

T.C.
NİĞDE ÜNİVERSİTESİ
MÜHENDİSLİK-MİMARLIK FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME ÖDEVİ

ÜÇ ODA BİR SALON BİR EV İÇİN HIRSIZ ALARMININ
GERÇEKLEŞTİRİLMESİ

HAZIRLAYAN
Cevdet Selçuk KAHYALAR

DANIŞMAN
Yrd. Doç. Dr. Murat UZAM

HAZİRAN – 2002
NİĞDE

TÜRKİYE CUMHURİYETİ
NİĞDE ÜNİVERSİTESİ MÜHENDİSLİK-MİMARLIK FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME ÖDEVİ DERSİ SINAV TUTANAĞI

ÖĞRENCİNİN:

Adı ve Soyadı : Cevdet Selçuk KAHYALAR
Sınıf ve Numarası : EEM 4 (İ.Ö) 2980603011
Bitirme Ödevinin Adı : ÜÇ ODA BİR SALON BİR EV İÇİN HIRSIZ
ALARMININ GERÇEKLEŞTİRİLMESİ

Bitirme Ödevini Yürüten

Öğretim Elemanın Adı ve Soyadı : Yrd. Doç. Dr. Murat UZAM

BİTİRME ÖDEVİ SINAV JURİSİ

Başkan :
Üye :
Üye :
Üye :

Bu çalışma -----/-----/ 2002 tarihinde yapılan sınav sonucunda jürimiz tarafından oy birliği / çokluğu ile başarılı / başarısız bulunmuştur.

UYGUNDUR

-----/-----/-----

Elektrik – Elektronik Mühendisliği

Bölüm Başkanı

Doç. Dr. Saadetdin HERDEM

ÖZET

3 (ÜÇ) ODA 1 (BİR) SALON BİR EV İÇİN HIRSIZ ALARMININ GERÇEKLEŞTİRİLMESİ

KAHYALAR, Cevdet Selçuk

Danışman: Yrd. Doç. Dr. Murat **UZAM**

Haziran 2002

Bu çalışmada PIC 16F877 mikrodenetleyicisi kullanılarak üç oda bir salon evin güvenlik sistemi gerçekleştirilmiştir ve hırsız alarm sistemi uygulaması için maket ev yapılmıştır. Bu uygulamada sistem kontrolünde PIC 16F877 kullanılması nedeniyle tasarımı kolay, kullanımı kolay ve maliyeti düşük bir hırsız alarm sistemi gerçekleştirilmiştir. Bir hırsız alarm sisteminde bulunması gereken özellikler göz önüne alınarak program geliştirilmiş olup bu program PIC 16F877 mikrodenetleyicisi içerisine yüklenmiştir.

TEŞEKKÜR

Niğde Üniversitesi, Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümünde 2001-2002 eğitim-öğretim yılında bitirme ödevi olarak hazırlanan bu çalışma da hiçbir yardımını esirgemeyen danışman hocam **Yrd. Doç. Dr. Murat UZAM**'a, bana verdiği manevi destekten dolayı **Öğr. Gör. İbrahim YÜCEL**'e, proje çalışmalarına verdikleri destekten dolayı **Arş. Gör. Kürşat YALÇIN**'a ve **Durmuş GÖKÇEBAY**'a **teşekkürlerimi sunarım.**

Hayatım boyunca bana maddi ve manevi her türlü desteği veren, bütün düşünce ve eylemlerimi büyük bir özveri ve fedakarlıkla karşılayan, benim için her şeyden fazla önem taşıyan **aileme teşekkür ve şükranlarımı sunarım.**

İÇİNDEKİLER

	Sayfa
ÖZET.....	iii
TEŞEKKÜR.....	iv
İÇİNDEKİLER DİZİNİ.....	v
TABLolar DİZİNİ.....	vii
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ.....	ix
BÖLÜM I GİRİŞ	1
BÖLÜM II ELEMANLAR VE YÖNTEM	2
2.1. Pic Mikrodenetleyicisi	2
2.2. Program Geliştirme Yazılımı (PIC BASIC PRO)	2
2.2.1 PIC BASIC PRO Komut Seti	3
2.2.2 Değişkenler (VARIABLES)	4
2.2.3 Etiketler (LABELS)	4
2.2.4 Nümerik Sabitler (NUMERIC CONSTANTS)	4
2.2.5 Portlar	4
2.2.6 Zaman Gecikmesi (PAUSE)	5
2.2.7 Döngü Düzenlemek (FOR...NEXT)	6
2.2.8 Kontrol Komutları (IF... THEN)	6
2.2.9 Kesmeler (INTERRUPTS)	7
2.2.9.1 INTCON Register	8
2.2.9.2 Kesme Kaynakları	9
2.2.9.3 TMR0 Sayıcı/Zamanlayıcısı	9
2.2.9.4 OPTION Register	9
2.2.9.5 Prescaler Kullanılması	10
2.2.10 Microcode Studio	12
2.3. Programlama Yazılımı	12
2.4. Programlama Devresi	13
2.5. Kişisel Bilgisayar	14
2.6. Uygulama Devresi	14

2.7. Güvenlik Sisteminin Kurulacağı Maket Ev.....	14
BÖLÜM III PIC 16F877 MİKRODENETLEYİCİSİNE AİT TEMEL BİLGİLER.....	17
3.1. PIC 16F877.....	17
3.2. PIC 16F877 Mikrodenetleyicisinin I/O (Giriş/Çıkış) Portları.....	17
3.3. PIC 16F877 Mikrodenetleyicisinin Besleme ve Genel Çalışma Bağlantısı...18	
BÖLÜM IV HIRSIZ ALARM SİSTEMİNİN ÇALIŞMA PRENSİBİ.....	20
BÖLÜM V HIRSIZ ALARM SİSTEMİ İÇİN PROGRAM DEVRE UYGULAMALARI.....	22
5.1. Uygulama Devresi.....	22
5.2. Program Akış Şeması.....	23
BÖLÜM 6 SONUÇ ve ÖNERİLER.....	25
EKLER.....	26
EK-A. Bu Uygulama İçin Hazırlanmış .BAS Dosyaları.....	26
KAYNAKLAR.....	37

TABLolar DİZİNİ

	Sayfa
Tablo 1. KONNEKTÖR BACAk BAĞLANTI NUMARALARI.....	16
Tablo 2. PIC 16F877 ve PIC 16F84 MİKRODENETLEYİCİLERİN ÖZELLİKLERİ...17	17
Tablo 3. OSİLATÖR ÇEŞİTLERİ ve ÖZELLİKLERİ.....	19

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1. KESME İŞLEMİNİN SEMBOLİK OLARAK GÖSTERİMİ.....	7
Şekil 2. MICROCODE STUDIO ANA SAYFASI.....	12
Şekil 3. PROPIC MENÜSÜ.....	13
Şekil 4. PIC PROGRAMLAMA DEVRESİ.....	13
Şekil 5. 12 V DC GÜÇ KAYNAĞI DEVRE ŞEMASI.....	14
Şekil 6. 3*4 MATRİS TUŞ TAKIMI ŞEMATİĞİ ve PIC'e BAĞLANTISI.....	15
Şekil 7. BUZZER BAĞLANTI ŞEMASI.....	15
Şekil 8. HIRSIZ ALARM SİSTEMİ İÇİN YAPILAN MAKET EV.....	15
Şekil 9. PIC 16F877 PİN ŞEMATİĞİ.....	18
Şekil 10. PIC 16F877 GENEL ÇALIŞMA DEVRESİ.....	19
Şekil 11. UYGULAMA DEVRESİ.....	22
Şekil 12. PROGRAM AKIŞ ŞEMASI.....	24

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 1. PIC BASIC PRO KOMUT SETİ.....	3
Çizelge 2. İNTCON REGİSTER.....	8
Çizelge 3. OPTİON REGİSTER.....	10
Çizelge 4. PRESCALER DEĞERLERİ.....	11

BÖLÜM I

GİRİŞ

Yapılan son arařtırmalar ve istatistikler, hırsızlık ve soygunların her geen gün hızla arttığını ortaya koymaktadır. Günümüzde sadece lüks evler ve villalar deęil, bütün evler ve işyerleri soyulma riski altındadır. Eviniz ve eşyalarınız sigortalı da olsa bir hırsızın evinize ya da işyerinize vereceęi hasar, hele siz evinizdeyken başınıza gelebilecek bir hırsızlık girişimi, sigortadan karşılanamayacak zararlara neden olabilmektedir.

Evinizi ve işyerinizi hırsızlığa karşı korumanın en etkili yolu bir hırsız alarm sistemi kurdurmaktır. Ama bu sistemlerin fiyatlarından dolayı bir hırsız alarm sistemi kurdurmak külfetlidir. Bu nedenle hırsız alarm sistemini daha az maliyetle ve daha az elemanla uygun olarak gerçekleştirebilmek için bu uygulama yapılmıştır.

Bu tez çalışmasında, mikrodenetleyici kullanarak sistem kontrolü gerçekleştirilmiştir. Mikrodenetleyici olarak PIC 16F877 kullanılmıştır. PIC, mikrodenetleyicisinin bu uygulamada kullanılması ile, piyasadaki eş sistemlere göre daha ucuz ve daha az yer kaplayan kullanımı kolay ve az elemanla gerçekleştirilen bir hırsız alarm sistemi meydana gelmektedir. Bu çalışmada bir PIC (Peripheral Interface Controller) mikrodenetleyicisi için program yazmak ve geliřtirmek üzere, “PIC BASIC” gibi yüksek seviyeli program kullanılmıştır.

Bölüm II’de bu uygulamayı gerçekleřtirmek için gerekli eleman ve yöntemlerden bahsedilmiştir. Bölüm III’de bu uygulamada kullanılan PIC 16F877 ye ait temel bilgiler ve bağlantı şemaları verilmiştir. Bölüm IV’de hırsız alarm sisteminin ve sensörlerin çalışma prensibinden ve özelliklerinden bahsedilmiştir. Bölüm V’de ise bu uygulamada PIC’e yüklenerek hırsız alarm sisteminin çalışma şeklini belirleyen ana program verilmiştir. Bölüm VI’da ise sonuç ve öneriler bulunmaktadır.

BÖLÜM II

ELEMANLAR VE YÖNTEM

Bu tez çalışmasında bir güvenlik sistemi için kullanılmış olan tüm elemanları şu şekilde sıralayabiliriz.

- 1) PIC mikrodenetleyicisi (PIC 16F877),
- 2) Program geliştirme yazılımı (PIC BASIC PRO),
- 3) Programlama yazılımı (PROPIC),
- 4) Programlama devresi,
- 5) Kişisel bilgisayar,
- 6) Uygulama devresi,
- 7) Güvenlik sisteminin kurulacağı maket ev.

Bu elemanları inceleyecek olursak;

2.1. PIC Mikrodenetleyicisi

Güvenlik sisteminin kontrolünü sağlamak amacıyla PIC 16F877 mikrodenetleyici kullanılmıştır. Bu mikrodenetleyici;

- Güvenlik sistemi için yeterli sayıda I/O (giriş/çıkış) ucuna sahiptir.
- 1000 defa programlanıp silinebilen EEPROM (Flash) belleğe sahiptir.
- TMR0, TMR1, TMR2 yazılım kesmesine sahiptir.
- 4-20 Mhz arası yüksek bir çalışma frekansına sahiptir.

2.2. Program geliştirme yazılımı (PIC BASIC PRO)

Bu tezde PIC mikrodenetleyicisinin programlanmasında “**BASIC**” dili kullanılmıştır. Bu dilde program geliştirmek için “**PIC BASIC PRO**” programı kullanılmıştır. Üst seviyeli programlama dilleri (Örn: PIC BASIC PRO) sayesinde çok daha rahat ve hızlı bir şekilde programlama yapılabilir. Ancak bu programlar çok karmaşık

sistemlerin çözümünde yetersiz kalabilmektedir. Bu durumda yüksek seviyeli bir PIC programlama dili ve PIC assembly dili birlikte kullanıldığında bu sorun ortadan kalkmaktadır. Yüksek seviyeli bu programlama dilinin (PIC BASIC PRO) yararları bu uygulamada görülmüştür.

2.2.1 PIC BASIC PRO Komut Seti

PIC BASIC'in komut seti BASIC programlama dilinin komut setinden alınmıştır. BASIC programlama dilinde olmayan komutlar da mevcuttur. Çizelge 1.'de PIC BASIC PRO komutları bulunmaktadır.

Çizelge 1. PIC BASIC PRO komut seti

@	END	NAP	SELECT CASE
ADCIN	FOR...NEXT	ON DEBUG	SERIN
ASM...ENDASM	FREQOUT	ON INTERRUPT	SERIN2
BRANCH	GOSUB	OUTPUT	SEROUT
BRANCHL	GOTO	OWIN	SEROUT2
BUTTON	HIGH	OWOUT	SHIFTIN
CALL	HPWM	PAUSE	SHIFTOUT
CLEAR	HSERIN	PAUSEUS	SLEEP
CLEARWDT	HSEROUT	PEEK	SOUND
COUNT	I2CREAD	POKE	STOP
DATA	I2CWRITE	POT	SWAP
DEBUG	IF...THEN	PULSIN	TOGGLE
DEBUGIN	INPUT	PULSOUT	USBIN
DISABLE	LCDIN	PWM	USBINIT
DESABLE DEBUG	LCDOUT	RANDOM	USBOUT
DISABLE INTERRUPT	LET	RCTIME	WHILE...WEND
DTMFOUT	LOOKDOWN	READ	WRITE
EEPROM	LOOKDOWN2	READCODE	WRITECODE
ENABLE	LOOKUP	RESUME	XIN
ENABLE DEBUG	LOOKUP2	RETURN	XOUT

PIC BASIC PRO programlama dilinin tüm yönleriyle açıklanması bir kitap konusu olacağından burada sadece bu çalışma için gerekli olan komutlar ve bunların özelliklerinden bahsedilmiştir.

2.2.2 Değişkenler (Variables)

PIC BASIC PRO programlama dilinde üç tip değişken tipi kullanılmaktadır. Değişkenler BİT, BYTE veya WORD büyüklüğünde olabilmektedir.

Örneğin: **K VAR BYTE** (Byte büyüklüğünde K değişkeni)

2.2.3 Etiketler (Labels)

Etiketler GOTO veya GOSUB gibi komutlara referans teşkil etmektedirler. Bu komutların icrasından sonra program etiketle belirtilen satıra dallanmaktadır. Etiketler herhangi bir kelime olabilir ve sonuna iki nokta üst üste (:) konulmalıdır.

Örneğin: Etiket: **HIGH PORTB.5**

2.2.4 Nümerik sabitler (Numeric constants)

Nümerik sabitler üç şekilde tanımlanabilmektedir. Bunlar decimal, binary ve hexadecimal'dır. Binary değer tanımlanırken değer başına '%' ve hexadecimal değer başına '\$' işareti konulmaktadır. Decimal değer tanımlanırken herhangi bir işaret kullanılmamaktadır.Örneğin:

100 : Decimal değer .

%100 : Decimal 4 için binary değer.

\$100 : Decimal 256 için hexadecimal değer.

2.2.5 Portlar

PIC'in herhangi bir portunun istenilen pini giriş yada çıkış olarak tanımlanabilmektedir. Daha sonra istenilen değer ilgili porta yüklenilebilir. Değerin porta'ya yüklenmesi için farklı yollar vardır.

Örneğin:

TRISB = 0 : Portb'nin tüm pinleri çıkış olarak tanımlanmaktadır.

TRISA=1 : Portb'nin tüm pinleri giriş olarak tanımlanmaktadır.

TRISC = %11110000 : Portb'nin ilk dört biti çıkış, son dört biti giriş olarak tanımlanmaktadır.

Yönlendirilen port pinlerine bilgilerin gönderilmesi için birkaç yol vardır.

Örneğin:

PORTB =%11110000 : Portb'nin ilk dört pinine lojik-0 son dört pinine ise lojik-1 değeri gönderilmektedir .

PORTB = 1 : Portb'nin bütün pinlerine lojik-1 değeri gönderilmektedir.

HIGH PORTB.4 :PORTB'nin dördüncü pinine lojik-1 değeri gönderilmektedir.

LOW PORTB.6 : PORTB'nin altıncı pinine lojik-0 değeri gönderilmektedir.

2.2.6 Zaman gecikmesi (Pause)

'PAUSE', programı istenilen bir süre beklemeye almaktadır. En fazla 65535 milisaniyelik bir gecikme sağlanabilmektedir. Bu da bir dakikanın biraz üstünde bir değerdir. 'PAUSE' komutundaki değer milisaniye büyüklüğündedir. Örneğin, PAUSE 49000 (49 saniyelik gecikme).

Eğer 49 saniyelik gecikmeyi assembly kodunda yazmak gerekseydi, şu şekilde yazılmalıydı.

```
LIST = P16F877
INCLUDE "P16F877.INC"
SAYI EQU H'20'
SAYI1 EQU H'21'
SAYI2 EQU H'22'
GECIKME
    MOVLW H'FF'
    MOVWF SAYI
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
```

Oysa bunu PIC BASIC PRO programlama dilinde PAUSE 49000 şeklinde tek satırda gerçekleştirmek mümkün olmaktadır. Görüldüğü üzere PIC BASIC PRO ile program yazmak hem çok rahat hem de çok basit olmaktadır.

2.2.7 Döngü düzenlemek (For...Next)

Bazı işlemlerin önceden belirlenen sayıda tekrarlanması gerekmektedir. PIC BASIC PRO programlama dilinde bu işi 'FOR...NEXT' döngüsü bir sayaç gibi çalışarak gerçekleştirmektedir. 'FOR' komutundan sonra bir değişkene ilk değer atanıp, bu değişkenin sayacağı son değer de belirtilmelidir. Ayrıca değişkenin kaçar kaçar artacağı 'STEP' komutuyla kontrol edilebilmektedir. Son olarak 'NEXT' döngüsüne gelen program eğer son değere ulaşılmamış ise döngüsüne devam edecektir. Örneğin:

```
FOR K=2 to 10 STEP 2  
PAUSE 1000  
NEXT
```

Burada k değişkeni 2'den 10'a kadar ikişer ikişer saymaktadır. Her bir döngüde bir saniyelik bir gecikme sağlanmaktadır. Döngü toplam beş tur atacağından 5 saniyelik bir gecikme sağlanmış olacaktır.

2.2.8 Kontrol komutları (If...Then)

Bazı işlemlerde bir durumun kontrol edilmesi gerekebilir. PIC BASIC PRO programlama dilinde bu işi 'IF...THEN' komutu sağlar. Bu kontrol deyimi bir veya

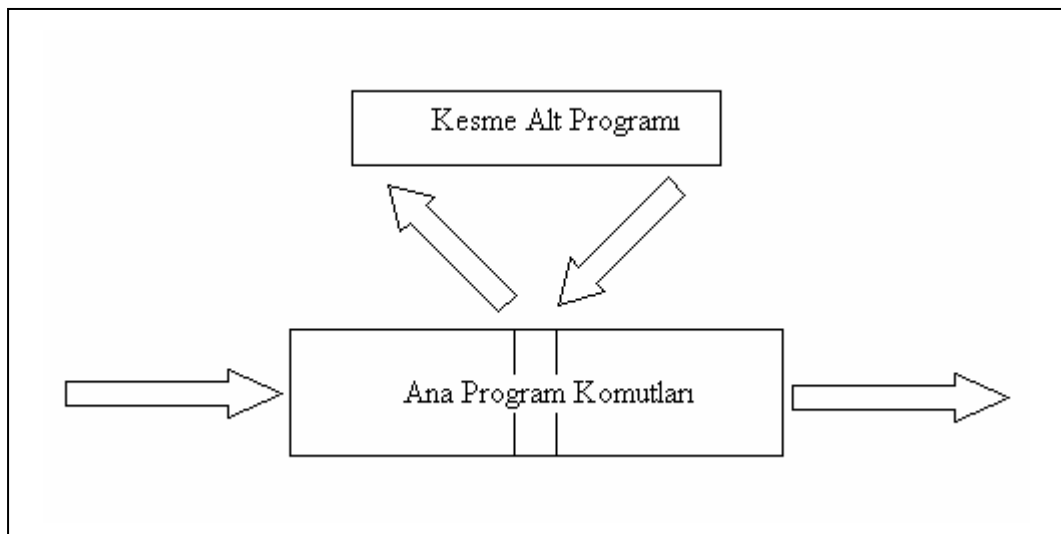
birden çok karşılaştırmayı icra edebilir. 'IF...THEN' kontrolörü bir durumun doğru (true) veya yanlış (false) olmasını karşılaştırarak değerlendirir. Eğer değerlendirme doğru ise 'THEN' komutundan sonraki ifade işlenir. Eğer değerlendirme yanlış ise program bir sonraki satıra geçecektir. Örneğin:

```
IF PORTB.2 = 0 THEN HIGH PORTA.1  
ELSE HIGH PORTB.3
```

Bu program parçasında eğer portb'nin ikinci pini lojik-0 olmuşsa porta'nın birinci pini lojik-1 yapılır aksi taktirde portb'nin üçüncü pini lojik-1 olacaktır.

2.2.9 Kesmeler (Interrupts)

PIC'in port girişlerinden veya donanım içerisindeki bir sayıcıdan gelen sinyal nedeniyle belleğinde çalışmakta olan programın kesilmesi olayına kesme denilmektedir. Programın kesildiği andan itibaren önceden hazırlanan bir alt program çalışmaktadır. Alt program işlevini bitirdikten sonra ana program kaldığı yerden itibaren çalışmasına devam etmektedir. Netice olarak bir kesme, ana program çalışmasını sadece duraklatır, ama hiçbir zaman işlevinin devam ettirmesini engellememektedir. Interrupt alt programları kullanarak, program içerisinde kullanılacak komut sayısı azaltılıp bir sürü mantıksal karışıklıklar önlenmektedir. Kesme işlemi Şekil 1.'de sembolik olarak gösterilmiştir.



Şekil 1. Kesme işleminin sembolik olarak gösterimi

2.2.9.1 INTCON Register

INTCON (İnterrupt Control) registeri RAM bellekte h'18B' adresinde bulunan özel registerden bir tanesidir. Bu register içerisinde her bir kesme kaynağı için bir flag ve bir de global kesme bulunmaktadır..Çizelge 2.'de intcon registerin her bir bitinin ne işe yaradığı gösterilmiştir.

Çizelge 2. İntcon register

GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
7	6	5	4	3	2	1	0
<p>GIE : Tüm kesme işlemlerini iptal etme bayrağı 0: Tüm kesmeler geçersiz 1: Aktif yapılmış olan tüm kesmeler geçerli</p> <p>EEIE : EEPROM belleğe yazma işlemi tamamlama kesmesi 0: Geçersiz 1: Geçerli</p> <p>TOIE : TMR0 sayıcı kesmesini aktif yapma bayrağı 0: Geçersiz 1: Geçerli</p> <p>INTE : Harici kesmeyi aktif yapma bayrağı 0: Geçersiz 1: Geçerli</p> <p>RBIE : PORTB(4, 5, 6, 7.bitleri) değişiklik kesmesini aktif yapma bayrağı 0: Geçersiz 1: Geçerli</p> <p>TOIF : TMR0 sayıcısı zaman aşımı bayrağı 0: Zaman aşımı yok 1: Zaman aşımı var</p> <p>INTF : Harici kesme bayrağı 0: Harici kesme oluşmadığında</p> <p>RBIF : PORTB değişiklik bayrağı 0: RB4~RB7 uçlarında değişiklik yok 1: RB4~RB7 uçlarından en az birisinde değişiklik var.</p>							

2.2.9.2 Kesme Kaynakları

Kesme birkaç yoldan yapılabilir. Bunlardan bazıları:

- Harici(external) kesme.
- TMR0 sayıcısında oluşan zaman aşımı kesmesi.
- PORTB'deki (4, 5, 6, 7 bitler) lojik seviye değişikliğinden.
- EEPROM belleğe yazma işleminin tamamlanmasında meydana gelen kesme.

Bu çalışmada kullanılan kesme TMR0 sayıcısından oluşan zaman aşımı kesmesidir.

2.2.9.3 TMR0 Sayıcısı/Zamanlayıcısı

PIC 16F877'nin RAM belleğinin h'101' adresinde TMR0 adı verilen özel bir register vardır. TMR0 programlanabilen bir sayıcıdır. Yani saymaya istenilen bir değerden veya baştan başlatılabilir. Herhangi bir anda içeriği sıfırlanabilir. Belli başlı özellikleri şunlardır.

- 8-bit bir sayıcıdır.
- Yazılabilir/okunabilir.
- Programlanabilen frekans bölme değeri (prescaler value) vardır.
- Sayı artışı harici veya dahili clock saykılı ile yapılabilir.
- Düşen ve yükselen kenar tetikleme (harici olarak)
- Sayıcı değeri artan yöndedir.
- TMR0'ın değeri h'FF'den h'00'a geldiğinde ilgili flag'i "1" yaparak kesme oluşturur.

TMR0 sayıcısının önemli özelliklerinden biri de ana program veya kesme alt programları çalışırken sayma işlemini durdurmamasıdır.

2.2.9.4 OPTION Register

OPTION register, RAM belleğin h'81'adresinde bulunan özel bir registerdir. TMR0 sayıcısının kontrolünde kullanılmaktadır. Çizelge 3.'de bu registerin her bir bitinin ne görevler yaptığı açıklanmıştır.

Çizelge 3. Option register

RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
7	6	5	4	3	2	1	0

PS0, PS1, PS2 :Frekans bölme sayısı

PSA :Frekans bölücü seçme biti
0:Frekans bölme sayısı TMR0 için geçerli
1:Frekans bölme sayısı WDT için geçerli

TOSE :TMR0 sinyal kaynağı kenar seçme biti
0:RA4/TOCKI ucundan düşen kenar tetiklemesi
1: RA4/TOCKI ucundan yükselen kenar tetiklemesi

TOCS :TMR0 sinyal kaynağı seçme biti
0:Dahili komut saykılı seçilir
1:Harici dijital sinyal(RA4/TOCKI ucu)

INTEDG :Harici kesme sinyali kenar seçme biti
0:RB0/INT ucundan düşen kenarda tetikleme
1: RB0/INT ucundan yükselen kenarda tetikleme

RBPU :PORTB pull up geçerli yapma biti
0:PORTB uçlarındaki pull-up'lar iptal edilir.
1: PORTB uçlarındaki pull-up'lar geçerli yapılır

2.2.9.5 Prescaler Kullanılması

Option registerin 0, 1, 2. bitleri (PS0~PS2) içerisine yerleştirilen sayılar, TMR0 veya WDT'ye uygulanan sinyali böler. Böylece sayma hızları değiştirilebilir. Üç bitlik bu sayı TMR0 veya WDT'de birbirinden farklı 8 farklı oran seçme olanağı oluşturur.

Çizelge 4.'de prescaler değerleri görülmektedir.

Çizelge 4. Prescaler değerleri

Frekans bölme sayısı	TMR0 Oranı	WDT oranı
000	1/2	1/1
001	1/4	1/2
010	1/8	1/4
011	1/16	1/8
100	1/32	1/16
101	1/64	1/32
110	1/128	1/64
111	1/256	1/128

TMR0 veya WDT sayıcılarının kaç dahili komut saykılında bir defa bir üst sayıya geçeceğini belirleyen orandır. Örneğin:

TMR0 oranı 1/2 ise, 2 komut saykılında bir defa üst sayıya geçiş olur.

TMR0 oranı 1/8 ise, 8 komut saykılında bir defa üst sayıya geçiş olur.

Program belleğine yerleştirilen komutların çalışabilmesi için harici bir osilatörden clock sinyali (f_{osc}) uygulanması gerekmektedir. Bu frekans PIC tarafından 4'e bölünerek OSC2 ucundan dışarıya verilmektedir. İşte 4'e bölünen bu frekansın bir saykılı bir komutun icrası için geçen süreyi belirlemektedir. Bu çalışmada 4 MHz'lik bir kristal osilatör kullanılmıştır. Bu frekanstaki dahili komut saykılı 1MHz'dir. Periyodu ise 1 μ s'dir. Yani dahili komut saykılı 1 μ s'dir. Bir komut 1 μ s'lik bir sürede icra edilmektedir. Prescaler değeri ile TMR0 sayıcısının kaç μ s aralıklarla saydığı veya kaç μ s aralıklarla kesme verdiği belirlenmektedir.

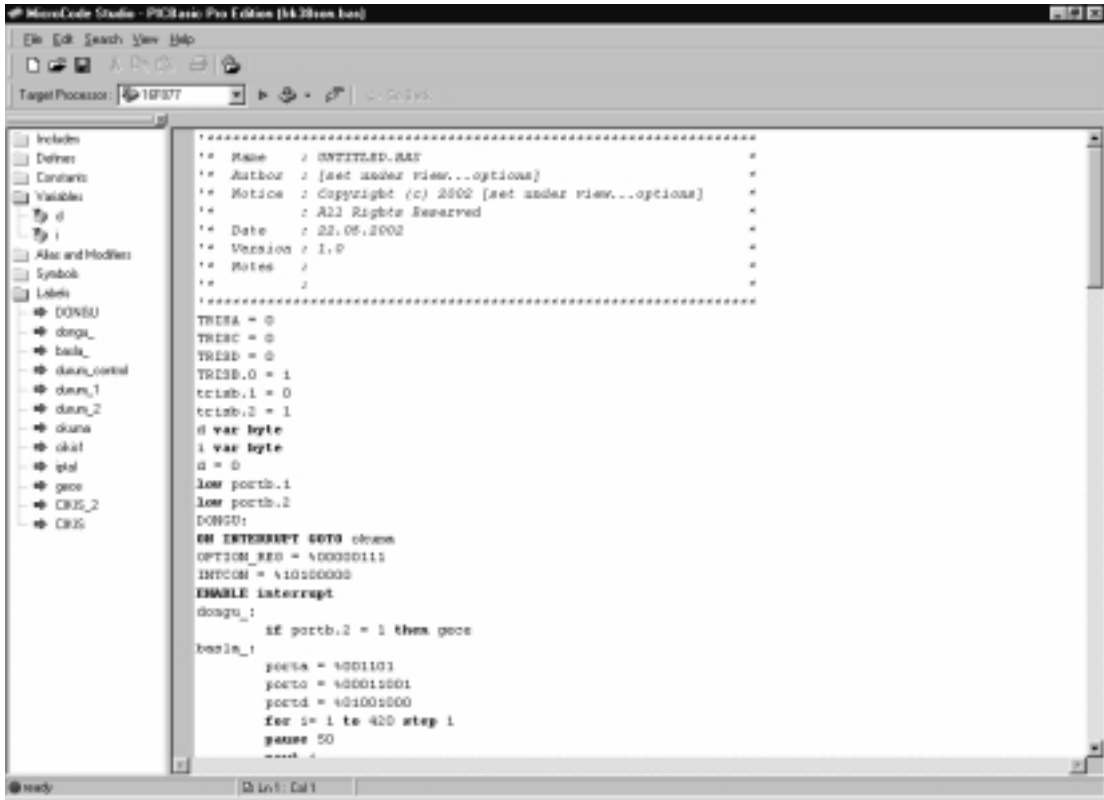
Örneğin, bu çalışmada prescaler değeri b'111' seçilmiştir. Prescaler b'111' olduğunda TMR0 oranı 1/256 olur. (f_{osc}) değeri 4MHz olduğundan, komut saykılı 1 μ s olmaktadır. Buradan TMR0 sayıcısının içeriği 256 dahili komut saykılında bir arttığı anlaşılmaktadır. TIMER0 sayma aralığı bu çalışma için 256 μ s'dir. TMR0 saymaya

başladığında ilk sayı h'00' olduğundan TIMER0;

$256 \mu s \times 256 = 65536 \mu s(65.5ms)$ aralıklarla kesme sinyali verecektir.

2.2.10 Microcode Studio

PIC BASIC PRO programlama dilinde yazılan bir program Microcode Studio programında derlenmektedir (compile). Herhangi bir editörde yazılan PIC BASIC PRO kodları Microcode Studio programında derlenebildiği gibi, bu programın kendine has editörü de bulunmaktadır. Şekil 2.'de Microcode Studio programının ana sayfası görülmektedir.



Şekil 2. Microcode Studio ana sayfası

2.3. Programlama yazılımı (PROPIC)

PRO PIC, PIC BASIC ile hazırlanıp derlenen Programın PIC mikrodenetleyicisine yüklenmesi için gereken yazılımdır. PIC'in belleğine yüklenmek üzere, Microcode Studio programında yazılan program derlenmiştir. Hexadecimal kodlar (derlenmiş program) bilgisayarın paralel portuna bağlanan programlayıcı (PROTOPIC) aracılığı ile PIC'in program belleğine yazılmıştır. PROTOPIC programlayıcısının PROPIC adında bir programı bulunmaktadır. Şekil 3.'de PROPIC'in menüsü görülmektedir.

2.5. Kişisel bilgisayar

Bütün bu işlemleri yapabilmek için hiç de yabancı olmadığımız günümüzde hemen hemen her evde, ofiste bulunan bir kişisel bilgisayar kullanılmıştır. Yukarıda belirtilen işlemleri yapabilmek için bilgisayarın minimum, Pentium 166 MMX işlemciye ve 10 MB boş hard disk alanına sahip olması gerekmektedir.

2.6. Uygulama devresi

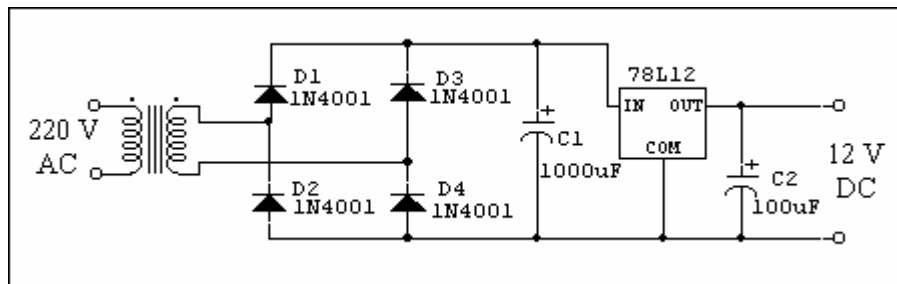
Uygulama devresi, enerji ve osilatör girişleri vs. gibi PIC' in çalışması için gerekli olan gereksinimleri üzerinde bulunduran ve PIC ile sensörler, tuş takımı gibi maket üzerindeki elemanlar arasında iletişimi sağlayan tarafımdan gerçekleştirilmiş olan Şekil 11.'de görülen devredir.

2.7. Güvenlik sisteminin kurulacağı maket ev

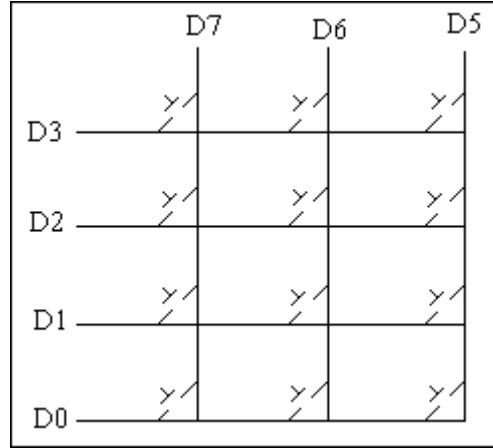
Hırsız alarmı kurulacak maket ev Şekil 8.'de gösterilmiş olup Niğde Üniversitesi, Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü Mikroişlemci laboratuvarında bulunmaktadır.

Şekil 8.'de gösterilen maket, kapı sensörlerinden, pencere sensörlerinden ve hareket sensörlerinden PIC'e bilgi aktarmak amacıyla yapılmıştır.

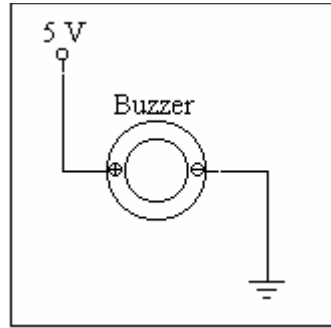
Maket'de Şekil 8.'den de görüleceği gibi 5 (beş) adet pencere sensörü, 1 (bir) adet kapı sensörü, 2 (iki) adet hareket sensörü, şifre girişi için Şekil 6.'da bağlantılarıyla birlikte görülen 1 (bir) adet tuş takımı, elemanların bağlantılarının yapıldığı ve bacak bağlantıları Tablo 1.'de verilmiş olan 25 (yirmi beş) pinli konnektör, alarm için Şekil 7.' da görülen buzzer, uygulama devresi ve sensör enerji beslemeleri için Şekil 5.'de görülen 12 V güç kaynağı bulunmaktadır.



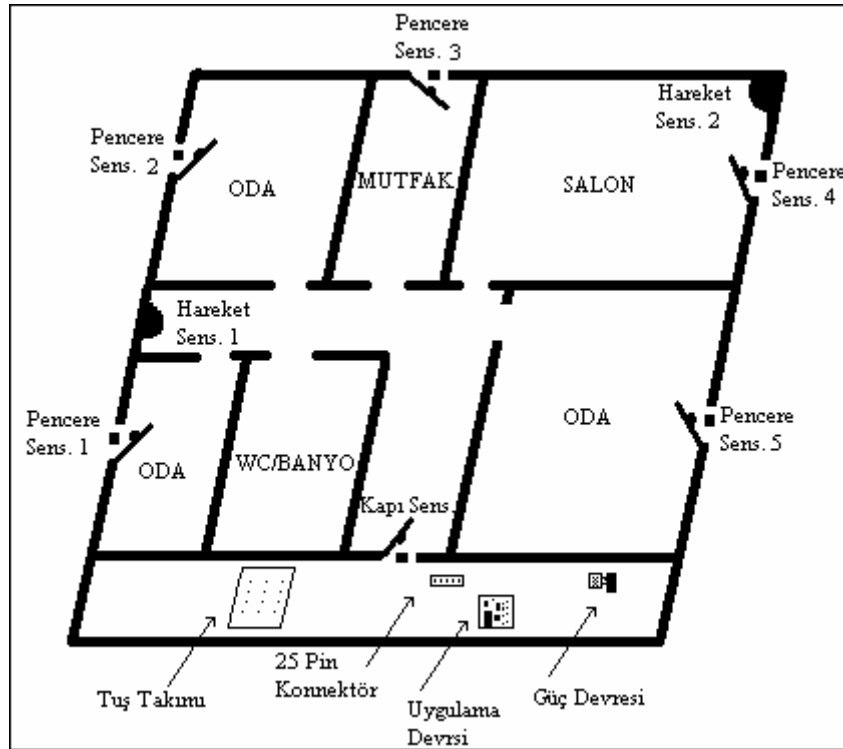
Şekil 5. 12 V DC güç kaynağı devre şeması



Şekil 6. 3*4 Matris tuş takımı şematığı ve PIC'e bağlantısı



Şekil 7. Buzzer bağlantı şeması



Şekil 8. Hırsız Alarm Sistemi İçin Yapılan Maket Ev

Tablo 1. Konnektör bacak bağlantı numaraları

Konnektör bacak bağlantı numarası	Maket elemanları bağlantı numarası	PIC 16F877 bacak bağlantı numarası
1	Tuş takımı	22 (D3)
2	Tuş takımı	30 (D7)
3	Tuş takımı	21 (D2)
4	Tuş takımı	29 (D6)
5	Tuş takımı	20 (D1)
6	Tuş takımı	28 (D5)
7	Tuş takımı	19 (D0)
8	Pencere sensörü 1	0 V
9	Pencere sensörü 1	-----
10	Hareket sensörü 1	-----
11	Hareket sensörü 1	-----
12	Pencere sensörü 2	-----
13	Pencere sensörü 2	-----
14	Kapı sensörü 1	5 V
15	Kapı sensörü 1	0 V
16	Alarm (Buzzer)	0 V
17	Alarm (Buzzer)	35 (B2)
18	Pencere sensörü 5	-----
19	Pencere sensörü 5	-----
20	Pencere sensörü 4	-----
21	Pencere sensörü 4	-----
22	Hareket sensörü 2	-----
23	Hareket sensörü 2	-----
24	Pencere sensörü 3	-----
25	Pencere sensörü 3	5 V

BÖLÜM III

PIC 16F877 MİKRODENETLEYİCİSİNE AİT TEMEL BİLGİLER

3.1. PIC 16F877

PIC 16F877 mikrodnetleyicisi motor kontrolü uygulamalarından, günümüz araçları, uzaktan çalışan sensörler gibi bir çok alanda kullanılmaktadır. Özellikle I/O (giriş/çıkış) sayısının fazlalığı, EEPROM (flash) belleğe sahip olmaları, geniş hafıza, çalışma frekansı, TMR yazılım kesmeleri gibi geniş özelliklere sahip olmalarından dolayı tercih edilmektedir.

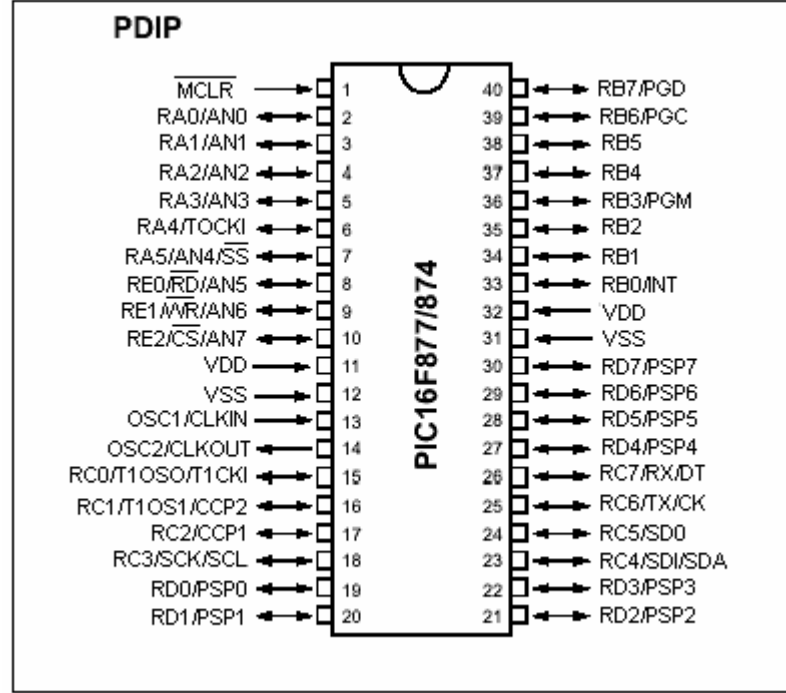
PIC16F877, belki de en popüler PIC işlemcisi olan 16F84'den sonra kullanıcılara yeni ve gelişmiş olanaklar sunmasıyla hemen göze çarpmaktadır. Program belleği FLASH ROM olan 16F877'de yüklenen program 16F84'de olduğu gibi elektriksel olarak silinip yeniden yüklenebilmektedir. Aşağıdaki Tablo 2.'de 16F877 ve 16F84 işlemcileri arasında özellik karşılaştırması yer almaktadır.

Tablo 2. PIC 16F877 ve PIC 16F84 mikrodnetleyicilerin özellikleri

ÖZELLİKLER	PIC16F877	PIC 16F84
Çalışma hızı	DC - 20 MHz	DC - 10 MHz
Program belleği	8Kx14 word Flash ROM	1Kx14 word Flash ROM
EEPROM Veri belleği	256 byte	64 byte
Kullanıcı RAM	368x8 byte	68x8 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 Compare 10 bit PWM çözünürlük	YOK
Seri çevresel arayüz	SPI(Master) ve I2C(Master/Slave) modunda SPI portu	YOK
Paralel slave port	8 bit, harici RD, WR ve CS kontrollü	YOK

3.2. PIC 16F877 Mikrodenetleyicisinin I/O (giriş/çıkış) portları

PIC 16F877 mikrodenetleyicisinde 40 (kırk) adet pin bulunmaktadır. Bu pinlerden 29 (yirmi dokuz) tanesi I/O olarak kullanılabilir. Bu I/O portundan 8 tanesi B portunu, 8 tanesi D portunu, 6 tanesi A portunu, 4 tanesi C portunu, 3 tanesi E portunu oluşturur. Şekil 9.'da PIC 16F877' nin pin görünüşü verilmiştir.



Şekil 9. PIC 16F877 Pin şematığı

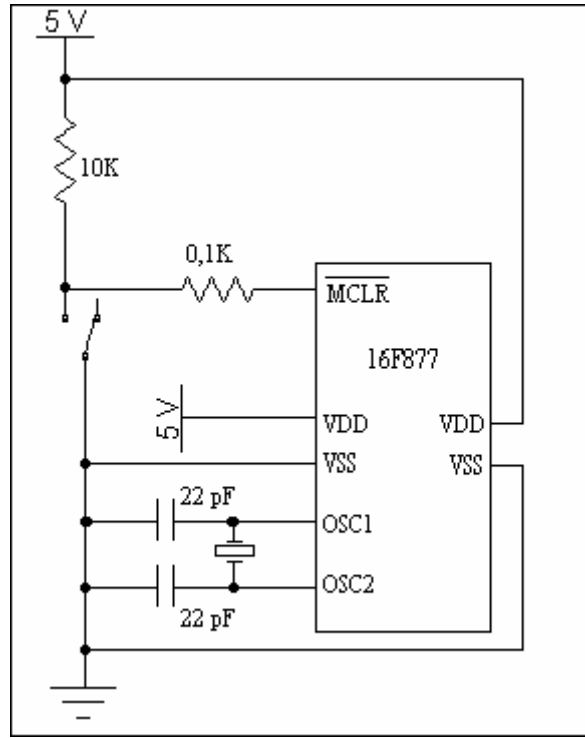
3.3. PIC 16F877 mikrodenetleyicisinin besleme ve genel çalışma bağlantıları

PIC 16F877 nin besleme gerilimi 11, 12, 31, 32 numaralı pinlerden oluşmaktadır. 11 ve 32 numaralı pinler Vdd uçları olup bu uçlara 5 V luk gerilim uygulanır. 12 ve 31 numaralı pinler Vss uçları olup bu uçlara toprağa bağlanır. PIC'e ilk enerji verildiği anda gerilim dalgalanmasını önlemek amacıyla 100 nF kondansatör Vss ile Vdd arasına bağlanır. Kullanıcının istediği anda programı kesip başa döndürmesi için PIC'in 1 nolu bacağı (MCLR) kullanılır. Programın başa dönmesi için MCLR ucunun 0 V olması gerekmektedir ve dolayısıyla programın normal olarak çalışması için MCLR ucunun 5 V olması gerekir. PIC'in çalışması için PIC'e clock girişi olmalıdır. Bu girişler 13 numaralı (OSC1) ve 14 numaralı (OSC2) uçlardır. Bu girişlere 4 çeşit osilatör tipinden biri seçilerek bağlantı yapılabilir. Osilatör çeşitleri Tablo 3.'de verilmiştir.

Tablo 3. Osilatör çeşitleri ve Özellikleri

Osilatör Tipi	Tanımı	Özelliği	Frekans
LP	Kristal osilatör, seramik rezonatör	Asgari akım	40 Khz
XT	Kristal osilatör, seramik rezonatör	Genel amaçlı	4 Mhz
HS	Kristal osilatör, seramik rezonatör	Yüksek hız	20 Mhz
RC	Direnç/Kapasitör zaman sabiti	Düşük maliyet	4 Mhz

PIC 16F877 için yukarıda bahsettiğimiz bağlantılar kristal osilatör kullanılarak Şekil 10. de verilmiştir.



Şekil 10. PIC 16F877' nin genel çalışma devresi

BÖLÜM IV

HIRSIZ ALARM SİSTEMİNİN ÇALIŞMA PRENSİBİ

Maket üzerinde bulunan kapı ve pencere sensörleri kapı ve pencereler kapalıyken kısa devre, kapı ve pencereler açıkken ise açık devre olurlar. Yani bu sensörlere 5 V luk bir gerilim uygulandığında bir uçtan uygulanan gerilimi olduğu gibi diğer uca iletirler fakat kapı veya pencerelerden herhangi biri açıldığında sensör kontağı açılacağından 5 V luk gerilimi iletmeyecektir. Hareket sensörleri de aynı mantıkla çalışmaktadırlar: hareket algılandığı zaman kapalı kontak açılarak enerjiyi kemektedir. Bu mantıkla bu uygulamada PIC mikrodenetleyicisinin komut alması ve program akışına göre alarm verme ya da alt programlara dallanma işlemi gerçekleştirilmektedir.

Bu hırsız alarm sisteminin basitçe çalışma şekli şöyledir;

- Maket üzerinde bulunan tuş takımından 5 (beş) haneli şifre girilerek enter (#) tuşuna basılarak alarm kurulacak (Set), aynı şifre tekrar girildiğinde ve enter (#) tuşuna basıldığı zaman alarm Reset olacak.
- Önce yıldız (*) tuşuna basıldığında eski şifre doğru olarak girildikten sonra 5 haneli yenişifre girilip enter tuşuna (#) basılırsa şifre değişir yani şematik olarak
(*) , “5 Haneli eski şifre” , (#) , “5 Haneli yeni şifre” , (#)
- Yıldız (*) tuşuna basıldıktan sonra yanlış şifre girildiğinde şifre değişimi gerçekleşmeyecek ve 5 (beş) defa hatalı şifre girilmesinden sonra alarm çalacak.
- Alarm set durumundayken pencere sensörlerinden veya hareket sensörlerinden hareket algılandığında alarm çalacak.

- Alarm Set durumundayken kapı sensöründen kapı açıldı bilgisi geldiğinde içeri giren kişinin doğru şifreyi girerek alarmı Reset etmesi için 3 (üç) dakika süresi vardır. Kişi bu süre içerisinde şifreyi doğru olarak giremediyse alarm çalacaktır. Kişi şifreyi doğru girerse alarm Reset olacaktır.
- Bu 3 (üç) dakika içerisinde diğer sensörlerden alarm çal bilgisi gelirse alarm 3 (üç) dakikalık sürenin dolmasını beklemeden alarm çalar.
- Hangi durumda olursa olsun kişinin şifreyi 5 (beş) defa yanlış girme hakkı vardır. Eğer kişi şifreyi 5 (beş) defa yanlış girerse hemen alarm çalar.
- Alarm çalarken şifre doğru girilirse alarm susacaktır.

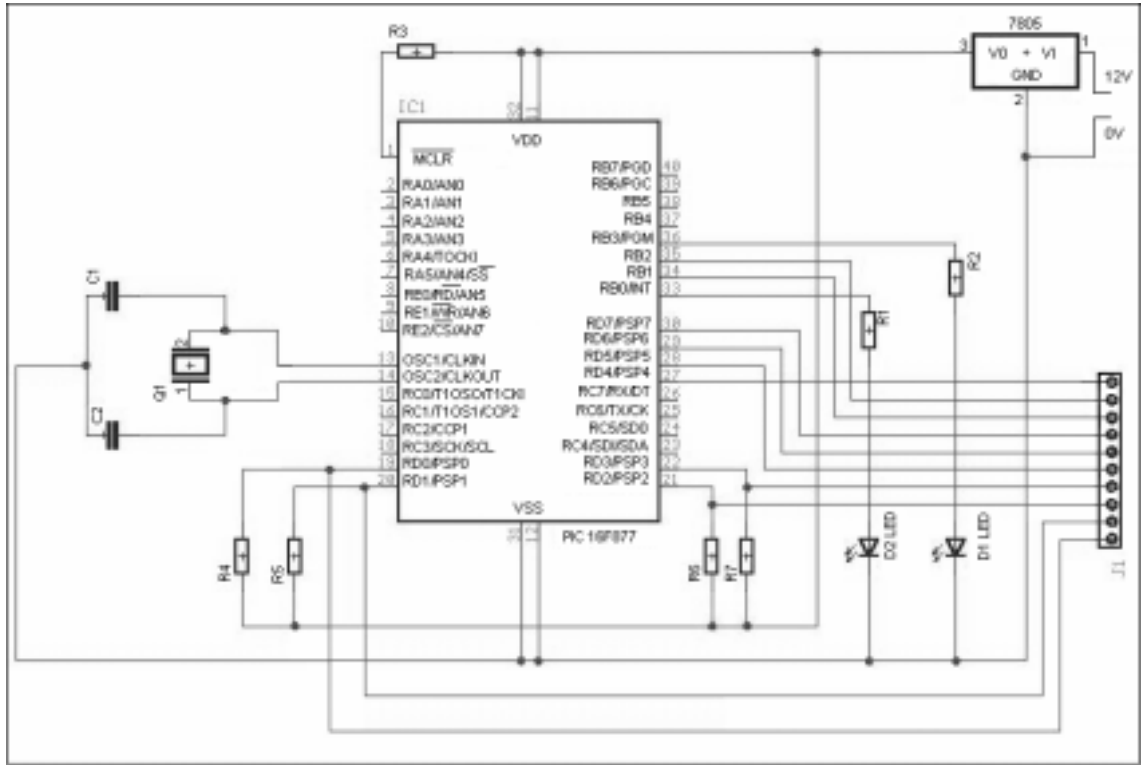
BÖLÜM V.

HIRSIZ ALARM SİSTEMİ İÇİN PROGRAM ve DEVRE UYGULAMALARI

5.1. Uygulama Devresi

Hırsız alarm sistemi uygulaması için gerçekleştirilen maket ev üzerinde bulunan sensörler, tuş takımı, ve buzzer arasındaki iletişimi sağlamak ve uygun program yazılımını gerçekleştirebilmek için yapılan ve Şekil 11.'de görülen devredir.

Devrede görülen L7805CV voltaj regülatörü güç kaynağından gelen 12 V luk gerilimi 5 V'a düşürüp sabitlemek için kullanılmıştır. Ayrıca 4 MHz kristal ve 22 pF kondansatör PIC'in clock darbesi için, kırmızı ve yeşil led ise ara işlem sinyal ledleri olarak kullanılmıştır. Maket içerisinde bulunan elemanlar ile bağlantı 25 (yirmi beş) pinli bir konnektörle yapılmaktadır Tablo 1.'de pin bağlantı numaraları verilmiştir.



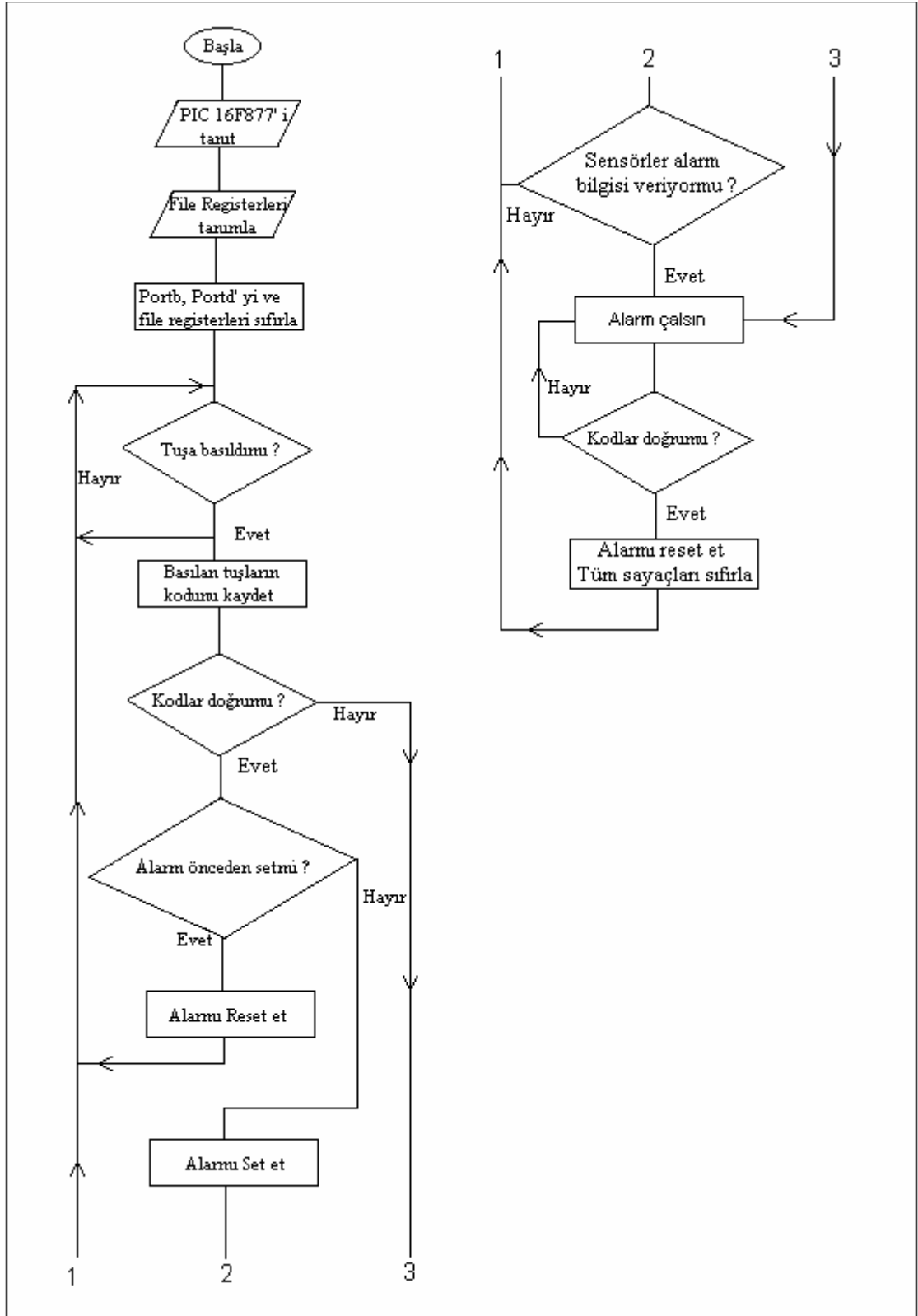
Şekil 11. Uygulama devresi

5.2. Program Akış Şeması

Bu akış şeması bu uygulamadaki alarm sistemi için istenilen özelliklere göre hazırlanmış olup bu akış şemasına göre yazılmış olan program EK-A da verilmiştir.

Programlamaya başlamadan önce yapılması istenen programın akış şeması çıkarılarak daha rahat program yazımı sağlanmıştır.

Bu uygulamada program yazmaya başlarken PIC 16F877 tanıtılmıştır. Daha sonra giriş/çıkış atamaları register tanımlamaları yapılmıştır. Programın icrası sırasında herhangi bir hata ile karşılaşmamak için bütün registerler, tüm çıkışlar başlangıç olarak sıfırlanmıştır. Bu işlemlerden sonra esas program yazılmıştır. Maket üzerinde bulunan tuş takımından herhangi bir tuşa basılmadıysa program tuşa basılana kadar tuş tarama işlemine devam etmektedir. Eğer tuşa basıldıysa basılan tuşun kodunu kaydetme işlemi gerçekleşmektedir. Daha sonra basılan tuşların kodları kontrol edilmektedir. Eğer kodlar doğruysa alarm set/reset olmaktadır. Alarm set olduysa pencere ve kapı sensörlerinden bilgi okumaktadır. Sensörlerden alarm bilgisi geldiğinde alarm çalmaktadır. Bu sırada tuş takımı okunmaktadır. Doğru şifrenin girilmesiyle alarm susup sistem reset olmaktadır ve tuş okumaya devam etmektedir. . Şifre doğru girilmediyse alarm çalmaya devam etmektedir. Bu işlemlerin şematik gösterimi Şekil 12.'de verilmiştir.



Şekil 12. Program akış şeması

BÖLÜM VI

SONUÇ VE ÖNERİLER

Bu çalışmada PIC 16F877 mikrodenetleyicisi ile maket bir eve hırsız alarm sistemi tasarlanıp gerçekleştirilmiştir. Kontrol işleminin yapılabilmesi için ilgili bölümlerde açıklandığı gibi, program geliştirme yazılımı, programlama yazılımı, programlama devresi ve uygulama devresi gibi bazı donanım ve yazılımların kullanılması gerekmektedir.

PIC mikrodenetleyicilerinin, çok geniş ürün ailesine sahip olmaları, ucuz fiyatları, EEPROM belleğe sahip olmaları, kelime kapasitelerinin yüksek olması vs. gibi özelliklerden dolayı gün geçtikçe popülerlikleri artmaktadır.

Bu tez çalışmasında PIC 16F877 kullanılmasından dolayı uzun program yazılımlarına olanak sağlanmakla birlikte giriş/çıkış adetinin fazla olması da ayrı bir rahatlıktır. Bu tez çalışmasında PIC programlama dili olarak PIC BASIC kullanıldığı için program geliştirmek daha kolay ve daha hızlı olmaktadır.

Bu hırsız alarm sistemi uygulaması için yazılan programın .HEX uzantılı dosyasının büyüklüğü 2064x14 bit olduğundan ve de sonradan programın geliştirilmesine olanak sağlamak amacıyla PIC 16F877 tercih edilmiştir.

Sonuç olarak, günümüz koşullarında can ve mal güvenliğimizi korumak için ev veya işyerimize alarm sistemi kurdurma gereksinimi artmaktadır. Ama hırsız alarm sistemlerinin fiyatı pahalı olduğundan bu uygulama üç oda bir salon ev için hırsız alarm sistemini ucuz ve güvenilir şekilde gerçekleştirmek amacı ile yapılmıştır.

EKLER

EK-A. BU UYGULAMA İÇİN HAZIRLANMIŞ .BAS DOSYASI

```
*****
'* Name      : selcuk.BAS                      *
'* Author    : [set under view...options]     *
'* Notice    : Copyright (c) 2002 [set under view...options] *
'*           : All Rights Reserved            *
'* Date      : 22.05.2002                      *
'* Version   : 1.0                             *
'* Notes     :                                 *
'*           :                                 *
*****

include "modedefs.bas"

trisb = %00000010
trisd = %00011111

i var byte
ii var byte
iii var byte
iiii var byte
ikur var byte
hane1 var byte
hane2 var byte
hane3 var byte
hane4 var byte
hane5 var byte
hane6 var byte
hane7 var byte
hane8 var byte
kur var byte
hane11 var byte
hane22 var byte
hane33 var byte
hane44 var byte
hane55 var byte
```

```
d var word
beskur var byte
timerset VAR BYTE
timerset = 0
low portb.2
low portb.3
low portb.0
low portb.4
beskur = 0
i = 0
ikur = 0
ii = 0
iii = 0
iiii = 0
kur = 0
ON INTERRUPT GOTO timer1
OPTION_REG = %00000111
INTCON = %10100000
ENABLE interrupt
basla1:
    if kur = 30 then alarmkur1
basla: portd.7 = 0
    portd.6 = 0
    portd.5 = 0
    if portd.3 = 0 then ucuncu_sutun
    if portd.2 = 0 then ucuncu_sutun
    if portd.1 = 0 then ucuncu_sutun
    if portd.0 = 0 then ucuncu_sutun
    goto basla1
ucuncu_sutun:
    pause 30
    if hanel = 11 then sifreyaz
    portd.7 = 1
    portd.6 = 1
    portd.5 = 0
    if portd.3 = 0 then ucuncu_tus
    portd.7 = 1
    portd.6 = 1
    portd.5 = 0
    if portd.2 = 0 then altinci_tus
    portd.7 = 1
```

```
portd.6 = 1
portd.5 = 0
if portd.1 = 0 then dokuzuncu_tus
portd.7 = 1
portd.6 = 1
portd.5 = 0
if portd.0 = 0 then enter_tusu
ikinci_sutun: portd.7 = 1
portd.6 = 0
portd.5 = 1
if portd.3 = 0 then ikinci_tus
portd.7 = 1
portd.6 = 0
portd.5 = 1
if portd.2 = 0 then besinci_tus
portd.7 = 1
portd.6 = 0
portd.5 = 1
if portd.1 = 0 then sekizinci_tus
portd.7 = 1
portd.6 = 0
portd.5 = 1
if portd.0 = 0 then sifir_tusu
birinci_sutun:
portd.7 = 0
portd.6 = 1
portd.5 = 1
if portd.3 = 0 then birinci_tus
portd.7 = 0
portd.6 = 1
portd.5 = 1
if portd.2 = 0 then dorduncu_tus
portd.7 = 0
portd.6 = 1
portd.5 = 1
if portd.1 = 0 then yedinci_tus
portd.7 = 0
portd.6 = 1
portd.5 = 1
if portd.0 = 0 then yildiz_tusu
goto basla
```

!*****

ucuncu_tus:

```
    i = i + 1
    if i = 1 then hane1 = 3
    if i = 2 then hane2 = 3
    if i = 3 then hane3 = 3
    if i = 4 then hane4 = 3
    if i = 5 then hane5 = 3
    if i = 6 then hane6 = 3
    if i = 6 then kontrol
```

bas:

```
    if portd.3 = 0 then bas
    goto basla
```

altinci_tus:

```
    i = i + 1
    if i = 1 then hane1 = 6
    if i = 2 then hane2 = 6
    if i = 3 then hane3 = 6
    if i = 4 then hane4 = 6
    if i = 5 then hane5 = 6
    if i = 6 then hane6 = 6
    if i = 6 then kontrol
```

bas1:

```
    if portd.2 = 0 then bas1
    goto basla
```

dokuzuncu_tus:

```
    i = i + 1
    if i = 1 then hane1 = 9
    if i = 2 then hane2 = 9
    if i = 3 then hane3 = 9
    if i = 4 then hane4 = 9
    if i = 5 then hane5 = 9
    if i = 6 then hane6 = 9
    if i = 6 then kontrol
```

bas2: if portd.1 = 0 then bas2

```
    goto basla
```

enter_tusu:

```
    i = i + 1
    if i = 1 then hane1 = 10
    if i = 2 then hane2 = 10
    if i = 3 then hane3 = 10
```

```
        if i = 4 then hane4 = 10
        if i = 5 then hane5 = 10
        if i = 6 then hane6 = 10
        if i = 6 then kontrol
bas4:
        if portd.0 = 0 then bas4
        goto basla
ikinci_tus:
        i = i + 1
        if i = 1 then hane1 = 2
        if i = 2 then hane2 = 2
        if i = 3 then hane3 = 2
        if i = 4 then hane4 = 2
        if i = 5 then hane5 = 2
        if i = 6 then hane6 = 2
        if i = 6 then kontrol
bas5:
        if portd.3 = 0 then bas5
        goto basla
besinci_tus:
        i = i + 1
        if i = 1 then hane1 = 5
        if i = 2 then hane2 = 5
        if i = 3 then hane3 = 5
        if i = 4 then hane4 = 5
        if i = 5 then hane5 = 5
        if i = 6 then hane6 = 5
        if i = 6 then kontrol
bas6:
        if portd.2 = 0 then bas6
        goto basla
sekizinci_tus:
        i = i + 1
        if i = 1 then hane1 = 8
        if i = 2 then hane2 = 8
        if i = 3 then hane3 = 8
        if i = 4 then hane4 = 8
        if i = 5 then hane5 = 8
        if i = 6 then hane6 = 8
        if i = 6 then kontrol
bas7:  if portd.1 = 0 then bas7
```

```
        goto basla
sifir_tusu:        i = i + 1
        if i = 1 then hanel = 0
        if i = 2 then hane2 = 0
        if i = 3 then hane3 = 0
        if i = 4 then hane4 = 0
        if i = 5 then hane5 = 0
        if i = 6 then hane6 = 0
        if i = 6 then kontrol

bas8:
        if portd.0 = 0 then bas8
        goto basla

birinci_tus:
        i = i + 1
        if i = 1 then hanel = 1
        if i = 2 then hane2 = 1
        if i = 3 then hane3 = 1
        if i = 4 then hane4 = 1
        if i = 5 then hane5 = 1
        if i = 6 then hane6 = 1
        if i = 6 then kontrol

bas9:
        if portd.3 = 0 then bas9
        goto basla

dorduncu_tus:
        i = i + 1
        if i = 1 then hanel = 4
        if i = 2 then hane2 = 4
        if i = 3 then hane3 = 4
        if i = 4 then hane4 = 4
        if i = 5 then hane5 = 4
        if i = 6 then hane6 = 4
        if i = 6 then kontrol

bas10:
        if portd.2 = 0 then bas10
        goto basla

yedinci_tus:
        i = i + 1
        if i = 1 then hanel = 7
        if i = 2 then hane2 = 7
        if i = 3 then hane3 = 7
```

```
        if i = 4 then hane4 = 7
        if i = 5 then hane5 = 7
        if i = 6 then hane6 = 7
        if i = 6 then kontrol
bas11:
        if portd.1 = 0 then bas11
        goto basla
yildiz_tusu:
        i = i + 1
        if i = 1 then hane1 = 11
        if i = 2 then hane2 = 11
        if i = 3 then hane3 = 11
        if i = 4 then hane4 = 11
        if i = 5 then hane5 = 11
        if i = 6 then hane6 = 11
        if i = 6 then kontrol
bas12:
        if portd.0 = 0 then bas12
        goto basla
kontrol:
        i = 0
        if hane6 = 10 then karsilastir
        goto tekrarhak
karsilastir:
        if hane8 = 16 then yenisifregirl
karsilastir0:
        read 0,hane11
        if hane11 = hane1 then karsilastir1
        goto tekrarhak
karsilastir1:
        read 1,hane22
        if hane22 = hane2 then karsilastir2
        goto tekrarhak
karsilastir2:
        read 2,hane33
        if hane33 = hane3 then karsilastir3
        goto tekrarhak
karsilastir3:
        read 3,hane44
        if hane44 = hane4 then karsilastir4
        goto tekrarhak
```

```
karsilastir4:
    read 4,hane55
    if hane55 = hane5 then setreset
    goto tekrarhak
setreset:
    if hane7 = 15 then yenisifregir
    if kur = 30 then reset
    if beskur = 10 then reset
alarmkur:
    i = -1
    kur = 30
alarmkur1:
    high portb.0
    pause 25
    low portb.0
    pause 35
    if portb.1 = 1 then alarm
    goto basla
sifreyaz:
    i = 0
    hane1 = 30
    hane7 = 15
    goto basla
yenisifregir: hane8 = 16
    high portb.3
    pause 700
    i = 0
    goto basla
yenisifregir1:
    low portb.3
    write 0,hane1
    pause 10
    write 1,hane2
    pause 10
    write 2,hane3
    pause 10
    write 3,hane4
    pause 10
    write 4,hane5
    pause 10
    high portb.3
```

```
    pause 500
    low portb.3
    hane8 = 25
    i = 0
    hane7 = 25
    goto basla
tekrarhak:
    hane7 = 25
    i = -1
    ii = ii + 1
    if ii = 5 then alarm
    high portb.3
    pause 60
    low portb.3
    goto basla
alarm:
    iiii = iiii + 1
    if iiii > 1 then cik
    if ii = 5 then ikur1
    i = 0
    goto ikur2
ikur1:
    i = -1
ikur2:
    ii = 0
    iii = 0
    beskur = 10
    timerset = 10
    hane7 = 25
    d = 0
    high portb.2
    low portb.0
    pause 10
cik:
    enable interrupt
    goto basla
reset:    i = -1
    iii= 0
    ii = 0
    iiii = 0
    kur = 19
```

```
beskur = 0
low portb.0
pause 10
low portb.2
pause 10
timerset = 10
hane7 = 25
d = 0
goto basla
```

DISABLE INTERRUPT

```
timer1:
    if kur <> 30 then CIKIS
    if timerset = 80 then cikis3
    if portd.4 = 1 then cikis1
    goto CIKIS
cikis1:
    timerset = 80
cikis3:
asm
    bcf status,2
    bcf status,0
endasm
    if portd.3 = 0 then CIKIS
    if portd.2 = 0 then CIKIS
    if portd.1 = 0 then CIKIS
    if portd.0 = 0 then CIKIS
asm
    bcf status,2
    bcf status,0
endasm
    d = d + 1
    if d > 35000 then say
    GOTO CIKIS
say:    d = 0
        timerset = 10
        iii = iii + 1
        if iii = 1 then CIKIS
        i = 0
        ii = 0
        iii = 0
```

```
hane7 = 25
high portb.2
pause 10
CIKIS:
RESUME
ENABLE INTERRUPT
END
```

KAYNAKLAR

1. Altınbaşak O., 2000, Mikrodenetleyiciler ve Pic Programlama, Atlas Yayınevi, İstanbul
2. Turgutlu H.F., 2002, Pic Mikrodenetleyicisi Kullanarak Deneysel Bir Endüstriyel Sistemin kontrol edilmesi. Yüksek Lisans Tezi, Niğde
3. Microchip, 2000, Microchip Tecnical Library CD-ROM, Microchip Tecnology Inc.
4. Antrak Gazetesi, Pic Uygulamaları, <http://www.antrak.org.tr>