            ;       "
        CLRF    CGECIK1                 ;       "
        CLRF    CGECIK2                 ;       "
        BSF     CGECIK,1                ;       "
        BSF     CC,0                    ;       "
CGECIK_SAY                              ;       "
        MOVLWH'24'                      ;       "
        SUBWF CGECIK2,W                 ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    CGECIK_SAY_KONTROL      ;       "
```

```
        MOVLWH'99'                  ;       "
        SUBWF CGECIK1,W             ;       "
CGECIK_SAY_KONTROL                  ;       "
        BTFSS   STATUS,0            ;       "
        GOTO    CGECIK_SET          ;       "
CGECIK_ITME                         ;       "
        BCF     CGECIK,1            ;       "
        BCF     STATUS,5            ;       "
        BCF     CC,0                ;       "
CGECIK_SET                          ;       "
        BTFSS   BB,0                ;       "
        GOTO    CGECIK_SIL          ;       "
        BSF     CGECIK,0            ;       "
        GOTO    CGECIK_GECIKME_CIKIS;       "
CGECIK_SIL                          ;       "
        BCF     CGECIK,0            ;       "
CGECIK_GECIKME_CIKIS                ;       "
;************************************************
        BTFSC   CC,0                ;BU BÖLÜMDE CC REG. 0. BİTİNİN
        GOTO    DGECIK_SAY          ;1 (SET) OLDUĞU ZAMAN SÜRESİNCE
; RESET                             ;SAYMAYA DEVAM EDER VE BELLİ BİR
        CLRF    DGECIK1             ;GECİKME İLE DD REG. 0. BİTİNİ
        CLRF    DGECIK2             ;SET EDER. DOLAYISIYLA YUKARIDAKİ
        BCF     DD,0                ;ZAMAN GECİKMESİ BU BÖLÜMDEKİNDEN
        GOTO    DGECIK_CIKIS        ;FAZLA OLMALIDIR.
DGECIK_SAY                          ;       "
        BTFSC   DD,0                ;       "
        GOTO    DGECIK_CIKIS        ;       "
        MOVLW H'23'                 ;       "
        SUBWF DGECIK2,W             ;       "
        BTFSS   STATUS,0            ;       "
        GOTO    DGECIK_CIKIS        ;       "
        MOVLW H'51'                 ;       "
        SUBWF DGECIK1,W             ;       "
        BTFSS   STATUS,0            ;       "
        GOTO    DGECIK_CIKIS        ;       "
        BSF     DD,0                ;       "
DGECIK_CIKIS                        ;       "
;************************************************
        BTFSC   EGECIK,1            ;BU BÖLÜM DD REG. 0. BİTİNİN
        GOTO    EGECIK_SAY          ;SET OLDUĞU İLK ANDA ÇIKIŞ VERİR
        BTFSC   EGECIK,0            ;YANİ EE,0=1 SET OLUR VEKISA BİR
        GOTO    EGECIK_SET          ;SÜRE BEKLETİR. BİZİM ASIL KULLANACAĞIMIZ
        BCF     STATUS,5            ;REGISTER EE REGISTERIDIR. BELLİ BİR SÜRE
        BTFSS   DD,0                ;SONRA EE,0=1 OLDUĞU DURUMDA CUBUK VE
        GOTO    EGECIK_SET          ;HALKA DURUMLARI KARŞILAŞTIRILACAK VE
EGECIK_RESET                        ;GEREKİYORSA ÜRETİM OLACAKTIR.
        CLRF    EGECIK1             ;BU KARŞILAŞTIRMA YAPILIRKEN ARKADAN GELEN
        CLRF    EGECIK2             ;CUBUK İÇİN SAYIM YENİDEN BAŞLATILABİLECEK
        BSF     EGECIK,1            ;VE BU SAYIM DİĞERİNİ ETKİLEMEYECEKTİR
        BCF     STATUS,5            ;       "
        BSF     EE,0                ;       "
EGECIK_SAY                          ;       "
        MOVLWH'01'                  ;       "
        SUBWF EGECIK2,W             ;       "
        BTFSS   STATUS,2            ;       "
        GOTO    EGECIK_SAY_KONTROL  ;       "
        MOVLWH'66'                  ;       "
        SUBWF EGECIK1,W             ;       "
EGECIK_SAY_KONTROL                  ;       "
        BTFSS   STATUS,0            ;       "
        GOTO    EGECIK_SET          ;       "
EGECIK_ITME                         ;       "
        BCF     EGECIK,1            ;       "
        BCF     STATUS,5            ;       "
        BCF     EE,0                ;       "
EGECIK_SET                          ;       "
        BTFSS   DD,0                ;       "
        GOTO    EGECIK_SIL          ;       "
```

```
        BSF     EGECIK,0                ;       "
        GOTO    EGECIK_GECIKME_CIKIS;           "
EGECIK_SIL                              ;       "
        BCF     EGECIK,0                ;       "
EGECIK_GECIKME_CIKIS                    ;       "
;*************************************************
BIRAKACAKMI?                            ;
        BTFSC   KENAR,0                 ;BİR CUBUK İÇİN ÜRETİM DEVAM EDİYOR MU?
        GOTO    RESET_PEG               ;EVET
        BTFSS   EE,0                    ;EE,0=1 Mİ? ZAMAN GECIKMESİ DOLDU MU?
        GOTO    RESET_PEG               ;HAYIR
        BTFSC   MCUBUK,1                ;GELEN İLK CUBUK METAL Mİ?
        GOTO    METAL_PEG               ;EVET İSE
        BTFSC   PCUBUK,1                ;HAYIR İSE PLASTİK Mİ?
        GOTO    PLASTIK_PEG             ;EVET
        GOTO    RESET_PEG               ;HAYIR
PLASTIK_PEG                             ;
        BTFSC   PRING,1                 ;TOPLAMA BÖL.'DE İLK BÖLÜMDE PLASTİK HALKA MI VAR?
        GOTO    PRPP_URET               ;PLASTİK HALKA YOK İSE METAL HALKAYA BAK
        BTFSC   MRING,1                 ;METAL HALKA MI VAR?
        GOTO    MRPP_URET               ;EVET
        GOTO    PKAYDIR                 ;HİÇ HALKA YOKSA MEVCUT CUBUĞU SİL
METAL_PEG                               ;
        BTFSC   KENAR,0                 ;BİR CUBUK İÇİN ÜRETİM DEVAM EDİYOR MU?
        GOTO    RESET_PEG               ;EVET
        BTFSC   PRING,1                 ;TOPLAMA BÖLGESİ İLK BÖLÜMDE PLASTİK HALKA MI VAR?
        GOTO    PRMP_URET               ;PLASTİK HALKA YOK İSE METAL HALKAYA BAK
        BTFSC   MRING,1                 ;METAL HALKA MI VAR?
        GOTO    MRMP_URET               ;EVET
        GOTO    PKAYDIR                 ;HİÇ HALKA YOKSA MEVCUT CUBUĞU SİL
;*************************************************
PRPP_URET                               ;
        CALL    GECIKME                 ;BİRAZ BEKLE
        CALL    GECIKME                 ;       "
        CALL    GECIKME                 ;       "
        CALL    GECIKME                 ;       "
        BTFSC   PORTB,1                 ;ÜRETİLMİŞ OLAN HALKA BEKLİYOR MU?
        GOTO    RESET_PEG               ;EVET
        MOVLWH'03'                      ;PRPP YERLİ SAYIDA ÜRETİLMİŞ Mİ?
        SUBWF   PRPP,0                  ;       "
        BTFSC   STATUS,0                ;       "
        GOTO    PKAYDIR                 ;SAYI YETERLİ İSE MEVCUT ÇUBUĞU SİL
        BSF     PEG,1                   ;SAYI YERLİ DEĞİLSE ÜRETME EMRİ VER
        BSF     KENAR,0                 ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        INCF    PRPP,1                  ;PRPP=PRPP+1
        GOTO    KAYDIR                  ;HEM HALKALARI HEMDE CUBUKLARI KAYDIR
;*********************************
MRPP_URET                               ;
        CALL    GECIKME                 ;BİRAZ BEKLE
        CALL    GECIKME                 ;       "
        CALL    GECIKME                 ;       "
        CALL    GECIKME                 ;       "
        BTFSC   PORTB,1                 ;ÜRETİLMİŞ OLAN BİR HALKA BEKLİYOR MU?
        GOTO    RESET_PEG               ;EVET
        MOVLWH'02'                      ;MRPP YERLİ SAYIDA ÜRETİLMİŞ Mİ?
        SUBWF   MRPP,0                  ;       "
        BTFSC   STATUS,0                ;       "
        GOTO    PKAYDIR                 ;SAYI YETERLİ İSE MEVCUT ÇUBUĞU SİL
        BSF     PEG,1                   ;SAYI YERLİ DEĞİLSE ÜRETME EMRİ VER
        BSF     KENAR,0                 ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        INCF    MRPP,1                  ;MRPP=MRPP+1
        GOTO    KAYDIR                  ;HEM HALKALARI HEMDE CUBUKLARI KAYDIR
;*********************************
PRMP_URET                               ;
        CALL    GECIKME                 ;BİRAZ BEKLE
        CALL    GECIKME                 ;       "
        CALL    GECIKME                 ;       "
        CALL    GECIKME                 ;       "
        BTFSC   PORTB,1                 ;ÜRETİLMİŞ OLAN BİR HALKA BEKLİYOR MU?
```

```
        GOTO   RESET_PEG          ;EVET
        MOVLWH'02'                ;PRMP YERLİ SAYIDA ÜRETİLMİŞ Mİ?
        SUBWF  PRMP,0             ;       "
        BTFSC  STATUS,0           ;       "
        GOTO   PKAYDIR            ;SAYI YETERLİ İSE MEVCUT ÇUBUĞU SİL
        BSF    PEG,1              ;SAYI YERLİ DEĞİLSE ÜRETME EMRİ VER
        BSF    KENAR,0            ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        INCF   PRMP,1             ;PRMP=PRMP+1
        GOTO   KAYDIR             ;HEM HALKALARI HEMDE CUBUKLARI KAYDIR
;*********************************
;
MRMP_URET                         ;
        CALL   GECIKME            ;BİRAZ BEKLE
        CALL   GECIKME            ;       "
        CALL   GECIKME            ;       "
        CALL   GECIKME            ;       "
        BTFSC  PORTB,1            ;ÜRETİLMİŞ OLAN BİR HALKA BEKLİYOR MU?
        GOTO   RESET_PEG          ;EVET
        MOVLWH'03'                ;MRMP YERLİ SAYIDA ÜRETİLMİŞ Mİ?
        SUBWF  MRMP,0             ;       "
        BTFSC  STATUS,0           ;       "
        GOTO   PKAYDIR            ;SAYI YETERLİ İSE MEVCUT ÇUBUĞU SİL
        BSF    PEG,1              ;SAYI YERLİ DEĞİLSE ÜRETME EMRİ VER
        BSF    KENAR,0            ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        INCF   MRMP,1             ;MRMP=MRMP+1
        GOTO   KAYDIR             ;HEM HALKALARI HEMDE CUBUKLARI KAYDIR
;*********************************
KAYDIR                            ;
        CALL   GECIKME            ;BİRAZ BEKLE
        CALL   GECIKME            ;       "
        MOVLWB'00111110'          ;W <-- B'00111110'
        RRF    HALKA,F            ;HALKA REG. BİTLERİNİ BİRER SAĞA KAYDIR
        RRF    PRING,F            ;PRING REG      "
        RRF    MRING,F            ;MRING REG      "
        ANDWF HALKA,F             ;HALKA REG. KULLANILMAYAN BİTLERİNİ SİL
        ANDWF PRING,F             ;PRING REG      "
        ANDWF MRING,F             ;MRING REG      "
        RRF    CUBUK,F            ;CUBUK REG. BİTLERİNİ BİRER SAĞA KAYDIR
        RRF    PCUBUK,F           ;PCUBUK REG     "
        RRF    MCUBUK,F           ;MCUBUK REG     "
        ANDWF CUBUK,F             ;CUBUK REG. KULLANILMAYAN BİTLERİNİ SİL
        ANDWF PCUBUK,F            ;PCUBUK REG     "
        ANDWF MCUBUK,F            ;MCUBUK REG     "
        GOTO   BIRAK_CIKIS        ;BIRAK_CIKIS ETİKETİNE GİT
;*********************************
PKAYDIR                           ;
        CALL   GECIKME            ;KULLANILMAYACAK OLAN CUBUK VARSA
        CALL   GECIKME            ;BİRAZ BEKLE
        CALL   GECIKME            ;       "
        CALL   GECIKME            ;       "
        CALL   GECIKME            ;       "
        CALL   GECIKME            ;       "
        CALL   GECIKME            ;       "
        CALL   GECIKME            ;       "
        CALL   GECIKME            ;       "
        MOVLW B'00001110'         ;W <--  B'00001110'
        RRF    CUBUK,F            ;CUBUK REG. BİTLERİNİ BİRER SAĞA KAYDIR
        RRF    PCUBUK,F           ;PCUBUK REG     "
        RRF    MCUBUK,F           ;MCUBUK REG     "
        ANDWF CUBUK,F             ;CUBUK REG. KULLANILMAYAN BİTLERİNİ SİL
        ANDWF PCUBUK,F            ;PCUBUK REG     "
        ANDWF MCUBUK,F            ;MCUBUK REG     "
        GOTO   METAL_PEG_SIL      ;METAL_PEG_SIL ETİKETİNE GİT
;*********************************
PEG_SET                           ;
        BSF    KENAR,0            ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        GOTO   BIRAK_CIKIS        ;
RESET_PEG                         ;
        BCF    PEG,1              ;ÜRETME EMRİ VERME
METAL_PEG_SIL                     ;
```

```
        BTFSC  PORTB,5              ;GELEN PEG AYIRMA BÖLGESİNİ TERKETMİŞSE
        GOTO   BIRAK_CIKIS          ;
        CALL   GECIKME              ;BİRAZ BEKLE
        BCF    PLASTIK,0            ;PLASTİK,0=0
        BCF    METAL,2              ;METAL,2=0
        BCF    KENAR,0              ;KENAR,0=0
        BCF    MARKER,0             ;MARKER,0=0
BIRAK_CIKIS                         ;
;*************************************************
ITICI_GECIKME                       ;BU BÖLÜM İTİLECEK OLAN HALKA İÇİN
        BTFSC  ITICI,1              ;İTME SÜRESİNİ TAYİN EDER.
        GOTO   ITICI_SAY            ;EĞER İTME EMRİ (RING,1=1) VARSA
        BTFSC  ITICI,0              ;O ANDA PARÇA İTİLİR VE SELENOİD İTİLİ
        GOTO   ITICI_SET            ;OLARAK BİRAZ BEKLER SONRA GERİ GELİR
        BCF    STATUS,5             ;İTME İÇİN PORTA,0=1 OLMALIDIR
        BTFSS  RING,1               ;       "
        GOTO   ITICI_SET            ;       "
ITICI_RESET                         ;       "
        CLRF   ITICI1               ;       "
        CLRF   ITICI2               ;       "
        BSF    ITICI,1              ;       "
        BSF    PORTA,0              ;       "
ITICI_SAY                           ;       "
        MOVLWH'05'                  ;       "
        SUBWF  ITICI2,W             ;       "
        BTFSS  STATUS,2             ;       "
        GOTO   ITICI_SAY_KONTROL    ;       "
        MOVLWH'66'                  ;       "
        SUBWF  ITICI1,W             ;       "
ITICI_SAY_KONTROL                   ;       "
        BTFSS  STATUS,0             ;       "
        GOTO   ITICI_SET            ;       "
ITICI_ITME                          ;       "
        BCF    ITICI,1              ;       "
        BCF    STATUS,5             ;       "
        BCF    PORTA,0              ;       "
ITICI_SET                           ;       "
        BCF    STATUS,5             ;       "
        BTFSS  RING,1               ;       "
        GOTO   ITICI_SIL            ;       "
        BSF    ITICI,0              ;       "
        GOTO   ITICI_GECIKME_CIKIS  ;       "
ITICI_SIL                           ;       "
        BCF    ITICI,0              ;       "
ITICI_GECIKME_CIKIS                 ;       "
;*************************************************
;BIRAKTIKTAN SONRA BEKLE            ;
        BTFSC  BIRAK,1              ;BU BÖLÜM HALKA BIRAKMA YADA ÜRETİM
        GOTO   BIRAK_SAY            ;EMRİ VERİLDİĞİNDE (PEG,1=1) OLDUĞUNDA
        BTFSC  BIRAK,0              ;HEMEN HALKAYI BIRAKTIRIR (PORTA,1=1)
        GOTO   BIRAK_SET            ;VE PORTA,1 E BAĞLI BIRAKMA SELENOİDİNİN
        BCF    STATUS,5             ;BELLİ BİR SÜRE ÇALIŞMASINI SAĞLAR.
        BTFSS  PEG,1                ;KESME GECİKMESİ BÖLÜMÜDÜR.
        GOTO   BIRAK_SET            ;       "
BIRAK_RESET                         ;       "
        CLRF   BIRAK1               ;       "
        CLRF   BIRAK2               ;       "
        BSF    BIRAK,1              ;       "
        BCF    STATUS,5             ;       "
        BSF    PORTA,1              ;       "
BIRAK_SAY                           ;       "
        MOVLWH'10'                  ;       "
        SUBWF  BIRAK2,W             ;       "
        BTFSS  STATUS,2             ;       "
        GOTO   BIRAK_SAY_KONTROL    ;       "
        MOVLWH'66'                  ;       "
        SUBWF  BIRAK1,W             ;       "
BIRAK_SAY_KONTROL                   ;       "
        BTFSS  STATUS,0             ;       "
```

```
        GOTO    BIRAK_SET               ;       "
BIRAK_ITME                              ;       "
        BCF     BIRAK,1                 ;       "
        BCF     STATUS,5                ;       "
        BCF     PORTA,1                 ;       "
BIRAK_SET                               ;       "
        BCF     STATUS,5                ;       "
        BTFSS   PEG,1                   ;       "
        GOTO    BIRAK_SIL               ;       "
        BSF     BIRAK,0                 ;       "
        GOTO    BIRAK_GECIKME_CIKIS     ;       "
BIRAK_SIL                               ;       "
        BCF     BIRAK,0                 ;       "
BIRAK_GECIKME_CIKIS                     ;       "
;*************************************************
BOZUK_KONTROL1                          ;BOZUK,0. BİT 0 (BOZUK PARÇA YOK) MI?
        BTFSC   BOZUK,0                 ;HAYIR (BOZUK,0=1) İSE HAFIZADA TUT
        GOTO    BOZUK_KONTROL4          ;EVET (BOZUK,0=0 İSE) TEKRAR KONTROL ETMEYE BASLA
        BTFSS   PORTB,4                 ;PORTB,4 = 1 Mİ? (BOZUK BÖLGE ALT SENSÖR)
        GOTO    BOZUK_ITME              ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL2                          ;
        BTFSC   PORTB,6                 ;PORTB,6 = 0 MI? (BOZUK BÖLGE ÜST SENSÖR)
        GOTO    BOZUK_ITME              ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL3                          ;
        BSF     BOZUK,0                 ;BOZUK,0 = 1 YAP (BOZUK PARÇAYI HAFIZAYA AL)
        GOTO    BOZUK_KONTROL4          ;BOZUK_KONTROL4 ETİKETİNE GİT
BOZUK_ITME                              ;
        BCF     BOZUK,0                 ;HAFIZAYI SİL
BOZUK_KONTROL4                          ;
        BTFSS   BOZUK,0                 ;BOZUK PARÇA ALGILANMIŞ MI?
        GOTO    BOZUK_KONTROL7          ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL5                          ;
        BTFSS   PORTB,3                 ;BOZUK PARÇA AYIRICI SELENOİD YANINDA MI?
        GOTO    BOZUK_KONTROL7          ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL6                          ;
        BSF     BOZUK,1                 ;BOZUK PARÇAYI İTME EMRİ VER (BOZUK,1=1)
        BCF     BOZUK,0                 ;HAFİZAYI SİL
        GOTO    BOZUK_CIKIS             ;BOZUK_CIKIS ETİKETİNE GİT
BOZUK_KONTROL7                          ;
        BCF     BOZUK,1                 ;BOZUK PARÇA İTME EMRİ VERME (BOZUK,1=0)
BOZUK_CIKIS                             ;
;*************************************************
BOZUK_GECIKME                           ;
        BTFSC   BOZ,1                   ; BU BÖLÜM BOZUK PARÇA İTEN SELENOİD(PORTA,2) İÇİN
        GOTO    BOZUK_SAY               ; KESME GECİKMESİ SAĞLAR.
        BTFSC   BOZ,0                   ; İTME EMRİ (BOZUK,1=1) ALGILANDIĞI ANDA PORTA,2=1
        GOTO    BOZUK_SET               ; OLUR PARÇA İTİLİR.
        BCF     STATUS,5                ; İTTİKTEN SONRA BİR SÜRE BEKLENMESİ SAĞLANIR
        BTFSS   BOZUK,1                 ;       "
        GOTO    BOZUK_SET               ;       "
BOZUK_RESET                             ;       "
        CLRF    BOZ1                    ;       "
        CLRF    BOZ2                    ;       "
        BSF     BOZ,1                   ;       "
        BSF     PORTA,2                 ;       "
BOZUK_SAY                               ;       "
        MOVLWH'05'                      ;       "
        SUBWF   BOZ2,W                  ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    BOZUK_SAY_KONTROL       ;       "
        MOVLWH'66'                      ;       "
        SUBWF   BOZ1,W                  ;       "
BOZUK_SAY_KONTROL                       ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BOZUK_SET               ;       "
BOZU_ITME                               ;       "
        BCF     BOZ,1                   ;       "
        BCF     STATUS,5                ;       "
        BCF     PORTA,2                 ;       "
```

```
BOZUK_SET                       ;       "
        BCF     STATUS,5        ;       "
        BTFSS   BOZUK,1         ;       "
        GOTO    BOZUK_SIL       ;       "
        BSF     BOZ,0           ;       "
        GOTO    BOZUK_GECIKME_CIKIS ;   "
BOZUK_SIL                       ;       "
        BCF     BOZ,0           ;       "
BOZUK_GECIKME_CIKIS             ;       "
        GOTO    START           ;START ETİKETİNE GİT
;*********************************************************
INT_ALT_PROG                    ;KESME ALT PROGRAMI BAŞLANGICI
        MOVWF SAKLA_W           ;W --> SAKLA_W
        SWAPF  STATUS,W         ;STATUS REG. SWAP YAP VE W YE YÜKLE
        MOVWF SAKLA_S           ;W --> SAKLA_S
        BCF     INTCON,2        ;TOIF <-- '0' (zaman aşımı yok)
        INCFSZ ITICI1,F         ;ITICI1 = ITICI1+1, ITICI,1 = 0 MI?
        GOTO    CIKIS1          ;HAYIR İSE
        INCF    ITICI2,F        ;EVET İSE ITICI2 = ITICI2+1
CIKIS1    .                     ;
        INCFSZ  AGECIK1,F       ;KESME ALT PROGRAMINDA BULUNAN BÜTÜN
        GOTO    CIKIS2          ; REGİSTERLER İÇİN AYNI İŞLEMİ)
        INCF    AGECIK2,F       ; (İÇERİK ARTIRMA TEKRARLA "
CIKIS2                          ;       "
        INCFSZ  BOZ1,F          ;       "
        GOTO    CIKIS3          ;       "
        INCF    BOZ2,F          ;       "
CIKIS3                          ;       "
        INCFSZ  BIRAK1,F        ;       "
        GOTO    CIKIS4          ;       "
        INCF    BIRAK2,F        ;       "
CIKIS4                          ;       "
        INCFSZ  BGECIK1,F       ;       "
        GOTO    CIKIS5          ;       "
        INCF    BGECIK2,F       ;       "
CIKIS5                          ;       "
        INCFSZ  CGECIK1,F       ;       "
        GOTO    CIKIS6          ;       "
        INCF    CGECIK2,F       ;       "
CIKIS6                          ;       "
        INCFSZ  DGECIK1,F       ;       "
        GOTO    CIKIS7          ;       "
        INCF    DGECIK2,F       ;       "
CIKIS7                          ;       "
        INCFSZ  EGECIK1,F       ;       "
        GOTO    CIKIS8          ;       "
        INCF    EGECIK2,F       ;       "
CIKIS8                          ;       "
        MOVLW H'00'             ;TMRO I SIFIRLA
        MOVWF TMR0              ;       "
        SWAPF  SAKLA_S,W        ;W VE STATUS REG. TEKRAR YÜKLE
        MOVWF STATUS            ;       "
        SWAPF  SAKLA_W,F        ;       "
        SWAPF  SAKLA_W,W        ;       "
        RETFIE                  ;ANA PROGRAMA GERİDÖN
;*********************************************************
GECIKME                         ;ÇİFT DÖNGÜLÜ ZAMAN
        MOVLW H'11'             ;GECİKMESİ ALT
        MOVWF SAY1              ;PROGRAMI
DONGU1                          ;       "
        MOVLW H'FF'             ;       "
        MOVWF SAY2              ;       "
DONGU2                          ;       "
        DECFSZ SAY2,1           ;       "
        GOTO    DONGU2          ;       "
        DECFSZ SAY1,1           ;       "
        GOTO    DONGU1          ;       "
        RETURN                  ;       "
        END                     ;PROGRAM SONU
```

EK-A.5 Toplu Program

```
        LIST    P=16F84A                ;16F84A YI TANIT
        INCLUDE "P16F84A.INC"           ;16F84A.INC DOSYASINI EKLE
        __CONFIG _CP_OFF & _PWRTE_ON & _WDT_OFF & _XT_OSC
        ORG     H'00'                   ;PROGRAM BAŞLANGIC ADRESİ H'00'
        GOTO    BASLA                   ;BASLA ETİKETİNE GİT
        ORG     H'04'                   ;İNTERRUPT(KESME SİNYALİ GELİRSE)
        GOTO    INT_ALT_PROG            ;INT_ALT_PROG ETİKETİNE GİT
BASLA                                   ;
ITICI           EQU     H'0C'           ;FİLE REGISTERLERI TANIMLA
ITICI1          EQU     H'0D'           ;       "
ITICI2          EQU     H'0E'           ;       "
SAKLA_W         EQU     H'0F'           ;       "
SAKLA_S         EQU     H'10'           ;       "
RING            EQU     H'11'           ;       "
MRING           EQU     H'12'           ;       "
PRING           EQU     H'13'           ;       "
MR              EQU     H'14'           ;       "
PR              EQU     H'15'           ;       "
METAL           EQU     H'16'           ;       "
MRMP            EQU     H'17'           ;       "
MRPP            EQU     H'18'           ;       "
PRPP            EQU     H'19'           ;       "
PRMP            EQU     H'1A'           ;       "
BIRAK           EQU     H'1B'           ;       "
BIRAK1          EQU     H'1C'           ;       "
BIRAK2          EQU     H'1D'           ;       "
PEG             EQU     H'1E'           ;       "
BOZ             EQU     H'1F'           ;       "
BOZ1            EQU     H'20'           ;       "
BOZ2            EQU     H'21'           ;       "
BOZUK           EQU     H'22'           ;       "
KENAR           EQU     H'23'           ;       "
CUBUK           EQU     H'24'           ;       "
PCUBUK          EQU     H'25'           ;       "
MCUBUK          EQU     H'26'           ;       "
MARKER          EQU     H'27'           ;       "
TIMER           EQU     H'28'           ;       "
HALKA           EQU     H'29'           ;       "
MHALKA          EQU     H'2A'           ;       "
PHALKA          EQU     H'2B'           ;       "
URET            EQU     H'2C'           ;       "
SAYAC           EQU     H'2D'           ;       "
SAY             EQU     H'2E'           ;       "
SAY1            EQU     H'2F'           ;       "
SAY2            EQU     H'30'           ;       "
PLASTIK         EQU     H'31'           ;       "
AGECIK          EQU     H'32'           ;       "
AGECIK1         EQU     H'33'           ;       "
AGECIK2         EQU     H'34'           ;       "
BGECIK          EQU     H'35'           ;       "
BGECIK1         EQU     H'36'           ;       "
BGECIK2         EQU     H'37'           ;       "
CGECIK          EQU     H'38'           ;       "
CGECIK1         EQU     H'39'           ;       "
CGECIK2         EQU     H'3A'           ;       "
DGECIK          EQU     H'3B'           ;       "
DGECIK1         EQU     H'3C'           ;       "
DGECIK2         EQU     H'3D'           ;       "
EGECIK          EQU     H'3E'           ;       "
EGECIK1         EQU     H'3F'           ;       "
EGECIK2         EQU     H'40'           ;       "
AA              EQU     H'41'           ;       "
BB              EQU     H'42'           ;       "
CC              EQU     H'43'           ;       "
DD              EQU     H'44'           ;       "
EE              EQU     H'45'           ;       "
```

```
METALR          EQU     H'46'           ;           "
        BCF     STATUS,5                ;BANK0'A GEÇ
        CLRF    PORTB                   ;PORTB Yİ SIFIRLA
        CLRF    PORTA                   ;PORTAYI SIFIRLA
        BSF     STATUS,5                ;BANK1 E GEÇ
        MOVLW B'11111111'               ;W REGİSTERİNE H'FF' SAYISINI YÜKLE
        MOVWF TRISB                     ;W REGİSTERİ--> TRİSB'YE YÜKLE (PORTB'Yİ GİRİŞ YAP)
        MOVLW B'00010000'               ;W <-- B'00010000'
        MOVWF TRISA                     ;W --> TRISA (PORTA YI 5. BİT HARİÇ ÇIKIŞ YAP)
        CLRWDT                          ;WDT Yİ SİL
        MOVLW B'10001111'               ;W <-- B'10001111'
        MOVWF OPTION_REG                ;W --> OPTION_REG
        BCF     STATUS,5                ;BANK0 A GEÇ
        MOVLW H'00'                     ;W <-- H'00'
        MOVWFTMR0                       ;W --> TMR0
        MOVLW B'10100000'               ;W <-- B'10100000'
        MOVWF INTCON                    ;W --> INTCON
        CLRF    PORTB                   ;PORTB Yİ SIFIRLA
        CLRF    PORTA                   ;PORTA YI SIFIRLA
        CLRF    ITICI                   ;DİĞER FILE REGISTERLERİ SİL
        CLRF    RING                    ;           "
        CLRF    MRING                   ;           "
        CLRF    PRING                   ;           "
        CLRF    HALKA                   ;           "
        CLRF    PEG                     ;           "
        CLRF    BOZUK                   ;           "
        CLRF    KENAR                   ;           "
        CLRF    METAL                   ;           "
        CLRF    TIMER                   ;           "
        CLRF    MHALKA                  ;           "
        CLRF    PHALKA                  ;           "
        CLRF    URET                    ;           "
        CLRF    PLASTIK                 ;           "
        CLRF    KENAR                   ;           "
        CLRF    CUBUK                   ;           "
        CLRF    MCUBUK                  ;           "
        CLRF    PCUBUK                  ;           "
        CLRF    MARKER                  ;           "
        CLRF    SAY1                    ;           "
        CLRF    SAY2                    ;           "
        CLRF    SAY                     ;           "
        CLRF    MR                      ;           "
        CLRF    PR                      ;           "
        CLRF    PRPP                    ;           "
        CLRF    MRMP                    ;           "
        CLRF    PRMP                    ;           "
        CLRF    MRPP                    ;           "
        CLRF    AA                      ;           "
        CLRF    BB                      ;           "
        CLRF    CC                      ;           "
        CLRF    DD                      ;           "
        CLRF    EE                      ;           "
        CLRF    SAYAC                   ;           "
        CLRF    BOZUK                   ;           "
        CLRF    METALR                  ;           "
        CLRF    BIRAK                   ;           "
        BSF     PORTA,3                 ;PORTA 3. BİTİNİ 1 YAP ( MOTORLAR ÇALIŞSIN )
;*********************************
        GOTO    METAL_TANIMLA           ;
;*********************************
;%%%%%%%%%%%%%%% BİRİNCİ PROGRAM %%%%%%%%%%%%%%%%%%%%
START1
;*********************************
        MOVLW H'01'                     ;W = H'01'
        SUBWF SAYAC,0                   ;SAYAC = SAYAC-W
        BTFSS   STATUS,0                ;CF=1 Mİ? (SAYAC>=1 Mİ)
        GOTO    BIRAKMA                 ;HAYIR(TOPLAMA BÖLGESİ BOŞ)İSE
;PORTB_KONTROL1                         ;
        BTFSC   PORTB,1                 ;PORTB 1.BİT 0 MI?
```

```
        GOTO    BIRAKMA             ;HAYIR İSE
;BIRAK                               ;
        CALL    GECIKME1            ;BIRAKMADAN ÖNCE BEKLE
        BSF     PORTA,1             ;EVET İSE, PORTA 1.BİTİNİ 1 YAP (BIRAK)
        CALL    GECIKME1            ;BIRAKTIKTAN SONRA BEKLE
        DECF    SAYAC,F             ;SAYAC = SAYAC-1 (SAYACI BİR AZALT)
        GOTO    KONTROL11_CIKIS     ;KONTROL11_CIKIS ETİKETİNE GİT
BIRAKMA                             ;
        BCF     PORTA,1             ;PORTA,1 = 0 YAP (BIRAKMA)
KONTROL11_CIKIS                     ;
;*********************************
        MOVLW H'05'                 ;W = H'05'
        SUBWF SAYAC,0               ;SAYAC = SAYAC-W
        BTFSC STATUS,0              ;CF=0 MI? (SAYAC<5 Mİ?)
        GOTO  ITME1                 ;HAYIR(TOPLAMA BÖLGESİNDE 5 HALKA VAR) İSE
PORTB_KONTROL2                      ;
        BTFSS PORTB,2               ;PORTB 2.BİT 1(AYIRMA BÖLGESİNDE PARÇA VAR) Mİ?
        GOTO  ITME1                 ;HAYIR İSE
PORTB_KONTROL3                      ;
        BTFSC PORTB,5               ;PORTB 5.BİT 0 (BU PARÇA HALKA MI) MI?
        GOTO  ITME1                 ;HAYIR İSE
IT                                  ;
        BSF     PORTA,0             ;EVET İSE, PORTB 0. BİTİNİ 1 YAP (İT)
        INCF    SAYAC,F             ;SAYAC = SAYAC+1(SAYACI BİR ARTIR)
        CALL    GECIKME1            ;İTTİKTEN SONRA BEKLE
        GOTO    KONTROL2_CIKIS      ;KONTROL2_CIKIS ETİKETİNE GİT
ITME1                               ;
        BCF     PORTA,0             ;PORTA,0 = 0 YAP (İTME)
KONTROL2_CIKIS                      ;
;*********************************
        GOTO    BOZUK_KONTROL1      ;BASLA ETİKETİNE GİT (PROGRAM BAŞINA)
;*********************************
;%%%%%%%%%%%%%%%% İKİNCİ PROGRAM %%%%%%%%%%%%%%%%%%%%
START2
;*********************************
ITICI_KONTROL1
;*********************************
        BTFSC   RING,1              ;RING,1 = 0 MI?
        GOTO    ITME                ;HAYIR
        BTFSC   PORTA,0             ;PORTA,0 = 0 MI?
        GOTO    ITME                ;HAYIR
        BTFSC   ITICI,1             ;ITICI,1 = 0 MI?
        GOTO    ITME                ;HAYIR
        MOVLW H'05'                 ;W <-- H'05'
        SUBWF SAYAC,W               ;SAYAC = SAYAC-W
        BTFSC STATUS,0              ;CF=0 MI? (SAYAC<5 Mİ?)
        GOTO  ITME                  ;TOPLAMA BÖLGESİ DOLU İSE
MR_IT                               ;
        BTFSS   PORTB,2             ;AYIRICI BÖLGEYE PARÇA GELDİ Mİ?
        GOTO    ITME                ;HAYIR
        BTFSC   PORTB,5             ;EVET İSE, BU PARÇA HALKA MI?
        GOTO    ITME                ;HAYIR
        BTFSS   METAL,1             ;HALKA METAL Mİ?
        GOTO    PR_IT1              ;METAL DEĞİL İSE PR_IT1 ETİKETİNE GİT
        BTFSC   PRING,1             ;PRING,1 = 0 MI?
        GOTO    ITME                ;HAYIR
MRING_SET1                          ;
        BSF     RING,1              ;RING,1 = 1
        BCF     METAL,0             ;METAL,0 = 0
        INCF    METALR,1            ;METALR = METALR + 1
        INCF    SAYAC,1             ;SAYAC = SAYAC + 1
        CALL    GECIKME2            ;BİRAZ BEKLE
        GOTO    ITME_CIKIS1         ;
;*************************************************************
PR_IT1                              ;
        BTFSS   PLASTIK,0           ;PLASTIK,0 = 1 Mİ? (HALKA PLASTİK Mİ?)
        GOTO    ITME                ;HAYIR
        MOVLW H'01'                 ;W <-- H'00'
        SUBWF METALR,W              ;SAYAC = METALR - W
```

```
        BTFSS   STATUS,0            ;ZF=0 MI? (METALR = 0 MI?)
        GOTO    ITME                ;METALR = 0 İSE
        BTFSS   PORTB,2             ;METALR = 0 DEĞİLSE
        GOTO    ITME                ;ITME ETİKETİNE GİT
        BTFSC   PORTB,5             ;PARÇA HALKA MI?
        GOTO    ITME                ;ÇUBUK İSE
;PRING_SET                          ;
        CALL    GECIKME2            ;İTMEDEN ÖNCE BEKLE
        BSF     RING,1              ;HALKA İSE, PORTB 0. BİTİNİ 1 YAP (İT)
        INCF    SAYAC,1             ;SAYAC = SAYAC+1
        DECFSZ METALR,1             ;METALR=METALR-1 , METALR=0 MI?
        GOTO    SET_PRING           ;METALR=0 DEĞİL İSE SET_PRING ETİKETİNE GİT
RESET_PRING                         ;
        BCF     PRING,1             ;PRING,1 = 0 YAP
        GOTO    ITME_CIKIS1         ;
SET_PRING                           ;
        BSF     PRING,1             ; PRING,1 = 1 YAP
        GOTO    ITME_CIKIS1         ;
;*********************************
;
ITME                                ;
        BTFSC   PORTB,2             ;PORTB,2 = 0 MI?
        GOTO    ITME_CIKIS1         ;HAYIR
        BTFSC   PORTB,5             ;PORTB,5 = 0 MI?
        GOTO    ITME_CIKIS1         ;HAYIR
        CALL    GECIKME2            ;BİRAZ BEKLE
        BCF     METAL,1             ;METAL,1 = 0
        BCF     PLASTIK,0           ;PLASTİK,0 = 0
        BCF     RING,1              ;RING,1 = 0
        BCF     METAL,2             ;
ITME_CIKIS1                         ;
;*********************************
;
BIRAK_KONTROL                       ;
        BTFSC   PORTA,1             ;BIRAKICI SELENOİD ÇALIŞIYORSA
        GOTO    RING_BIRAKMA        ;
        BTFSC   BB,0                ;BIRAKMA EMRİ VERİLMİŞ İSE
        GOTO    RING_BIRAKMA        ;
        BTFSC   AA,1                ;  "                "
        GOTO    RING_BIRAKMA        ;
        BTFSC   AGECIK,1            ;  "                "
        GOTO    RING_BIRAKMA        ;RING_BIRAKMA ETİKETİNE GİT
        MOVLWH'01'                  ;W <-- H'01'
        SUBWF   SAYAC,W             ;SAYAC = SAYAC-W
        BTFSS   STATUS,0            ;CF=1 Mİ? (SAYAC>=1 Mİ)
        GOTO    RING_BIRAKMA        ;HAYIR İSE
;PORTB_KONTROL1                     ;
        BTFSC   PORTB,1             ;PORTB 1.BİT 0 MI?
        GOTO    RING_BIRAKMA        ;HAYIR İSE
RING_BIRAK                          ;
        BSF     PEG,1               ;EVET İSE, PEG,1 = 1 YAP (BRAKMA EMRİ VER)
        DECF    SAYAC,F             ;SAYAC = SAYAC-1
        GOTO    KONTROL1_CIKIS      ;
RING_BIRAKMA                        ;
        BCF     PEG,1               ;BIRAKMA EMRİ VERME
KONTROL1_CIKIS                      ;
;*********************************
;
;BIRAK_GECIKME                      ;
;************************************************************
;
        BTFSC   AGECIK,1            ; HALKA BIRAKICI SELENOİD İÇİN
        GOTO    AGECIK_SAY1         ; KESME GECİKMESİ BÖLÜMLERİ BAŞLANGICI
        BTFSC   AGECIK,0            ;       "
        GOTO    AGECIK_SET1         ; BU BÖLÜMDE BIRAKMA EMRİ (PEG,1=1) VARSA
        BCF     STATUS,5            ; AA REGİSTERİNİN 0. BİTİ HEMEN 1 OLUR
        BTFSS   PEG,1               ; VE GECIKME BİTİNCE 0 OLUR
        GOTO    AGECIK_SET1         ;       "
        CLRF    AGECIK1             ;       "
        CLRF    AGECIK2             ;       "
        BSF     AGECIK,1            ;       "
        BSF     AA,0                ;       "
AGECIK_SAY1                         ;       "
```

```
        MOVLW H'16'                 ;       "
        SUBWF AGECIK2,W             ;       "
        BTFSS STATUS,2              ;       "
        GOTO  AGECIK_SAY_KONT1      ;       "
        MOVLW H'76'                 ;       "
        SUBWF AGECIK1,W             ;       "
AGECIK_SAY_KONT1                    ;       "
        BTFSS STATUS,0              ;       "
        GOTO  AGECIK_SET1           ;       "
        BCF   AGECIK,1              ;       "
        BCF   STATUS,5              ;       "
        BCF   AA,0                  ;       "
AGECIK_SET1                         ;       "
        BTFSS PEG,1                 ;       "
        GOTO  AGECIK_SIL1           ;       "
        BSF   AGECIK,0              ;       "
        GOTO  AGECIK_GECIK_CIK1     ;       "
AGECIK_SIL1                         ;       "
        BCF   AGECIK,0              ;       "
AGECIK_GECIK_CIK1                   ;       "
;*****************************************
;BIRAKMADAN ÖNCE BEKLE              ;
;*******************************    ; BU BÖLÜM AA REGİSTERİNİN 0. BİTİNİN 1 OLDUĞU
        BTFSC AA,0                  ; ZAMAN SÜRESİNCE SAYAR VE BELLİ BİR SÜRE SONRA
        GOTO  BGECIK_SAY1           ; BB,0 = 1 OLUR. YANİ YUKARIDAKİ BÖLÜMLE BİRLİKTE
        CLRF  BGECIK1               ; BIRAKICI SELENOİDİN BIRAKMADAN ÖNCE BEKLENMESİ
        CLRF  BGECIK2               ; BURADA SAĞLANIR. BURADA DİKKAT EDİLMESİGEREKEN
        BCF   BB,0                  ; NOKTA AA,0 =1 OLDUĞU ZAMAN SÜRESİNİN
        GOTO  BGECIK_CIKIS1         ; BU BÖLÜMDEKİ BEKLEME SÜRESİNDEN FAZLA OLMASI
BGECIK_SAY1                         ; GEREKTİĞİDİR
        BTFSC BB,0                  ;       "
        GOTO  BGECIK_CIKIS1         ;       "
        MOVLW H'16'                 ;       "
        SUBWF BGECIK2,W             ;       "
        BTFSS STATUS,0              ;       "
        GOTO  BGECIK_CIKIS1         ;       "
        MOVLW H'66'                 ;       "
        SUBWF BGECIK1,W             ;       "
        BTFSS STATUS,0              ;       "
        GOTO  BGECIK_CIKIS1         ;       "
        BSF   BB,0                  ;       "
BGECIK_CIKIS1                       ;       "
;*****************************************
        BTFSC BIRAK,1               ; BU BÖLÜM İSE BIRAKICI SELENOİDİN (PORTB,1)
        GOTO  BIRAK_SAY1            ; HEM ÇALIŞMASI HEMDE BELLİ BİR SÜRE ÇALIŞIR
HALDEBEKLEMESİ
        BTFSC BIRAK,0               ; İÇİN GEREKEN ZAMANI BİZE SAĞLAR
        GOTO  BIRAK_SET1            ; BAŞLANGIÇ ŞARTI (BB,0 =1, 1 PALS YETERLİDİR)
        BCF   STATUS,5              ; HEMEN PORTA,1=1 OLUR VE BELİLİ BİR
        BTFSS BB,0                  ; SÜRE BEKLEYEREK 0 OLUR      "
        GOTO  BIRAK_SET1            ;       "
        CLRF  BIRAK1                ;       "
        CLRF  BIRAK2                ;       "
        BSF   BIRAK,1               ;       "
        BCF   STATUS,5              ;       "
        BSF   PORTA,1               ;       "
BIRAK_SAY1                          ;       "
        MOVLWH'15'                  ;       "
        SUBWF BIRAK2,W              ;       "
        BTFSS STATUS,2              ;       "
        GOTO  BIRAK_SAY_KONT1       ;       "
        MOVLWH'66'                  ;       "
        SUBWF BIRAK1,W              ;       "
BIRAK_SAY_KONT1                     ;       "
        BTFSS STATUS,0              ;       "
        GOTO  BIRAK_SET1            ;       "
        BCF   BIRAK,1               ;       "
        BCF   STATUS,5              ;       "
        BCF   PORTA,1               ;       "
```

```
BIRAK_SET1                              ;       "
        BCF     STATUS,5                ;       "
        BTFSS   BB,0                    ;       "
        GOTO    BIRAK_SIL1              ;       "
        BSF     BIRAK,0                 ;       "
        GOTO    BIRAK_GECIK_CIK1        ;       "
BIRAK_SIL1                              ;       "
        BCF     BIRAK,0                 ;       "
BIRAK_GECIK_CIK1                        ;       "
;*****************************************
        GOTO    BOZUK_KONTROL1
;*********************************
;%%%%%%%%%%%%%%% ÜÇÜNCÜ PROGRAM %%%%%%%%%%%%%%%%%%%%
START3
;*********************************
ITICI_KONTROL2
;*****************************************
        BCF     STATUS,5                ; BANK0 A GEÇ
MR_IT                                   ;
        BTFSC   PORTB,5                 ;AYIRMA BÖLGESİNDE HALKA VARMI
        GOTO    ITME_CIKIS2             ;       "
        BTFSS   PORTB,2                 ;       "
        GOTO    RESET_RING1             ;HİÇ BİR PARÇA YOKSA
        BTFSS   METAL,1                 ;HALKA VAR, AMA METAL Mİ?
        GOTO    PR_IT2                  ;DEĞİL İSE PR_IT2 YE GİT
        BTFSC   MRING,0                 ;MRING,0 YOKSA
        GOTO    RESET_RING1             ;VARSA
        BTFSC   PRING,0                 ;PRING,0 YOKSA
        GOTO    RESET_RING1             ;VARSA
        MOVLWH'05'                      ;METAL HALKA YETERİ KADAR İTİLMİŞ Mİ?
        SUBWF   MR,0                    ;       "
        BTFSC   STATUS,0                ;       "
        GOTO    RESET_RING1             ;EVET İSE BİR DAHA İTME
MRING_SET2                              ;HAYIR İSE
        BSF     RING,1                  ;İTME EMRİ VER
        BSF     MRING,0                 ;MRİNG REG. 0.BİTİNİ SET ET
        INCF    MR,1                    ;MR=MR+1
        CALL    GECIKME2                ;BİRAZ BEKLE
        GOTO    ITME_CIKIS2             ;
;*****************************************
PR_IT2                                  ;
        BTFSS   PLASTIK,0               ;HALKA PLASTİK İSE
        GOTO    RESET_RING1             ;DEĞİLSE
        BTFSC   MRING,0                 ;MRING,0 YOKSA
        GOTO    RESET_RING1             ;VARSA
        BTFSC   PRING,0                 ;PRING,0 YOKSA
        GOTO    RESET_RING1             ;VARSA
        MOVLWH'05'                      ;PLASTIK HALKA YETERİ KADAR İTİLMİŞ Mİ?
        SUBWF   PR,0                    ;       "
        BTFSC   STATUS,0                ;       "
        GOTO    RESET_RING1             ;EVET İSE BİR DAHA İTME
;PRING_SET                              ;HAYIR İSE
        BSF     RING,1                  ;İTME EMRİ VER
        BSF     PRING,0                 ;PRİNG REG. 0.BİTİNİ SET ET
        INCF    PR,1                    ;PR=PR+1
        CALL    GECIKME2                ;BİRAZ BEKLE
        GOTO    ITME_CIKIS2             ;
;*****************************************
RESET_RING1                             ;
        BCF     RING,1                  ;İTME EMRİNİ GERİ ÇEK
;METAL_RING_SIL                         ;
        BTFSC   PORTB,5                 ;AYIRICI BÖLGEDE ÇUBUK YOKSA
        GOTO    ITME_CIKIS2             ;VARSA
        BTFSC   PORTB,2                 ;HALKA AYIRICI BÖLDEDEN GİTMİŞ İSE
        GOTO    ITME_CIKIS2             ;GİTMEMİŞSE
        BCF     METAL,1                 ;METAL,1=0 YAP
        CALL    GECIKME2                ;BİRAZ BEKLE
ITME_CIKIS2                             ;
;*****************************************
```

```
;BIRAKICI_KONTROL
;*********************************
        BTFSS   PORTB,5         ;CUBUK VAR MI?
        GOTO    RESET_PEG1      ;CUBUK YOKSA
        BTFSC   METAL,2         ;CUBUK PLASTİK Mİ?
        GOTO    METAL_PEG1      ;DEĞİLSE METAL ÇUBUĞA BAK
;PLASTIK_PEG                    ;
        BTFSS   PLASTIK,0       ;PLASTİK ÇUBUK VARSA
        GOTO    RESET_PEG1      ;YOKSA
        BTFSC   KENAR,0         ;BİR HALKA İÇİN ÜRETİM
        GOTO    RESET_PEG1      ;DEVAM EDİYORSA
        BTFSC   PRING,0         ;TOPLAMA BÖLGESİNDE PLASTİK HALKA VARMI?
        GOTO    PRPP_URET1      ;PLASTİK HALKA VAR İSE PRPP ÜRET
        BTFSC   MRING,0         ;TOPLAMA BÖLGESİNDE METAL HALKA VARMI?
        GOTO    MRPP_URET1      ;METAL HALKA VAR İSE MRPP ÜRET
        GOTO    RESET_PEG1      ;HİÇ BİR HALKA YOKSA
METAL_PEG1                      ;METAL CUBUK VARSA
        BTFSC   KENAR,0         ;BİR HALKA İÇİN ÜRETİM
        GOTO    RESET_PEG1      ;DEVAM EDİYORSA
        BTFSC   PRING,0         ;TOPLAMA BÖLGESİNDE PLASTİK HALKA VARMI?
        GOTO    PRMP_URET1      ;PLASTİK HALKA VAR İSE PRMP ÜRET
        BTFSC   MRING,0         ;TOPLAMA BÖLGESİNDE METAL HALKA VARMI?
        GOTO    MRMP_URET1      ;METAL HALKA VAR İSE MRMP ÜRET
        GOTO    RESET_PEG1      ;HİÇ BİR HALKA YOKSA
;*********************************************
;
PRPP_URET1                      ;PLASTİK HALKA+PLASTIK ÇUBUK ÜRETİMİ
        BTFSC   PORTB,1         ;ÜRETİLMİŞ BİR HALKA BEKLEMİYORSA
        GOTO    RESET_PEG1      ;BEKLİYORSA
        MOVLWH'03'              ;YETERLİ SAYIDA PRPP ÜRETİLMEMİŞSE
        SUBWF   PRPP,0          ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    RESET_PEG1      ;SAYI YETERLİ İSE
        BSF     PEG,1           ;ÜRETME EMRİ VER
        BSF     KENAR,0         ;
        INCF    PRPP,1          ;PRPP=PRPP+1
        BCF     PRING,0         ;TOPLAMA BÖLGESİNDEKİ PLASTİK HALKAYI SİL
        CALL    GECIKME2        ;BİRAZ BEKLE
        GOTO    BIRAK_CIKIS1    ;
;*********************************************
;
MRPP_URET1                      ;METAL HALKA+PLASTIK ÇUBUK ÜRETİMİ
        BTFSC   PORTB,1         ;ÜRETİLMİŞ BİR HALKA BEKLEMİYORSA
        GOTO    RESET_PEG1      ;BEKLİYORSA
        MOVLWH'02'              ;YETERLİ SAYIDA MRPP ÜRETİLMEMİŞSE
        SUBWF   MRPP,0          ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    RESET_PEG1      ;SAYI YETERLİ İSE
        BSF     PEG,1           ;ÜRETME EMRİ VER
        BSF     KENAR,0         ;
        INCF    MRPP,1          ;MRPP=MRPP+1
        BCF     MRING,0         ;TOPLAMA BÖLGESİNDEKİ PLASTİK HALKAYI SİL
        CALL    GECIKME2        ;BİRAZ BEKLE
        GOTO    BIRAK_CIKIS1    ;
;*********************************************
;
PRMP_URET1                      ;PLASTİK HALKA+METAL ÇUBUK ÜRETİMİ
        BTFSC   PORTB,1         ;ÜRETİLMİŞ BİR HALKA BEKLEMİYORSA
        GOTO    RESET_PEG1      ;BEKLİYORSA
        MOVLW H'02'             ;YETERLİ SAYIDA PRMP ÜRETİLMEMİŞSE
        SUBWF   PRMP,0          ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    RESET_PEG1      ;SAYI YETERLİ İSE
        BSF     PEG,1           ;ÜRETME EMRİ VER
        BSF     KENAR,0         ;
        INCF    PRMP,1          ;PRMP=PRMP+1
        BCF     PRING,0         ;TOPLAMA BÖLGESİNDEKİ PLASTİK HALKAYI SİL
        CALL    GECIKME2        ;BİRAZ BEKLE
        GOTO    BIRAK_CIKIS1    ;
;*********************************************
;
MRMP_URET1                      ;METAL HALKA+METAL ÇUBUK ÜRETİMİ
        BTFSC   PORTB,1         ;ÜRETİLMİŞ BİR HALKA BEKLEMİYORSA
```

```
        GOTO   RESET_PEG1           ;BEKLİYORSA
        MOVLW H'03'                 ;YETERLİ SAYIDA MRMP ÜRETİLMEMİŞSE
        SUBWF MRMP,0                ;
        BTFSC  STATUS,0             ;
        GOTO   RESET_PEG1           ;SAYI YETERLİ İSE
        BSF    PEG,1                ;ÜRETME EMRİ VER
        BSF    KENAR,0              ;
        INCF   MRMP,1               ;MRMP=MRMP+1
        BCF    MRING,0              ;TOPLAMA BÖLGESİNDEKİ PLASTİK HALKAYI SİL
        CALL   GECIKME2             ;BİRAZ BEKLE
        GOTO   BIRAK_CIKIS1         ;
;***********************************************
RESET_PEG1                          ;
        BCF    PEG,1                ;ÜRETME (HALKA BIRAKMA) EMRİNİ SİL
;METAL_PEG_SIL                      ;ÜRETİLEN BİR CUBUK AYIRICI
        BTFSC  PORTB,5              ;BÖLGEYİ TERKETMİŞ İSE
        GOTO   BIRAK_CIKIS1         ;       "
        BCF    METAL,2              ;METAL ÖZELLİĞİNİ SİL
        BCF    PLASTIK,0            ;PLASTİK ÖZELLİĞİNİ SİL
        BCF    KENAR,0              ;TEKRAR ÜRETİME İZİN VER
        CALL   GECIKME2             ;BİRAZ BEKLE
BIRAK_CIKIS1                        ;
;***********************************************
;BIRAK_GECIKME
;*********************************
        BTFSC  AGECIK,1             ; HALKA BIRAKICI SELENOİD İÇİN
        GOTO   AGECIK_SAY2          ; KESME GECİKMESİ BÖLÜMLERİ BAŞLANGICI
        BTFSC  AGECIK,0             ;       "
        GOTO   AGECIK_SET2          ; BU BÖLÜMDE BIRAKMA EMRİ (PEG,1=1) VARSA
        BCF    STATUS,5             ; AA REGİSTERİNİN 0. BİTİ HEMEN 1 OLUR
        BTFSS  PEG,1                ; VE GECIKME BİTİNCE 0 OLUR
        GOTO   AGECIK_SET2          ;       "
        CLRF   AGECIK1              ;       "
        CLRF   AGECIK2              ;       "
        BSF    AGECIK,1             ;       "
        BSF    AA,0                 ;       "
AGECIK_SAY2                         ;       "
        MOVLW H'26'                 ;       "
        SUBWF AGECIK2,W             ;       "
        BTFSS  STATUS,2             ;       "
        GOTO   AGECIK_SAY_KONT2     ;       "
        MOVLW H'76'                 ;       "
        SUBWF AGECIK1,W             ;       "
AGECIK_SAY_KONT2                    ;       "
        BTFSS  STATUS,0             ;       "
        GOTO   AGECIK_SET2          ;       "
        BCF    AGECIK,1             ;       "
        BCF    STATUS,5             ;       "
        BCF    AA,0                 ;       "
AGECIK_SET2                         ;       "
        BTFSS  PEG,1                ;       "
        GOTO   AGECIK_SIL2          ;       "
        BSF    AGECIK,0             ;       "
        GOTO   AGECIK_GECIK_CIK2    ;       "
AGECIK_SIL2                         ;       "
        BCF    AGECIK,0             ;       "
AGECIK_GECIK_CIK2                   ;       "
;*****************************************
;BIRAKMADAN ÖNCE BEKLE
;*********************************       ; BU BÖLÜM AA REGİSTERİNİN 0. BİTİNİN 1 OLDUĞU
        BTFSC  AA,0                 ; ZAMAN SÜRESİNCE SAYAR VE BELLİ BİR SÜRE SONRA
        GOTO   BGECIK_SAY2          ; BB,0 = 1 OLUR. YANİ YUKARIDAKİ BÖLÜMLE BİRLİKTE
        CLRF   BGECIK1              ; BIRAKICI SELENOİDİN BIRAKMADAN ÖNCE BEKLENMESİ
        CLRF   BGECIK2              ; BURADA SAĞLANIR. BURADA DİKKAT EDİLMESİ GEREKEN
        BCF    BB,0                 ; NOKTA AA,0 =1 OLDUĞU ZAMAN SÜRESİNİN
        GOTO   BGECIK_CIKIS2        ; BU BÖLÜMDEKİ BEKLEME SÜRESİNDEN FAZLA OLMASI
BGECIK_SAY2                         ; GEREKTİĞİDİR
        BTFSC  BB,0                 ;       "
        GOTO   BGECIK_CIKIS2        ;       "
```

```
        MOVLW H'25'             ;       "
        SUBWF  BGECIK2,W        ;       "
        BTFSS  STATUS,0         ;       "
        GOTO   BGECIK_CIKIS2    ;       "
        MOVLW H'66'             ;       "
        SUBWF  BGECIK1,W        ;       "
        BTFSS  STATUS,0         ;       "
        GOTO   BGECIK_CIKIS2    ;       "
        BSF    BB,0             ;       "
BGECIK_CIKIS2                   ;       "
;*********************************
        BTFSC  BIRAK,1          ; BU BÖLÜM İSE BIRAKICI SELENOİDİN (PORTB,1)
        GOTO   BIRAK_SAY2       ; HEM ÇALIŞMASI HEMDE BELLİ BİR SÜRE ÇALIŞIR
        BTFSC  BIRAK,0          ; HALDEBEKLEMESİ İÇİN GEREKEN ZAMANI SAĞLAR
        GOTO   BIRAK_SET2       ; BAŞLANGIÇ ŞARTI (BB,0 =1, 1 PALS YETERLİDİR)
        BCF    STATUS,5         ; HEMEN PORTA,1=1 OLUR VE BELİLİ BİR
        BTFSS  BB,0             ; SÜRE BEKLEYEREK 0 OLUR        "
        GOTO   BIRAK_SET2       ;       "
        CLRF   BIRAK1           ;       "
        CLRF   BIRAK2           ;       "
        BSF    BIRAK,1          ;       "
        BCF    STATUS,5         ;       "
        BSF    PORTA,1          ;       "
BIRAK_SAY2                      ;       "
        MOVLWH'15'              ;       "
        SUBWF  BIRAK2,W         ;       "
        BTFSS  STATUS,2         ;       "
        GOTO   BIRAK_SAY_KONT2  ;       "
        MOVLWH'66'              ;       "
        SUBWF  BIRAK1,W         ;       "
BIRAK_SAY_KONT2                 ;       "
        BTFSS  STATUS,0         ;       "
        GOTO   BIRAK_SET2       ;       "
        BCF    BIRAK,1          ;       "
        BCF    STATUS,5         ;       "
        BCF    PORTA,1          ;       "
BIRAK_SET2                      ;       "
        BCF    STATUS,5         ;       "
        BTFSS  BB,0             ;       "
        GOTO   BIRAK_SIL2       ;       "
        BSF    BIRAK,0          ;       "
        GOTO   BIRAK_GECIK_CIK2 ;       "
BIRAK_SIL2                      ;       "
        BCF    BIRAK,0          ;       "
BIRAK_GECIK_CIK2                ;       "
;************************************************
        GOTO   BOZUK_KONTROL1
;*********************************
;%%%%%%%%%%%%%%%% DÖRDÜNCÜ PROGRAM %%%%%%%%%%%%%%%%%%%%
START4
;***************************************
ITICI_KONTROL3
;***************************************
        BCF    STATUS,5         ;BANK0 A GEC
        BTFSC  RING,1           ;ITME EMRİ VARSA, VE İTİCI ÇALIŞIYORSA
        GOTO   RESET_RING       ;RESET_RING ETİKETİNE GİT
        BTFSC  PORTA,0          ;       "
        GOTO   RESET_RING       ;       "
        BTFSC  PORTB,5          ;CUBUK VARSA
        GOTO   ITME_CIKIS3      ;ITME_CIKIS3 ETİKETİNE GİT
        BTFSS  PORTB,2          ;
        GOTO   RESET_RING       ;HİÇ BİRİ YOKSA
        BTFSS  METAL,1          ;HALKA VAR AMA METAL Mİ?
        GOTO   PR_IT3           ;METAL DEĞİLSE PR_IT3 ETİKETİNE GİT
;MR_IT                          ;
        CALL   GECIKME3         ;BİRAZ BEKLE
        BTFSC  MHALKA,0         ;MHALKA YOKSA
        GOTO   RESET_RING       ;VARSA
        BTFSC  PHALKA,0         ;PHALKA YOKSA
```

```
        GOTO    RESET_RING          ;VARSA
;*********************************  ;
        BTFSS   MR,2                ;METAL HALKA YETERİ KADAR 5 ADET
        GOTO    CIK                 ;İTİLMİŞ İSE
        BTFSC   MR,0                ;       "
        GOTO    RESET_RING          ;RESET_RING ETİKETİNE GİT
CIK                                 ;
;*********************************  ;EĞER METAL HALKA YETERİ KADAR
        BTFSC   HALKA,5             ;İTİLMEMİŞ İSE HALKA REG. 5. BİTİNE BAK
        GOTO    RESET_RING          ;5. BİT DOLU(1)İSE RESET_RING ETİKETİNE GİT
        BTFSC   HALKA,4             ;HALKA REG. 5. BİTİ BOŞ İSE 4. BİTE BAK
        GOTO    MRING_5_SET         ;4. BİT DOLU(1) İSE MRING_5_SET ETİK. GİT
        BTFSC   HALKA,3             ;4. BİT BOŞ İSE 3. BİTE BAK
        GOTO    MRING_4_SET         ;3. BİT DOLU(1) İSE MRING_4_SET ETİK. GİT
        BTFSC   HALKA,2             ;3. BİT BOŞ(0) İSE 2. BİTE BAK
        GOTO    MRING_3_SET         ;2. BİT DOLU(1) İSE MRING_3_SET ETİK. GİT
        BTFSC   HALKA,1             ;2. BİT BOŞ(0) İSE 1. BİTE BAK
        GOTO    MRING_2_SET         ;1. BİT DOLU(1) İSE MRING_2_SET ETİK. GİT
MRING_1_SET                         ;1. BİT BOŞ(0) İSE 1. BİTİ DOLDUR
        BSF     MRING,1             ;MRING,1=1
        BSF     HALKA,1             ;HALKA,1=1
        GOTO    MRING_SET3          ;MRING_SET3 ETİKETİNE GİT
MRING_2_SET                         ;
        BSF     MRING,2             ;MRING,2=1
        BSF     HALKA,2             ;HALKA,2=1
        GOTO    MRING_SET3          ;
MRING_3_SET                         ;
        BSF     MRING,3             ;MRING,3=1
        BSF     HALKA,3             ;HALKA,3=1
        GOTO    MRING_SET3          ;
MRING_4_SET                         ;
        BSF     MRING,4             ;MRING,4=1
        BSF     HALKA,4             ;HALKA,4=1
        GOTO    MRING_SET3          ;
MRING_5_SET                         ;
        BSF     MRING,5             ;MRING,5=1
        BSF     HALKA,5             ;HALKA,5=1
        GOTO    MRING_SET3          ;
MRING_SET3                          ;
        BSF     RING,1              ;RING,1=1 METAL HALKA İTME EMRİ VER
        BSF     MHALKA,0            ;MHALKA,0=1
        INCF    MR,1                ;MR=MR+1
        GOTO    ITME_CIKIS3         ;ITME_CIKIS3 ETİKETİNE GİT
;*********************************
PR_IT3                              ;
        CALL    GECIKME3            ;BİRAZ BEKLE
        BTFSC   MHALKA,0            ;MHALKA,0 YOK MU?
        GOTO    RESET_RING          ;VARSA
        BTFSC   PHALKA,0            ;PHALKA,0 YOK MU?
        GOTO    RESET_RING          ;VARSA (İTİLECEK HALKA HENÜZ AYIRICI
;*****************                 ;BÖLGEYİ TERKETMEMİŞSE)
        BTFSS   PR,2                ;PLASTİK HALKA YETERİ KADAR 5 ADET
        GOTO    CIK1                ;İTİLMİŞ İSE
        BTFSC   PR,0                ;       "
        GOTO    RESET_RING          ;RESET_RING ETİKETİNE GİT
CIK1                                ;EĞER PLASTİK HALKA YETERİ KADAR
        BTFSC   HALKA,5             ;İTİLMEMİŞ İSE HALKA REG. 5. BİTİNE BAK
        GOTO    RESET_RING          ;5. BİT DOLU(1)İSE RESET_RING ETİKETİNE GİT
        BTFSC   HALKA,4             ;HALKA REG. 5. BİTİ BOŞ İSE 4. BİTE BAK
        GOTO    PRING_5_SET         ;4. BİT DOLU(1) İSE PRING_5_SET ETİK. GİT
        BTFSC   HALKA,3             ;4. BİT BOŞ İSE 3. BİTE BAK
        GOTO    PRING_4_SET         ;3. BİT DOLU(1) İSE PRING_4_SET ETİK. GİT
        BTFSC   HALKA,2             ;3. BİT BOŞ(0) İSE 2. BİTE BAK
        GOTO    PRING_3_SET         ;2. BİT DOLU(1) İSE PRING_3_SET ETİK. GİT
        BTFSC   HALKA,1             ;2. BİT BOŞ(0) İSE 1. BİTE BAK
        GOTO    PRING_2_SET         ;1. BİT DOLU(1) İSE PRING_2_SET ETİK. GİT
PRING_1_SET                         ;1. BİT BOŞ(0) İSE 1. BİTİ DOLDUR
        BSF     PRING,1             ;PRING,1=1
        BSF     HALKA,1             ;HALKA,1=1
```

```
        GOTO    PRING_SET           ;PRING_SET ETİKETİNE GİT
PRING_2_SET                         ;
        BSF     PRING,2             ;PRING,2=1
        BSF     HALKA,2             ;HALKA,2=1
        GOTO    PRING_SET           ;
PRING_3_SET                         ;
        BSF     PRING,3             ;PRING,3=1
        BSF     HALKA,3             ;HALKA,3=1
        GOTO    PRING_SET           ;
PRING_4_SET                         ;
        BSF     PRING,4             ;PRING,4=1
        BSF     HALKA,4             ;HALKA,4=1
        GOTO    PRING_SET           ;
PRING_5_SET                         ;
        BSF     PRING,5             ;PRING,5=1
        BSF     HALKA,5             ;HALKA,5=1
        GOTO    PRING_SET           ;
PRING_SET                           ;
        BSF     RING,1              ;RING,1=1 PLASTİK HALKA İTME EMRİ VER
        BSF     PHALKA,0            ;PHALKA,0=1
        INCF    PR,1                ;PR=PR+1
        GOTO    ITME_CIKIS3         ;ITME_CIKIS3 ETİKETİNE GİT
;*********************************
;
RESET_RING                          ;
        BCF     RING,1              ;İTME EMRİNİ GERİ ÇEK
;METAL_RING_SIL                     ;
        BTFSC   PORTB,5             ;RİNG AYIRICI BÖLGEYİ TERKETMİŞ İSE
        GOTO    ITME_CIKIS3         ;       "
        BTFSC   PORTB,2             ;       "
        GOTO    ITME_CIKIS3         ;       "
        CALL    GECIKME3            ;BİRAZ BEKLE
        BCF     METAL,1             ;METAL,1 =0
        BCF     MHALKA,0            ;MHALKA,0 =0
        BCF     PHALKA,0            ;PHALKA,0 =0 YAP
ITME_CIKIS3                         ;
;*********************************
;
BIRAKICI_KONTROL                    ;
;*********************************
;
        BTFSC   MARKER,0            ;
        GOTO    DOLDUR_CIKIS        ;
        BTFSS   PORTB,5             ;ÇUBUK VAR MI?
        GOTO    DOLDUR_CIKIS        ;ÇUBUK YOKSA DOLDUR_CIKIS ETİKETİNE GİT
        BTFSC   METAL,2             ;ÇUBUK PLASTİK Mİ?
        GOTO    M_PEG_DOLDUR        ;DEĞİLSE METAL PEG E BAK
P_PEG_DOLDUR                        ;
        BTFSC   KENAR,0             ;KENAR,0=0 MI?
        GOTO    RESET_PEG           ;HAYIR
        BTFSC   MARKER,0            ;MARKER,0=0 MI?
        GOTO    RESET_PEG           ;HAYIR
        BTFSS   PLASTİK,0           ;PLASTIK,0=1 Mİ?
        GOTO    RESET_PEG           ;HAYIR
        BTFSC   CUBUK,3             ;GELEN PLASTİK CUBUK İÇİN CUBUK REGİS.
        GOTO    DOLDUR_CIKIS        ;3. BİTİNE BAK DOLU İSE DOLDUR_CIKIS A GİT
        BTFSC   CUBUK,2             ;3. BİT BOŞ(0) İSE 2. BİTE BAK
        GOTO    PCUBUK_3_SET        ;2. BİT DOLU(1) İSE PCUBUK_3_SET ETİK. GİT
        BTFSC   CUBUK,1             ;2. BİT BOŞ(0) İSE 1. BİTE BAK
        GOTO    PCUBUK_2_SET        ;1. BİT DOLU(1) İSE PCUBUK_2_SET ETİK. GİT
        GOTO    PCUBUK_1_SET        ;1. BİT BOŞ(0) İSE PCUBUK VE CUBUK REGİS.
PCUBUK_1_SET                        ;1. BİTLERİNİ DOLDUR
        BSF     PCUBUK,1            ;PCUBUK,1 =1
        BSF     CUBUK,1             ;CUBUK,1 =1
        GOTO    CUBUK_SET           ;
PCUBUK_2_SET                        ;
        BSF     PCUBUK,2            ;PCUBUK,2 =1
        BSF     CUBUK,2             ;CUBUK,2 =1
        GOTO    CUBUK_SET           ;
PCUBUK_3_SET                        ;
        BSF     PCUBUK,3            ;PCUBUK,3 =1
        BSF     CUBUK,3             ;CUBUK,3 =1
```

```
        GOTO   CUBUK_SET            ;
;*********************************
M_PEG_DOLDUR                        ;CUBUK METAL İSE
        BTFSC  KENAR,0              ;KENAR,0 = 0 İSE
        GOTO   RESET_PEG            ;DEĞİLSE
        BTFSC  MARKER,0             ;MARKER,0 = 0 İSE
        GOTO   RESET_PEG            ;DEĞİLSE
        BTFSC  CUBUK,3              ;GELEN METAL CUBUK İÇİN CUBUK REGİS.
        GOTO   DOLDUR_CIKIS         ;3. BİTİNE BAK DOLU İSE DOLDUR_CIKIS A GİT
        BTFSC  CUBUK,2              ;3. BİT BOŞ(0) İSE 2. BİTE BAK
        GOTO   MCUBUK_3_SET         ;2. BİT DOLU(1) İSE MCUBUK_3_SET ETİK. GİT
        BTFSC  CUBUK,1              ;2. BİT BOŞ(0) İSE 1. BİTE BAK
        GOTO   MCUBUK_2_SET         ;1. BİT DOLU(1) İSE MCUBUK_2_SET ETİK. GİT
        GOTO   MCUBUK_1_SET         ;1. BİT BOŞ(0) İSE MCUBUK VE CUBUK REGİS.
MCUBUK_1_SET                        ;1. BİTLERİNİ DOLDUR
        BSF    MCUBUK,1             ;PCUBUK,1 =1
        BSF    CUBUK,1              ;CUBUK,1 =1
        GOTO   CUBUK_SET            ;
MCUBUK_2_SET                        ;
        BSF    MCUBUK,2             ;PCUBUK,2 =1
        BSF    CUBUK,2              ;CUBUK,2 =1
        GOTO   CUBUK_SET            ;
MCUBUK_3_SET                        ;
        BSF    MCUBUK,3             ;PCUBUK,3 =1
        BSF    CUBUK,3              ;CUBUK,3 =1
        GOTO   CUBUK_SET            ;
CUBUK_SET                           ;
        BSF    MARKER,0             ;
DOLDUR_CIKIS                        ;
;*********************************
;PEG_GELIRSE_ZAMANLAYICIYI_BASLAT;
;************************************************
        BTFSC  AGECIK,1             ;BU BÖLÜM CUBUK GELİRSE BELLİ BİR SÜRE SONRA
        GOTO   AGECIK_SAY3          ;HALKA İLE KARŞILAŞTIRMAYI SAĞLAYACAK OLAN
        BTFSC  AGECIK,0             ;KESME ZAMAN GECİKMESİNİN BAŞLANGICIDIR.
        GOTO   AGECIK_SET3          ;CUBUK GELDİĞİNDE(PORTB,5=1) HEMEN AA REG.
        BCF    STATUS,5             ;0. BİTİ 1 OLUR VE BELLİ BİR SÜRE AA REG.
        BTFSS  PORTB,5              ;0. BİTİNİ SET EDER, SÜRE BİTİNCE RESET EDER.
        GOTO   AGECIK_SET3          ;       "
        CLRF   AGECIK1              ;       "
        CLRF   AGECIK2              ;       "
        BSF    AGECIK,1             ;       "
        BSF    AA,0                 ;       "
AGECIK_SAY3                         ;       "
        MOVLW H'20'                 ;       "
        SUBWF  AGECIK2,W            ;       "
        BTFSS  STATUS,2             ;       "
        GOTO   AGECIK_SAY_KONT3     ;       "
        MOVLW H'38'                 ;       "
        SUBWF  AGECIK1,W            ;       "
AGECIK_SAY_KONT3                    ;       "
        BTFSS  STATUS,0             ;       "
        GOTO   AGECIK_SET3          ;       "
        BCF    AGECIK,1             ;       "
        BCF    AA,0                 ;       "
AGECIK_SET3                         ;       "
        BCF    STATUS,5             ;       "
        BTFSS  PORTB,5              ;       "
        GOTO   AGECIK_SIL3          ;       "
        BSF    AGECIK,0             ;       "
        GOTO   AGECIK_GECIK_CIK3    ;       "
AGECIK_SIL3                         ;       "
        BCF    AGECIK,0             ;       "
AGECIK_GECIK_CIK3                   ;       "
;************************************************
        BTFSC  AA,0                 ;BU BÖLÜM AA,0=1 OLDUĞU ANDA ÇOK KISA
        GOTO   BGECIK_SAY3          ;BİR ZAMAN DİLİMİ BB REG. 0. BİTİNİ
        CLRF   BGECIK1              ;SET EDER
        CLRF   BGECIK2              ;       "
```

```
        BSF     BB,0                    ;       "
        GOTO    BGECIK_CIKIS3           ;       "
BGECIK_SAY3                             ;       "
        BTFSS   BB,0                    ;       "
        GOTO    BGECIK_CIKIS3           ;       "
        MOVLW H'00'                     ;       "
        SUBWF   BGECIK2,W               ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BGECIK_CIKIS3           ;       "
        MOVLW H'78'                     ;       "
        SUBWF   BGECIK1,W               ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BGECIK_CIKIS3           ;       "
        BCF     BB,0                    ;       "
BGECIK_CIKIS3                           ;       "
;************************************************
;
        BTFSC   CGECIK,1                ;BU BÖLÜM İSE BB,0=1 OLDUĞU ANDA
        GOTO    CGECIK_SAY              ;CC REG. 0. BİTİNİN BELLİ BİR SÜRE
        BTFSC   CGECIK,0                ;SET OLMASINI VE ZAMAN DOLUNCA
        GOTO    CGECIK_SET              ;RESET OLMASINI SAĞLAR.
        BCF     STATUS,5                ;       "
        BTFSS   BB,0                    ;       "
        GOTO    CGECIK_SET              ;       "
        CLRF    CGECIK1                 ;       "
        CLRF    CGECIK2                 ;       "
        BSF     CGECIK,1                ;       "
        BSF     CC,0                    ;       "
CGECIK_SAY                              ;       "
        MOVLW H'24'                     ;       "
        SUBWF   CGECIK2,W               ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    CGECIK_SAY_KONTROL      ;       "
        MOVLW H'99'                     ;       "
        SUBWF   CGECIK1,W               ;       "
CGECIK_SAY_KONTROL                      ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    CGECIK_SET              ;       "
        BCF     CGECIK,1                ;       "
        BCF     STATUS,5                ;       "
        BCF     CC,0                    ;       "
CGECIK_SET                              ;       "
        BTFSS   BB,0                    ;       "
        GOTO    CGECIK_SIL              ;       "
        BSF     CGECIK,0                ;       "
        GOTO    CGECIK_GECIKME_CIKIS    ;       "
CGECIK_SIL                              ;       "
        BCF     CGECIK,0                ;       "
CGECIK_GECIKME_CIKIS                    ;       "
;************************************************
;
        BTFSC   CC,0                    ;BU BÖLÜMDE CC REG. 0. BİTİNİN
        GOTO    DGECIK_SAY              ;1 (SET) OLDUĞU ZAMAN SÜRESİNCE
        CLRF    DGECIK1                 ;SAYMAYA DEVAM EDER VE BELLİ BİR
        CLRF    DGECIK2                 ;GECİKME İLE DD REG. 0. BİTİNİ
        BCF     DD,0                    ;SET EDER. DOLAYISIYLA YUKARIDAKİ
        GOTO    DGECIK_CIKIS            ;ZAMAN GECİKMESİ BU BÖLÜMDEKİNDEN
DGECIK_SAY                              ;FAZLA OLMALIDIR.
        BTFSC   DD,0                    ;       "
        GOTO    DGECIK_CIKIS            ;       "
        MOVLW H'23'                     ;       "
        SUBWF   DGECIK2,W               ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    DGECIK_CIKIS            ;       "
        MOVLW H'51'                     ;       "
        SUBWF   DGECIK1,W               ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    DGECIK_CIKIS            ;       "
        BSF     DD,0                    ;       "
DGECIK_CIKIS                            ;       "
;************************************************
;
```

```
        BTFSC   EGECIK,1                ;BU BÖLÜM DD REG. 0. BİTİNİN
        GOTO    EGECIK_SAY              ;SET OLDUĞU İLK ANDA ÇIKIŞ VERİR
        BTFSC   EGECIK,0                ;YANİ EE,0=1 SET OLUR VEKISA BİR
        GOTO    EGECIK_SET              ;SÜRE BEKLETİR. BİZİM ASIL KULLANACAĞIMIZ
        BCF     STATUS,5                ;REGISTER EE REGISTERIDIR. BELLİ BİR SÜRE
        BTFSS   DD,0                    ;SONRA EE,0=1 OLDUĞU DURUMDA CUBUK VE
        GOTO    EGECIK_SET              ;HALKA DURUMLARI KARŞILAŞTIRILACAK VE
        CLRF    EGECIK1                 ;GEREKİYORSA ÜRETİM OLACAKTIR.
        CLRF    EGECIK2                 ;BU KARŞILAŞTIRMA YAPILIRKEN ARKADAN GELEN
        BSF     EGECIK,1                ;CUBUK İÇİN SAYIM YENİDEN BAŞLATILABİLECEK
        BCF     STATUS,5                ;VE BU SAYIM DİĞERİNİ ETKİLEMEYECEKTİR
        BSF     EE,0                    ;       "
EGECIK_SAY                              ;       "
        MOVLW H'01'                     ;       "
        SUBWF EGECIK2,W                 ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    EGECIK_SAY_KONTROL;     "
        MOVLW H'66'                     ;       "
        SUBWF EGECIK1,W                 ;       "
EGECIK_SAY_KONTROL                      ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    EGECIK_SET              ;       "
        BCF     EGECIK,1                ;       "
        BCF     STATUS,5                ;       "
        BCF     EE,0                    ;       "
EGECIK_SET                              ;       "
        BTFSS   DD,0                    ;       "
        GOTO    EGECIK_SIL              ;       "
        BSF     EGECIK,0                ;       "
        GOTO    EGECIK_GECIKME_CIKIS;   "
EGECIK_SIL                              ;       "
        BCF     EGECIK,0                ;       "
EGECIK_GECIKME_CIKIS                    ;       "
;*************************************************
BIRAKACAKMI?                            ;
        BTFSC   KENAR,0                 ;BİR CUBUK İÇİN ÜRETİM DEVAM EDİYOR MU?
        GOTO    RESET_PEG               ;EVET
        BTFSS   EE,0                    ;EE,0=1 Mİ? ZAMAN GECIKMESİ DOLDU MU?
        GOTO    RESET_PEG               ;HAYIR
        BTFSC   MCUBUK,1                ;GELEN İLK CUBUK METAL Mİ?
        GOTO    METAL_PEG               ;EVET İSE
        BTFSC   PCUBUK,1                ;HAYIR İSE PLASTİK Mİ?
        GOTO    PLASTIK_PEG             ;EVET
        GOTO    RESET_PEG               ;HAYIR
PLASTIK_PEG                             ;
        BTFSC   PRING,1                 ;TOPLAMA BÖL.'DE İLK BÖLÜMDE PLASTİK HALKA MI VAR?
        GOTO    PRPP_URET               ;PLASTİK HALKA YOK İSE METAL HALKAYA BAK
        BTFSC   MRING,1                 ;METAL HALKA MI VAR?
        GOTO    MRPP_URET               ;EVET
        GOTO    PKAYDIR                 ;HİÇ HALKA YOKSA MEVCUT CUBUĞU SİL
METAL_PEG                               ;
        BTFSC   KENAR,0                 ;BİR CUBUK İÇİN ÜRETİM DEVAM EDİYOR MU?
        GOTO    RESET_PEG               ;EVET
        BTFSC   PRING,1                 ;TOPLAMA BÖLGESİ İLK BÖLÜMDE PLASTİK HALKA MI VAR?
        GOTO    PRMP_URET               ;PLASTİK HALKA YOK İSE METAL HALKAYA BAK
        BTFSC   MRING,1                 ;METAL HALKA MI VAR?
        GOTO    MRMP_URET               ;EVET
        GOTO    PKAYDIR                 ;HİÇ HALKA YOKSA MEVCUT CUBUĞU SİL
;*************************************************
PRPP_URET                               ;
        CALL    GECIKME3                ;BİRAZ BEKLE
        CALL    GECIKME3                ;       "
        CALL    GECIKME3                ;       "
        CALL    GECIKME3                ;       "
        BTFSC   PORTB,1                 ;ÜRETİLMİŞ OLAN HALKA BEKLİYOR MU?
        GOTO    RESET_PEG               ;EVET
        MOVLW H'03'                     ;PRPP YERLİ SAYIDA ÜRETİLMİŞ Mİ?
        SUBWF PRPP,0                    ;       "
        BTFSC   STATUS,0                ;       "
```

```
        GOTO    PKAYDIR         ;SAYI YETERLİ İSE MEVCUT ÇUBUĞU SİL
        BSF     PEG,1           ;SAYI YERLİ DEĞİLSE ÜRETME EMRİ VER
        BSF     KENAR,0         ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        INCF    PRPP,1          ;PRPP=PRPP+1
        GOTO    KAYDIR          ;HEM HALKALARI HEMDE CUBUKLARI KAYDIR
;*********************************
;
MRPP_URET                       ;
        CALL    GECIKME3        ;BİRAZ BEKLE
        CALL    GECIKME3        ;       "
        CALL    GECIKME3        ;       "
        CALL    GECIKME3        ;       "
        BTFSC   PORTB,1         ;ÜRETİLMİŞ OLAN BİR HALKA BEKLİYOR MU?
        GOTO    RESET_PEG       ;EVET
        MOVLW H'02'             ;MRPP YERLİ SAYIDA ÜRETİLMİŞ Mİ?
        SUBWF MRPP,0            ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    PKAYDIR         ;SAYI YETERLİ İSE MEVCUT ÇUBUĞU SİL
        BSF     PEG,1           ;SAYI YERLİ DEĞİLSE ÜRETME EMRİ VER
        BSF     KENAR,0         ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        INCF    MRPP,1          ;MRPP=MRPP+1
        GOTO    KAYDIR          ;HEM HALKALARI HEMDE CUBUKLARI KAYDIR
;*********************************
;
PRMP_URET                       ;
        CALL    GECIKME3        ;BİRAZ BEKLE
        CALL    GECIKME3        ;       "
        CALL    GECIKME3        ;       "
        CALL    GECIKME3        ;       "
        BTFSC   PORTB,1         ;ÜRETİLMİŞ OLAN BİR HALKA BEKLİYOR MU?
        GOTO    RESET_PEG       ;EVET
        MOVLW H'02'             ;PRMP YERLİ SAYIDA ÜRETİLMİŞ Mİ?
        SUBWF PRMP,0            ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    PKAYDIR         ;SAYI YETERLİ İSE MEVCUT ÇUBUĞU SİL
        BSF     PEG,1           ;SAYI YERLİ DEĞİLSE ÜRETME EMRİ VER
        BSF     KENAR,0         ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        INCF    PRMP,1          ;PRMP=PRMP+1
        GOTO    KAYDIR          ;HEM HALKALARI HEMDE CUBUKLARI KAYDIR
;*********************************
;
MRMP_URET                       ;
        CALL    GECIKME3        ;BİRAZ BEKLE
        CALL    GECIKME3        ;       "
        CALL    GECIKME3        ;       "
        CALL    GECIKME3        ;       "
        BTFSC   PORTB,1         ;ÜRETİLMİŞ OLAN BİR HALKA BEKLİYOR MU?
        GOTO    RESET_PEG       ;EVET
        MOVLW H'03'             ;MRMP YERLİ SAYIDA ÜRETİLMİŞ Mİ?
        SUBWF MRMP,0            ;       "
        BTFSC   STATUS,0        ;       "
        GOTO    PKAYDIR         ;SAYI YETERLİ İSE MEVCUT ÇUBUĞU SİL
        BSF     PEG,1           ;SAYI YERLİ DEĞİLSE ÜRETME EMRİ VER
        BSF     KENAR,0         ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        INCF    MRMP,1          ;MRMP=MRMP+1
        GOTO    KAYDIR          ;HEM HALKALARI HEMDE CUBUKLARI KAYDIR
;*********************************
;
KAYDIR                          ;
        CALL    GECIKME3        ;BİRAZ BEKLE
        CALL    GECIKME3        ;       ;       "
        MOVLW B'00111110'       ;W <-- B'00111110'
        RRF     HALKA,F         ;HALKA REG. BİTLERİNİ BİRER SAĞA KAYDIR
        RRF     PRING,F         ;PRING REG      "
        RRF     MRING,F         ;MRING REG      "
        ANDWF HALKA,F           ;HALKA REG. KULLANILMAYAN BİTLERİNİ SİL
        ANDWF PRING,F           ;PRING REG      "
        ANDWF MRING,F           ;MRING REG      "
        RRF     CUBUK,F         ;CUBUK REG. BİTLERİNİ BİRER SAĞA KAYDIR
        RRF     PCUBUK,F        ;PCUBUK REG     "
        RRF     MCUBUK,F        ;MCUBUK REG     "
        ANDWF CUBUK,F           ;CUBUK REG. KULLANILMAYAN BİTLERİNİ SİL
        ANDWF PCUBUK,F          ;PCUBUK REG     "
```

```
        ANDWF MCUBUK,F          ;MCUBUK REG   "
        GOTO  BIRAK_CIKIS       ;BIRAK_CIKIS ETİKETİNE GİT
;*********************************
PKAYDIR                         ;
        CALL  GECIKME3          ;KULLANILMAYACAK OLAN CUBUK VARSA
        CALL  GECIKME3          ;BİRAZ BEKLE
        CALL  GECIKME3          ;       "
        CALL  GECIKME3          ;       "
        CALL  GECIKME3          ;       "
        CALL  GECIKME3          ;       "
        CALL  GECIKME3          ;       "
        CALL  GECIKME3          ;       "
        CALL  GECIKME3          ;       "
        MOVLW B'00001110'       ;W <-   B'00001110'
        RRF   CUBUK,F           ;CUBUK REG. BİTLERİNİ BİRER SAĞA KAYDIR
        RRF   PCUBUK,F          ;PCUBUK REG   "
        RRF   MCUBUK,F          ;MCUBUK REG   "
        ANDWF CUBUK,F           ;CUBUK REG. KULLANILMAYAN BİTLERİNİ SİL
        ANDWF PCUBUK,F          ;PCUBUK REG   "
        ANDWF MCUBUK,F          ;MCUBUK REG   "
        GOTO  METAL_PEG_SIL     ;METAL_PEG_SIL ETİKETİNE GİT
;*********************************
PEG_SET                         ;
        BSF   KENAR,0           ;AYNI CUBUK İÇİN ÜRETİM BİR KERE OLSUN
        GOTO  BIRAK_CIKIS       ;
RESET_PEG                       ;
        BCF   PEG,1             ;ÜRETME EMRİ VERME
METAL_PEG_SIL                   ;
        BTFSC PORTB,5           ;GELEN PEG AYIRMA BÖLGESİNİ TERKETMİŞSE
        GOTO  BIRAK_CIKIS       ;
        CALL  GECIKME3          ;BİRAZ BEKLE
        BCF   PLASTIK,0         ;PLASTİK,0=0
        BCF   METAL,2           ;METAL,2=0
        BCF   KENAR,0           ;KENAR,0=0
        BCF   MARKER,0          ;MARKER,0=0
BIRAK_CIKIS                     ;
;************************************************
;BIRAKTIKTAN SONRA BEKLE        ;
;*********************************
        BTFSC BIRAK,1           ;BU BÖLÜM HALKA BIRAKMA YADA ÜRETİM
        GOTO  BIRAK_SAY3        ;EMRİ VERİLDİĞİNDE (PEG,1=1) OLDUĞUNDA
        BTFSC BIRAK,0           ;HEMEN HALKAYI BIRAKTIRIR (PORTA,1=1)
        GOTO  BIRAK_SET3        ;VE PORTA,1 E BAĞLI BIRAKMA SELENOİDİNİN
        BCF   STATUS,5          ;BELLİ BİR SÜRE ÇALIŞMASINI SAĞLAR.
        BTFSS PEG,1             ;KESME GECİKMESİ BÖLÜMÜDÜR.
        GOTO  BIRAK_SET3        ;       "
        CLRF  BIRAK1            ;       "
        CLRF  BIRAK2            ;       "
        BSF   BIRAK,1           ;       "
        BCF   STATUS,5          ;       "
        BSF   PORTA,1           ;       "
BIRAK_SAY3                      ;       "
        MOVLW H'10'             ;       "
        SUBWF BIRAK2,W          ;       "
        BTFSS STATUS,2          ;       "
        GOTO  BIRAK_SAY_KONT3   ;       "
        MOVLW H'66'             ;       "
        SUBWF BIRAK1,W          ;       "
BIRAK_SAY_KONT3                 ;       "
        BTFSS STATUS,0          ;       "
        GOTO  BIRAK_SET3        ;       "
        BCF   BIRAK,1           ;       "
        BCF   STATUS,5          ;       "
        BCF   PORTA,1           ;       "
BIRAK_SET3                      ;       "
        BCF   STATUS,5          ;       "
        BTFSS PEG,1             ;       "
        GOTO  BIRAK_SIL3        ;       "
```

```
        BSF     BIRAK,0                 ;       "
        GOTO    BIRAK_GECIK_CIK3        ;       "
BIRAK_SIL3                              ;       "
        BCF     BIRAK,0                 ;       "
BIRAK_GECIK_CIK3                        ;       "
;*************************************************
;
;ALT BÖLÜMLER BÜTÜN PROGRAMLAR İÇİN ORTAK
;*************************************************
;
BOZUK_KONTROL1                          ;BOZUK,0. BİT 0 (BOZUK PARÇA YOK) MI?
        BTFSC   BOZUK,0                 ;HAYIR (BOZUK,0=1) İSE HAFIZADA TUT
        GOTO    BOZUK_KONTROL4          ;EVET (BOZUK,0=0 İSE) TEKRAR KONTROL ETMEYE BASLA
        BTFSS   PORTB,4                 ;PORTB,4 = 1 Mİ? (BOZUK BÖLGE ALT SENSÖR)
        GOTO    BOZUK_ITME              ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL2                          ;
        BTFSC   PORTB,6                 ;PORTB,6 = 0 MI? (BOZUK BÖLGE ÜST SENSÖR)
        GOTO    BOZUK_ITME              ;BOZUK_ITME ETİKETİNE GİT
BOZUK_KONTROL3                          ;
        BSF     BOZUK,0                 ;BOZUK,0 = 1 YAP (BOZUK PARÇAYI HAFIZAYA AL)
        GOTO    BOZUK_KONTROL4          ;BOZUK_KONTROL4 ETİKETİNE GİT
BOZUK_ITME                              ;
        BCF     BOZUK,0                 ;HAFIZAYI SİL
;*********************************
BOZUK_KONTROL4                          ;
        BTFSS   BOZUK,0                 ;BOZUK PARÇA ALGILANMIŞ MI?
        GOTO    BOZUK_KONTROL7          ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL5                          ;
        BTFSS   PORTB,3                 ;BOZUK PARÇA AYIRICI SELENOİD YANINDA MI?
        GOTO    BOZUK_KONTROL7          ;HAYIR İSE BOZUK_KONTROL7 ETİKETİNE GİT
BOZUK_KONTROL6                          ;
        BSF     BOZUK,1                 ;BOZUK PARÇAYI İTME EMRİ VER (BOZUK,1=1)
        BCF     BOZUK,0                 ;HAFIZAYI SİL
        GOTO    BOZUK_CIKIS             ;BOZUK_CIKIS ETİKETİNE GİT
BOZUK_KONTROL7                          ;
        BCF     BOZUK,1                 ;BOZUK PARÇA İTME EMRİ VERME (BOZUK,1=0)
BOZUK_CIKIS                             ;
;*********************************
;BOZUK_GECIKME                          ;
;*************************************************************
;
        BTFSC   BOZ,1                   ; BU BÖLÜM BOZUK PARÇA İTEN SELENOİD(PORTA,2) İÇİN
        GOTO    BOZUK_SAY               ; KESME GECİKMESİ SAĞLAR.
        BTFSC   BOZ,0                   ; İTME EMRİ (BOZUK,1=1) ALGILANDIĞI ANDA PORTA,2=1
        GOTO    BOZUK_SET               ; OLUR PARÇA İTİLİR.
        BCF     STATUS,5                ; İTTİKTEN SONRA BİR SÜRE BEKLENMESİ SAĞLANIR
        BTFSS   BOZUK,1                 ;       "
        GOTO    BOZUK_SET               ;       "
        CLRF    BOZ1                    ;       "
        CLRF    BOZ2                    ;       "
        BSF     BOZ,1                   ;       "
        BSF     PORTA,2                 ;       "
BOZUK_SAY                               ;       "
        MOVLW H'05'                     ;       "
        SUBWF BOZ2,W                    ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    BOZUK_SAY_KONTROL       ;       "
        MOVLW H'66'                     ;       "
        SUBWF BOZ1,W                    ;       "
BOZUK_SAY_KONTROL                       ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    BOZUK_SET               ;       "
        BCF     BOZ,1                   ;       "
        BCF     STATUS,5                ;       "
        BCF     PORTA,2                 ;       "
BOZUK_SET                               ;       "
        BCF     STATUS,5                ;       "
        BTFSS   BOZUK,1                 ;       "
        GOTO    BOZUK_SIL               ;       "
        BSF     BOZ,0                   ;       "
        GOTO    BOZUK_GECIKME_CIKIS     ;       "
BOZUK_SIL                               ;       "
```

```
        BCF     BOZ,0                   ;       "
BOZUK_GECIKME_CIKIS                     ;       "
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
ITICI_GECIKME                           ;BU BÖLÜM İTİLECEK OLAN HALKA İÇİN
        BTFSC   ITICI,1                 ;İTME SÜRESİNİ TAYİN EDER.
        GOTO    ITICI_SAY               ;EĞER İTME EMRİ (RING,1=1) VARSA
        BTFSC   ITICI,0                 ;O ANDA PARÇA İTİLİR VE SELENOİD İTİLİ
        GOTO    ITICI_SET               ;OLARAK BİRAZ BEKLER SONRA GERİ GELİR
        BCF     STATUS,5                ;İTME İÇİN PORTA,0=1 OLMALIDIR
        BTFSS   RING,1                  ;       "
        GOTO    ITICI_SET               ;       "
        CLRF    ITICI1                  ;       "
        CLRF    ITICI2                  ;       "
        BSF     ITICI,1                 ;       "
        BSF     PORTA,0                 ;       "
ITICI_SAY                               ;       "
        MOVLW H'05'                     ;       "
        SUBWF ITICI2,W                  ;       "
        BTFSS   STATUS,2                ;       "
        GOTO    ITICI_SAY_KONTROL       ;       "
        MOVLW H'66'                     ;       "
        SUBWF ITICI1,W                  ;       "
ITICI_SAY_KONTROL                       ;       "
        BTFSS   STATUS,0                ;       "
        GOTO    ITICI_SET               ;       "
        BCF     ITICI,1                 ;       "
        BCF     STATUS,5                ;       "
        BCF     PORTA,0                 ;       "
ITICI_SET                               ;       "
        BCF     STATUS,5                ;       "
        BTFSS   RING,1                  ;       "
        GOTO    ITICI_SIL               ;       "
        BSF     ITICI,0                 ;       "
        GOTO    ITICI_GECIKME_CIKIS     ;       "
ITICI_SIL                               ;       "
        BCF     ITICI,0                 ;       "
ITICI_GECIKME_CIKIS                     ;       "
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
METAL_TANIMLA                           ;
        BTFSS   PORTB,7                 ;METAL PARÇA VAR MI?
        GOTO    METALSIL3               ;METAL PARÇA YADA HİÇBİR PARÇA YOKSA
        BSF     METAL,0                 ;METAL REG. 0. BİTİNİ 1 YAP
METALSIL3                               ;
        BTFSC   METAL,1                 ;METAL REGISTERI NIN 1.BİTİ 0 MI?
        GOTO    METAL_TANIM3_CIKIS      ;METAL REGISTERI NIN 1.BİTİ 1 İSE
        BTFSC   METAL,2                 ;METAL REGISTERI NIN 2.BİTİ 0 MI?
        GOTO    METAL_TANIM3_CIKIS      ;METAL REGISTERI NIN 2.BİTİ 1 İSE
METAL_PEG_SET3                          ;
        BTFSS   METAL,0                 ;METAL REGISTERI NIN 0.BİTİ 1 Mİ?
        GOTO    METAL_TANIM3_CIKIS      ;METAL REGISTERI NIN 0.BİTİ 0 İSE
        BTFSS   PORTB,5                 ;PORTB NİN 5.BİTİ 1 Mİ?
        GOTO    METAL_RING_SET3         ;PORTB NİN 5.BİTİ 0 İSE (ÇUBUK YOKSA)
        CALL    GECIKME3                ;BİRAZ BEKLE
        BSF     METAL,2                 ;METAL REGISTERININ 2. BİTİNİ 1 YAP
        BCF     METAL,0                 ;METAL REGISTERININ 0. BİTİNİ 0 YAP
        GOTO    PROGRAM_SEC             ;
METAL_RING_SET3                         ;
        BTFSS   PORTB,2                 ;
        GOTO    METAL_TANIM3_CIKIS      ;
        CALL    GECIKME3                ;
        BSF     METAL,1                 ;METAL HALKA SET (METAL,1 = 1)
        BCF     METAL,0                 ;METAL,0 = 0
        GOTO    METAL_TANIM3_CIKIS      ;
METAL_TANIM3_CIKIS                      ;
        BTFSC   METAL,0                 ;METAL PARÇA ALGILANMIŞ İSE
        GOTO    PLASTIK_TANIM3_CIKIS    ;       "
        BTFSC   METAL,1                 ;       "
        GOTO    PLASTIK_TANIM3_CIKIS    ;       "
        BTFSC   METAL,2                 ;       "
```

```
        GOTO   PLASTIK_TANIM3_CIKIS ;PLASTİK_TANIMLA_CIKIS ETİKETİNE GİT
PLASTIK_TANIMLA3                    ;
        BSF    PLASTIK,0            ;PARÇAYI PLASTİK OLARAK TANI
        GOTO   PROGRAM_SEC          ;
PLASTIK_TANIM3_CIKIS                ;
        BCF    PLASTIK,0            ;
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
PROGRAM_SEC                         ;
        BTFSC  PORTA,4              ; PORTA,4  PORTB,0  PROGRAM
        GOTO   IKI_PROG             ;  0       ->     0  ->    1
        BTFSC  PORTB,0              ;  1       ->     0  ->    2
        GOTO   UC_PROG              ;  0       ->     1  ->    3
BIR_PROG                            ;  1       ->     1  ->    4
        GOTO   START1               ;
IKI_PROG                            ;          "
        BTFSS  PORTB,0              ;          "
        GOTO   START2               ;          "
DORT_PROG                           ;          "
        GOTO   START4               ;          "
UC_PROG                             ;          "
        GOTO   START3               ;          "
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
INT_ALT_PROG                        ;KESME ALT PROGRAMI BAŞLANGICI
        MOVWF SAKLA_W               ;W --> SAKLA_W
        SWAPF  STATUS,W             ;STATUS REG. SWAP YAP VE W YE YÜKLE
        MOVWF SAKLA_S               ;W --> SAKLA_S
        BCF    INTCON,2             ;TOIF <-- '0' (zaman aşımı yok)
        INCFSZ  ITIC11,F            ;ITIC11 = ITIC11+1, ITIC1,1 = 0 MI?
        GOTO   CIKIS1               ;HAYIR İSE
        INCF   ITIC12,F             ;EVET İSE ITIC12 = ITIC12+1
CIKIS1                              ;
        INCFSZ  AGECIK1,F           ;KESME ALT PROGRAMINDA BULUNAN BÜTÜN
        GOTO   CIKIS2               ; REGİSTERLER İÇİN AYNI İŞLEMİ
        INCF   AGECIK2,F            ; (İÇERİK ARTIRMA) TEKRARLA   "
CIKIS2                              ;          "
        INCFSZ  BOZ1,F              ;          "
        GOTO   CIKIS3               ;          "
        INCF   BOZ2,F               ;          "
CIKIS3                              ;          "
        INCFSZ  BIRAK1,F            ;          "
        GOTO   CIKIS4               ;          "
        INCF   BIRAK2,F             ;          "
CIKIS4                              ;          "
        INCFSZ  BGECIK1,F           ;          "
        GOTO   CIKIS5               ;          "
        INCF   BGECIK2,F            ;          "
CIKIS5                              ;          "
        INCFSZ  CGECIK1,F           ;          "
        GOTO   CIKIS6               ;          "
        INCF   CGECIK2,F            ;          "
CIKIS6                              ;          "
        INCFSZ  DGECIK1,F           ;          "
        GOTO   CIKIS7               ;          "
        INCF   DGECIK2,F            ;          "
CIKIS7                              ;          "
        INCFSZ  EGECIK1,F           ;          "
        GOTO   CIKIS8               ;          "
        INCF   EGECIK2,F            ;          "
CIKIS8                              ;          "
        MOVLW H'00'                 ;TMRO I SIFIRLA
        MOVWF TMR0                  ;          "
        SWAPF  SAKLA_S,W            ;W VE STATUS REG. TEKRAR YÜKLE
        MOVWF STATUS                ;          "
        SWAPF  SAKLA_W,F            ;          "
        SWAPF  SAKLA_W,W            ;          "
        RETFIE                      ;ANA PROGRAMA GERİDÖN
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
;DONGULU_ZAMAN_GECIKMELERI
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

```
GECIKME1                                ;
        MOVLW H'04'                     ;ÜÇLÜ DÖNGÜ İLE ZAMAN GECIKMESİ
        MOVWF SAY                       ;ALT PROGRAMI
DONGU                                   ;       "
        MOVLW H'FF'                     ;       "
        MOVWF SAY1                      ;       "
DONGU1                                  ;       "
        MOVLW H'FF'                     ;       "
        MOVWF SAY2                      ;       "
DONGU2                                  ;       "
        DECFSZ SAY2,F                   ;       "
        GOTO   DONGU2                   ;       "
        DECFSZ SAY1,F                   ;       "
        GOTO   DONGU1                   ;       "
        DECFSZ SAY,F                    ;       "
        GOTO   DONGU                    ;       "
        RETURN                          ;       "
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
GECIKME2                        ; ÇİFT DÖNGÜLÜ ZAMAN
        MOVLW H'08'             ; GECİKMESİ ALT
        MOVWF SAY1              ; PROGRAMI
DONGU11                                 ;       "
        MOVLW H'FF'             ;       "
        MOVWF SAY2              ;       "
DONGU22                                 ;       "
        DECFSZ SAY2,1           ;       "
        GOTO   DONGU22                  ;       "
        DECFSZ SAY1,1           ;       "
        GOTO   DONGU11                  ;       "
        RETURN                          ;       "
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
GECIKME3                                ;ÇİFT DÖNGÜLÜ ZAMAN
        MOVLW H'11'                     ;GECİKMESİ ALT
        MOVWF SAY1                      ;PROGRAMI
DONGU111                                ;       "
        MOVLW H'FF'                     ;       "
        MOVWF SAY2                      ;       "
DONGU222                                ;       "
        DECFSZ SAY2,1                   ;       "
        GOTO   DONGU222                 ;       "
        DECFSZ SAY1,1                   ;       "
        GOTO   DONGU111                 ;       "
        RETURN                          ;       "
        END                             ;PROGRAM SONU
```

EK-B Endüstriyel sistem üzerinde bulunan devrelerin açık şemaları

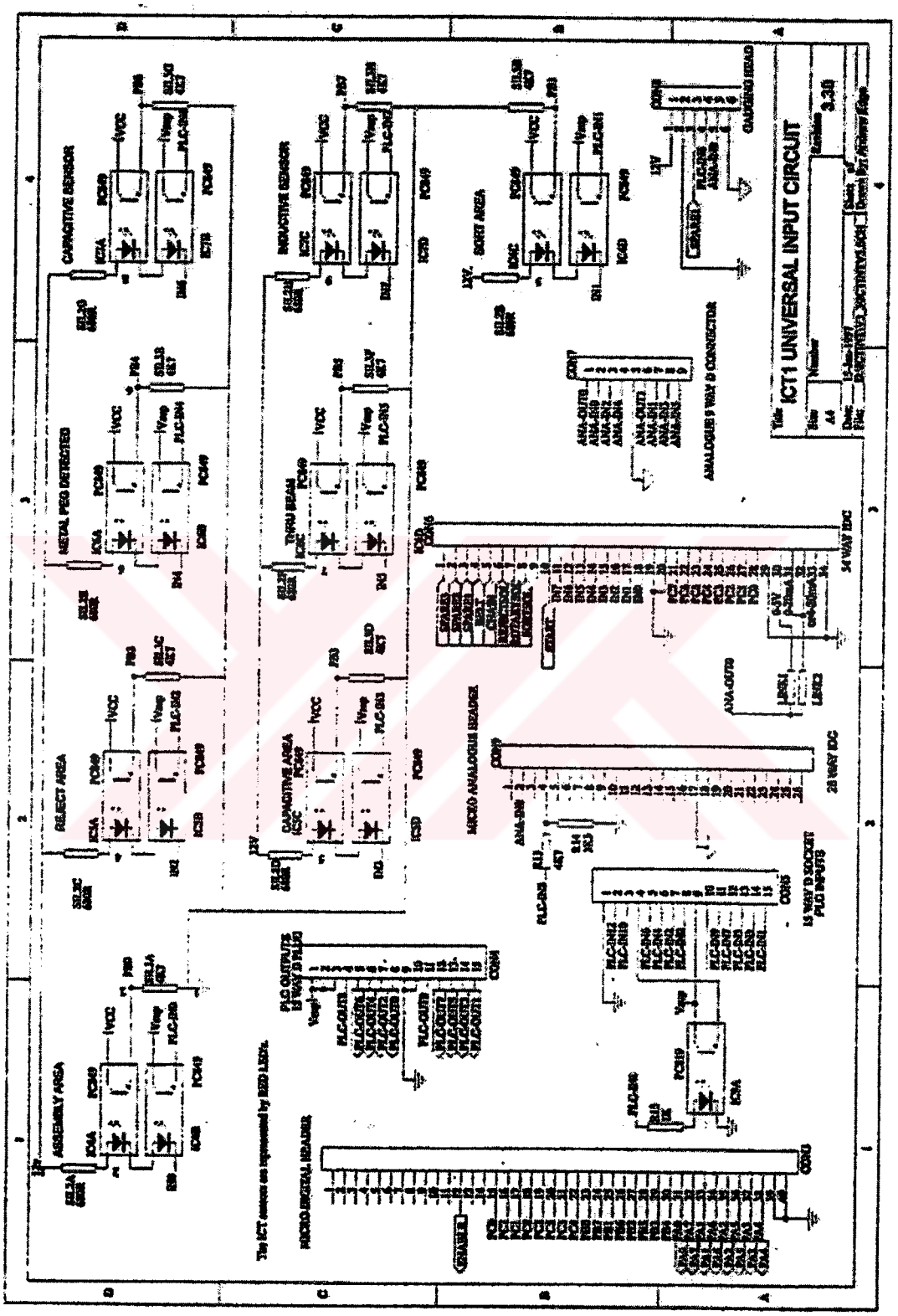


Figure 6.1. ICT Interface Board Circuit Diagram

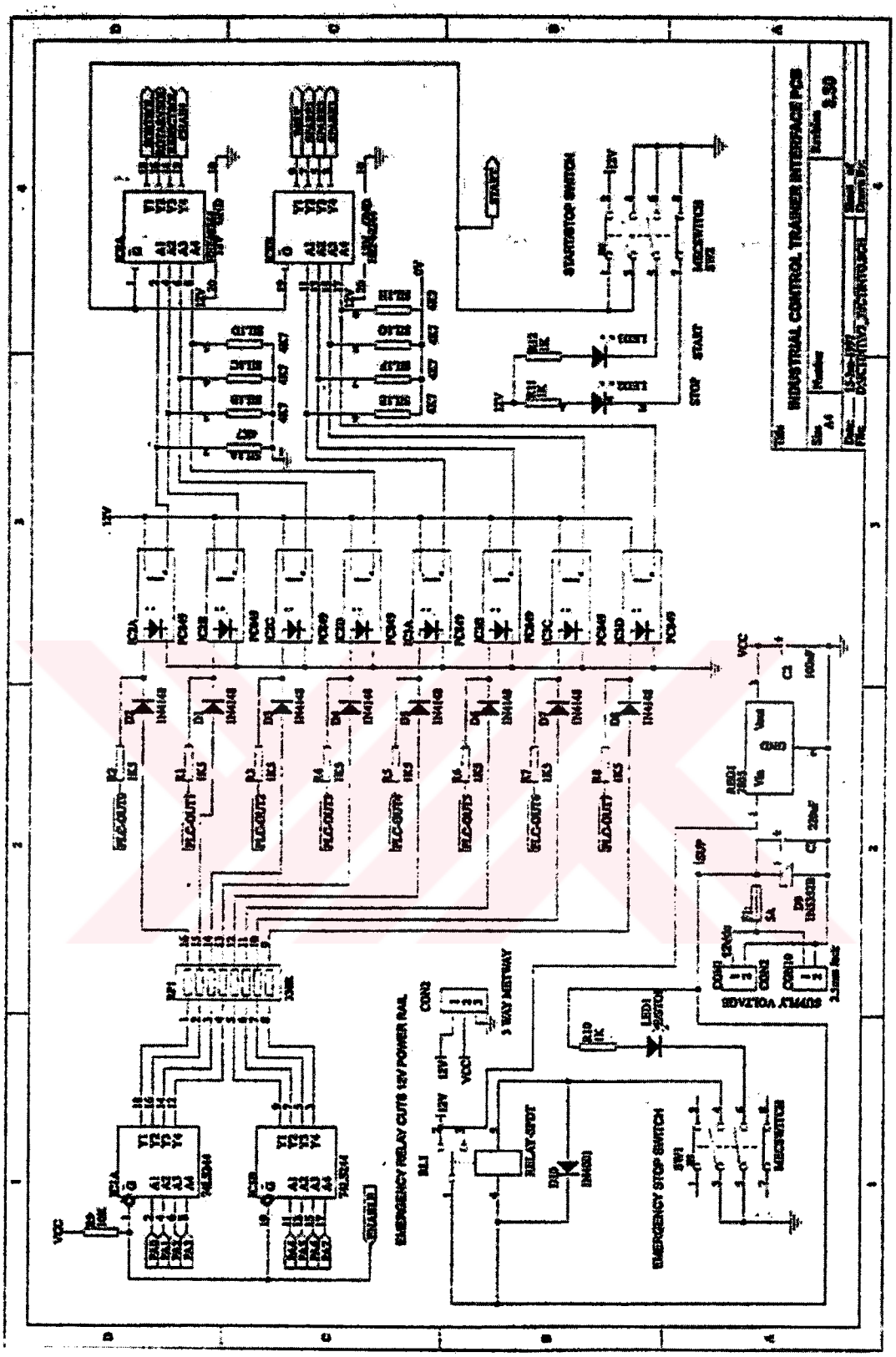


Figure 6.1. (continued) ICT Interface Board Circuit Diagram

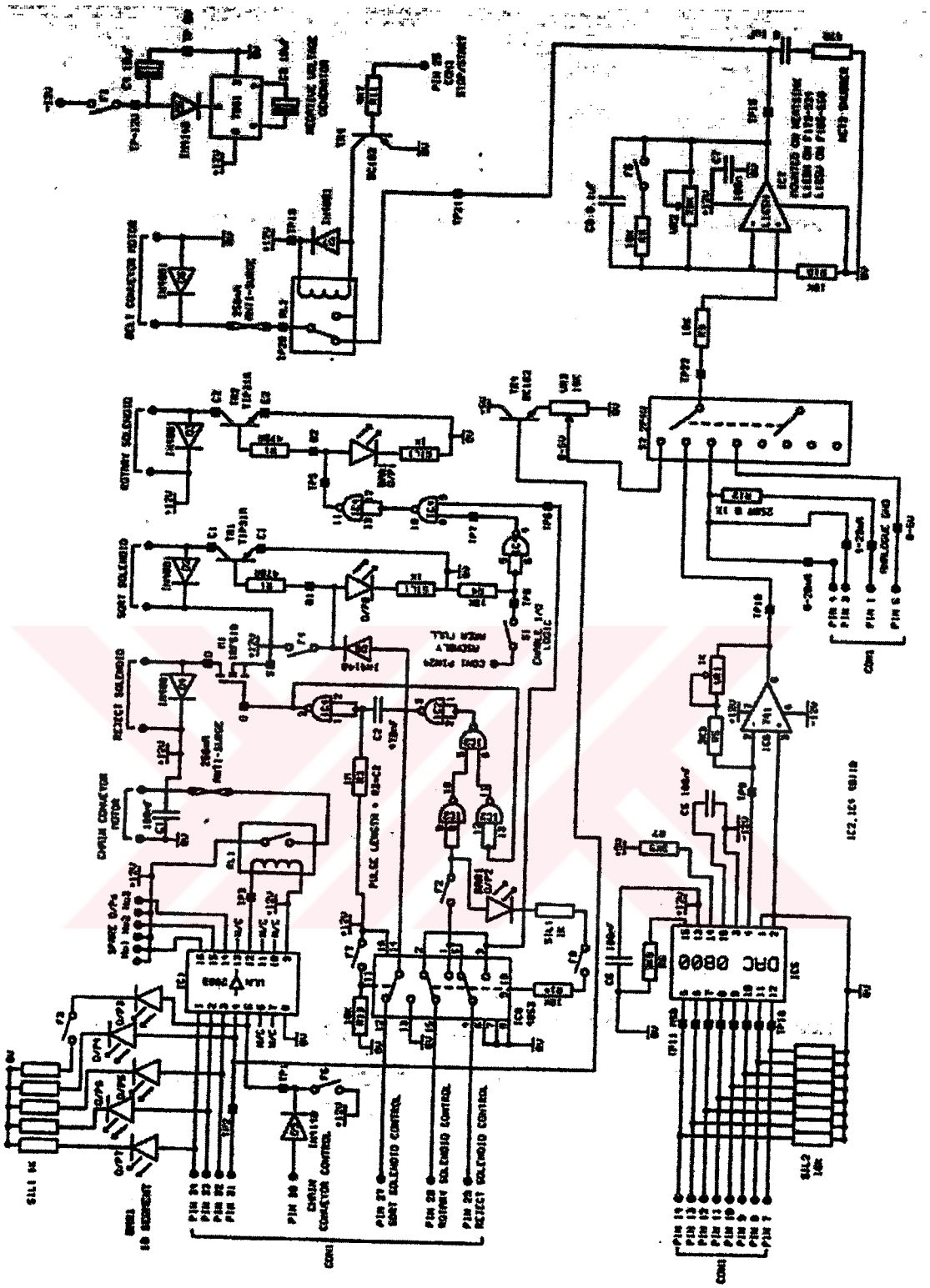


Figure 6.2. Output Driver Board Full Circuit Diagram

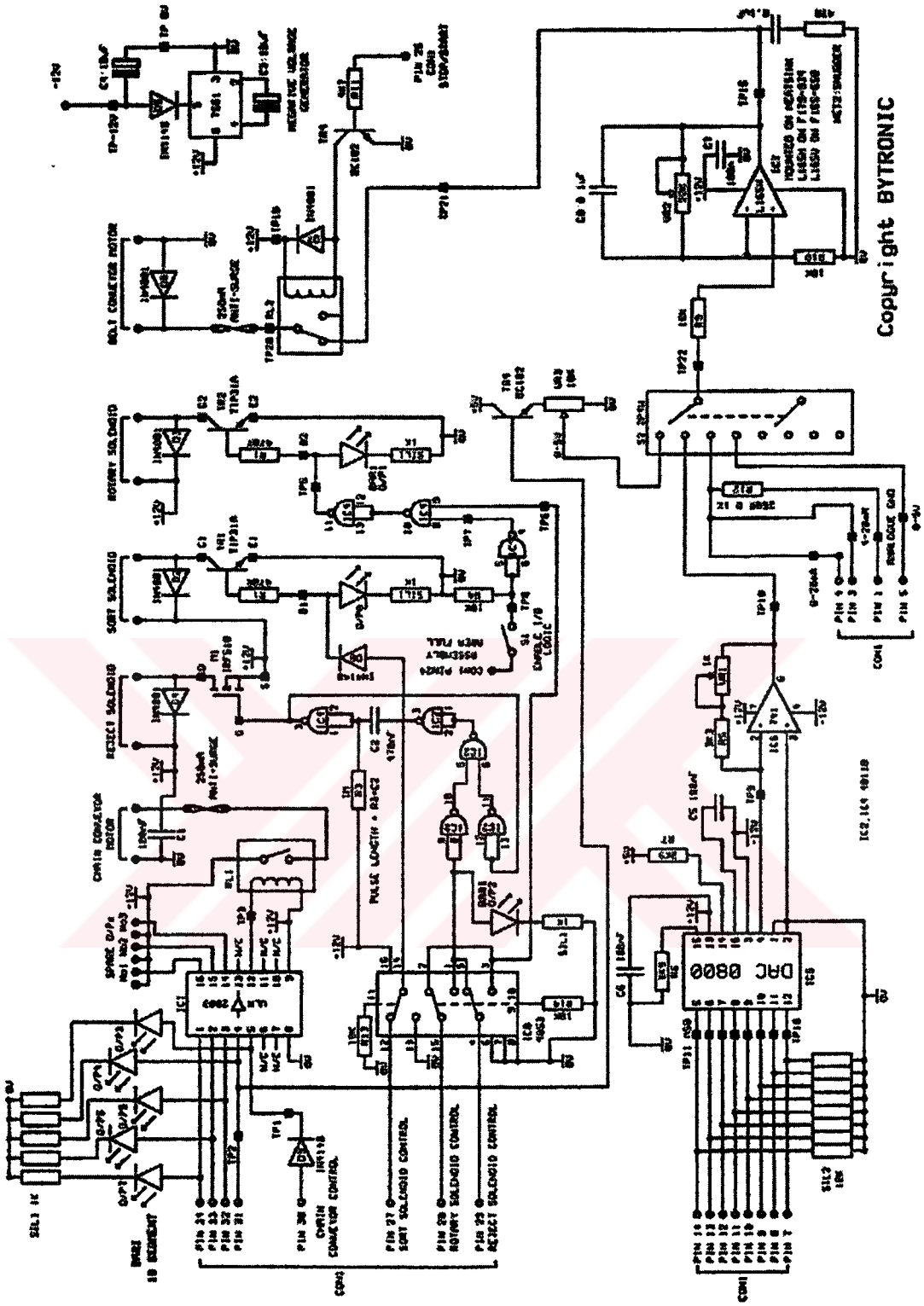


Figure 6.3. Output Driver Board without Switched Faults

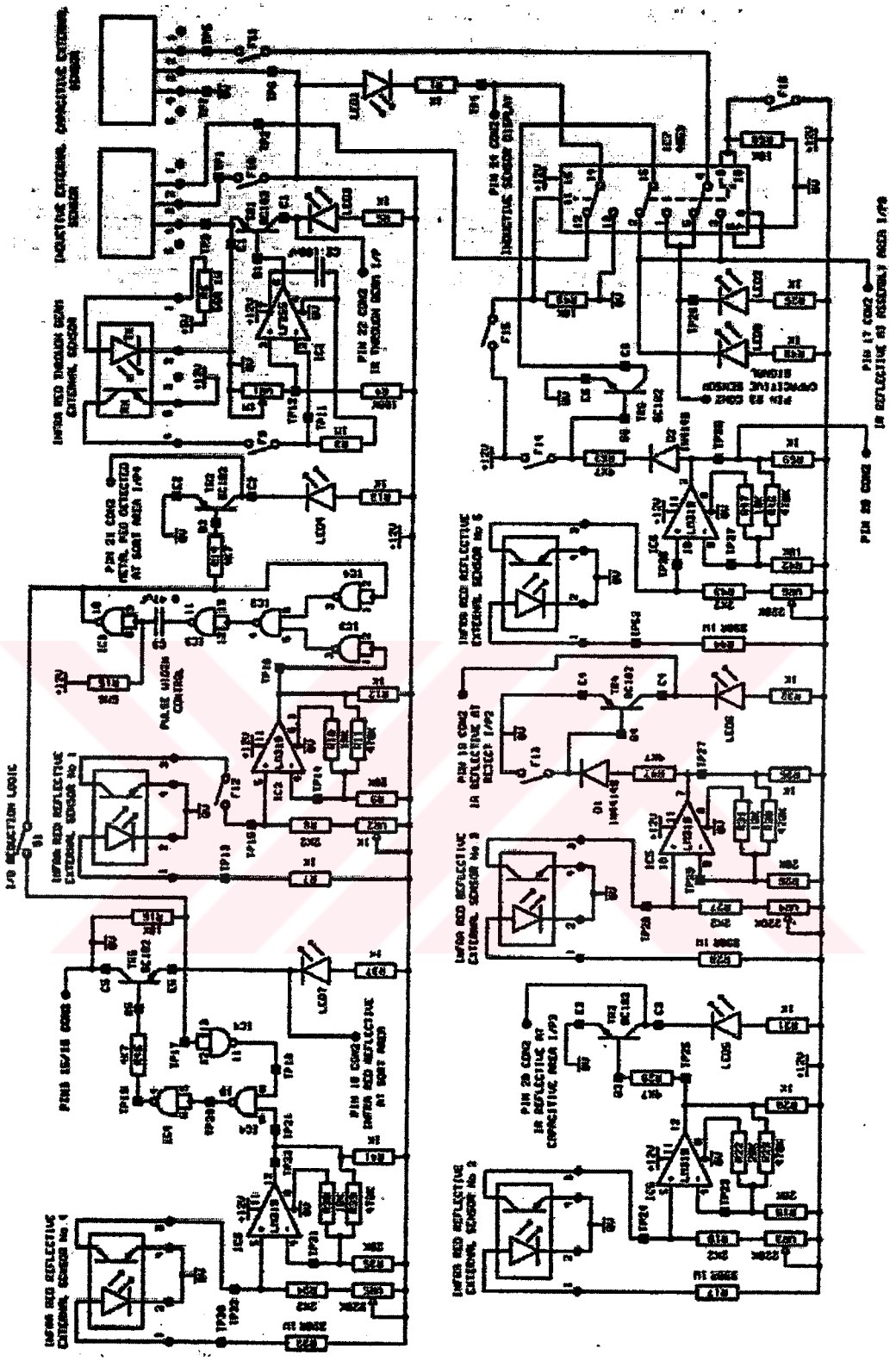


Figure 6.4. Signal Conditioning Board Full Circuit Diagram

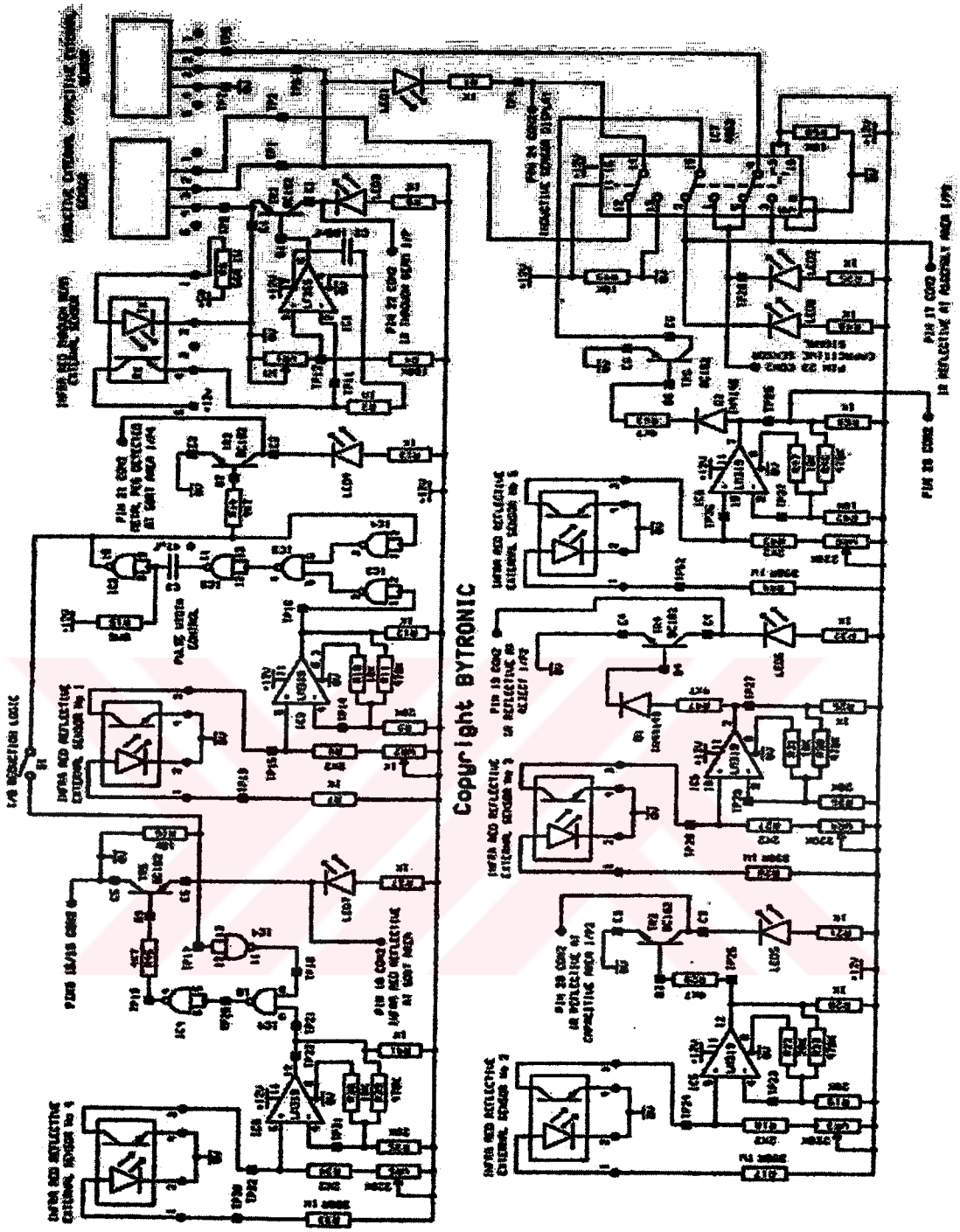


Figure 6.5. Signal Conditioning Board without Switched Faults